Outcomes, Perceptions, and Interaction Strategies of Novice Programmers Studying with ChatGPT

JACOB PENNEY, Northern Arizona University, USA
PAWAN ACHARYA, Northern Arizona University, USA
PETER HILBERT, Northern Arizona University, USA
PRIYANKA PAREKH, Northern Arizona University, USA
ANITA SARMA, Oregon State University, USA
IGOR STEINMACHER, Northern Arizona University, USA
MARCO A. GEROSA, Northern Arizona University, USA

Large Language Model (LLM) conversational agents are increasingly used in programming education, yet we still lack insight into how novices engage with them for conceptual learning compared with human tutoring. This mixed-methods study compared learning outcomes and interaction strategies of novices using ChatGPT or human tutors. A controlled lab study with 20 students enrolled in introductory programmig courses revealed that students employ markedly different interaction strategies with AI versus human tutors: ChatGPT users relied on brief, zero-shot prompts and received lengthy, context-rich responses but showed minimal prompt refinement, while those working with human tutors provided more contextual information and received targeted explanations. Although students distrusted ChatGPT's accuracy, they paradoxically preferred it for basic conceptual questions due to reduced social anxiety. We offer empirically grounded recommendations for developing AI literacy in computer science education and designing learning-focused conversational agents that balance trust-building with maintaining the social safety that facilitates uninhibited inquiry.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**.

Additional Key Words and Phrases: AI Literacy, Large Language Models, Conversational Agents, Computer Science Pedagogy, CS1, Software Engineering Education

ACM Reference Format:
Jacob Penney, Pawan Acharya, Peter Hilbert, Priyanka Parekh, Anita Sarma, Igor Steinmacher, and Marco A. Gerosa. 2025. Outcomes, Perceptions, and Interaction Strategies of Novice Programmers Studying with ChatGPT. In *Proceedings of the 7th ACM Conference on Conversational User Interfaces (CUI '25), July 8–10, 2025, Waterloo, ON, Canada.* ACM, New York, NY, USA, 22 pages. https://doi.org/10.1145/3719160.3736625

Authors' addresses: Jacob Penney, jacob_penney@nau.edu, Northern Arizona University, Flagstaff, AZ, USA; Pawan Acharya, pa577@nau.edu, Northern Arizona University, Flagstaff, AZ, USA; Peter Hilbert, peh53@nau.edu, Northern Arizona University, Flagstaff, AZ, USA; Priyanka Parekh, Priyanka.Parekh@nau.edu, Northern Arizona University, Flagstaff, AZ, USA; Anita Sarma, anita.sarma@oregonstate.edu, Oregon State University, Corvallis, OR, USA; Igor Steinmacher, igor.steinmacher@nau.edu, Northern Arizona University, Flagstaff, AZ, USA; Marco A. Gerosa, marco.gerosa@nau.edu, Northern Arizona University, Flagstaff, AZ, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
Manuscript submitted to ACM

1 INTRODUCTION

Programming literacy is fundamental to the global economy [20, 21, 30, 47], but learning to program is still difficult. Successfully completing introductory programming courses (CS1) requires developing specialized knowledge and diverse skills collectively called "computational thinking". This encompasses understanding notional machines to reason about computing environments, applying abstraction and problem decomposition to simplify complex problems, designing algorithms for solving these decomposed problems, mastering programming languages to express solutions, and utilizing appropriate tools to work effectively within those languages [39]. Novice programmers are required to develop all these competencies simultaneously while facing assessments that often exceed their current capabilities [2]. This demanding learning process leads to failures, with estimates indicating global CS1 pass rates of only 72% [3]. Students cite various challenges that contribute to attrition, including difficulty catching up after falling behind in their high workload [43], inadequate academic support and real-time feedback [7], and consequent poor grades [18, 39].

Students of introductory courses count on alternative resources to support them. Studies conducted in the last decades discuss how they make use of online Q&A and websites (e.g., StackOverflow, geeks4geeks) to aid their learning process [40]. More recently, students have been using Large Language Model-based conversational agents (LLM-based chatbots) to complement their classroom experience and help develop computational thinking skills. LLM-based chatbots have proven disruptive for programming education: they can quickly and accurately solve CS1 and CS2 problems [16, 17], and instructors and institutions are taking stances concerning adopting or forbidding them [5, 35, 45]. Research on how students use these LLM-based tools and their user experience in programming education is valuable for informing the development of chatbots for programming pedagogy. The literature on novice student interaction with AI for programming tasks in the education context is booming [45]. However, there is a gap in understanding how students interact with LLM-based chatbots to learn a programming concept.

This research seeks to reduce this gap by investigating the following research questions:

RQ1 *How do novice programming students' learning outcomes compare when studying programming concepts with ChatGPT versus with a human tutor over a conversational interface?*

RQ2 *How do novice programming students perceive ChatGPT as a learning resource for programming concepts?*

RQ3 *What strategies do novice programming students employ when interacting with ChatGPT to study programming concepts?*

To answer these questions, we conducted a lab study with 20 novice programming students from an American university. We divided the participants into two groups, one studying with ChatGPT and the other with a human over Discord, and observed how they interacted with their corresponding tutor to learn a programming concept. We found that novice programming students currently do not use prompt engineering techniques or provide contextualizing information in their prompts, do not engage deeply with the resulting outputs, and do not like using LLM-based chatbots for learning new programming concepts, but display higher comfort with asking LLM-based chatbots questions than they do with human tutors over instant messaging.

2 THEORETICAL FRAMEWORK

To better understand how students engage with chatbots and refine their learning strategies, we apply reflective practice as a lens for analysis [52]. Schön's framework is particularly relevant in AI-mediated learning environments, where students interact with digital tutors such as chatbots to meet specific learning goals and to learn to interact with the chatbot. His theory distinguishes between two modes of reflection: reflection-in-action, which involves critical thinking

and real-time adjustment, and reflection-on-action, where learners retrospectively evaluate their experiences to inform future practices.

In chatbot-assisted learning, students reflect-in-action as they iteratively refine their prompts based on chatbot responses [29], actively modifying their queries to elicit clearer explanations or more relevant examples, demonstrating adaptability and strategic thinking. Others exhibit minimal prompt engineering, suggesting a more static approach to chatbot interaction. This variation shows the need to further understand student-chatbot interactions through a lens that provides a structure for analyzing these interactions as continuous experimentation and refinement within their learning environments.

Students engage in reflection-on-action when assessing the reliability and effectiveness of chatbot-generated responses [19]. While some students recognize the strengths of AI tools, such as accessibility and quick response times, others critique their lack of pedagogical structuring and potential inaccuracies. These findings align with Schön's argument that professionals develop expertise through assessing past experiences and refining their approaches. The ability to critically evaluate AI-generated responses and adjust engagement strategies demonstrates an evolving understanding of digital tutoring [11].

Reflective practice captures students' navigation of multiple possibilities embedded in ChatGPT's responses, human tutors, and their internalized disciplinary knowledge. Reflection-in-action enables students to evaluate the credibility of chatbot outputs in real time, while reflection-on-action informs their future trust and engagement with chatbot assistance. This framework emphasizes the importance of fostering critical thinking and metacognitive awareness among students using AI tutors [32]. For example, structured interventions, such as prompt engineering exercises and guided discussions on AI literacy, can encourage deeper engagement with AI tools. Similarly, designing chatbot interfaces to integrate reflection prompts and adaptive scaffolding can foster iterative experimentation with queries and support the student in their zone of proximal development [57], positioning learners as active co-creators of knowledge rather than passive recipients of information.

In this study, the unit of analysis is the student–tutor interaction episode, which includes the student's prompt formulation, the tutor's (human or AI) response, and the student's real-time or retrospective evaluation of the interaction. This choice of unit highlights the study's focus on learning processes: the dynamic adjustments and reflections students enact, rather than solely on final learning outcomes. In AI-mediated learning, these episodes often require learners to navigate unpredictable outputs, adapt their strategies in action, and reassess their approaches afterward, making reflective practice especially appropriate. In human tutor-mediated learning, interaction episodes often require learners to navigate social dynamics, interpret nuanced feedback, and adjust their questioning or understanding in response to implicit cues, making reflective practice especially appropriate. In either setting, the process of mediation of human learning by an agent (social or technological) is the focus of this study. Within such mediation, examining learners' reflective processes is critical for understanding how they develop strategic engagement with diverse tutoring systems, and for informing the design of interventions that foster deeper metacognitive awareness, adaptability, and learning agency.

While alternative frameworks prioritizing self-regulation [61] or the influence of culture in shaping the use of AI as a tool [13] could offer broader insights, reflective practice underscores the micro-level, situated adaptations observed when learners engage in experimental inquiry and strategy refinement. Within this study, practice is understood as the students' situated interactional behaviors: prompt formulation, response evaluation, strategy adjustment, and critical reflection, through which they construct their learning trajectories in an evolving AI-mediated environment.

Applying Schön's reflective practice framework, this study provides a structured approach to understanding how students develop strategic engagement with AI tutors and refine their learning behaviors over time. This theoretical framing highlights the iterative, adaptive nature of chatbot interactions and the broader implications for AI-enhanced learning chatbots. In the following sections, we explore how this perspective informs our study's findings, particularly regarding students' perceptions of AI as an educational tool (RQ2) and learning strategies (RQ3).

3 RELATED WORK

This research sits at the intersection of LLM-based chatbots in programming education, student help-seeking and interaction with AI, and social factors around academic help-seeking.

3.1 Chatbots in CS Education

The competencies of chatbots concerning reviewing, editing, and producing code are established and continue to grow, and researchers are displaying how those abilities can address the needs of CS educational stakeholders. Phung et al. [44] examined LLM performance in six education scenarios and found that GPT-4 (the flagship model as of March 2024) "drastically outperforms ChatGPT (based on GPT-3.5) and comes close to human tutors' performance for several scenarios", including program repair, pair programming, and contextualized explanation. However, GPT-4 struggled with jobs such as grading feedback and task synthesis. Jury et al. [31] developed a tool to iteratively develop worked examples using prompt engineering and conducted a user study with CS1 students. They found that LLMs can generate worked examples with meaningful, logical step decomposition that experts found clear and that students found useful. Leinonen et al. [36] asked students in a university introductory programming course to rate the clarity of code explanations produced by peers and ChatGPT (GPT-3). Quantitative results display that the students preferred GPT-3's explanations because they were perceived as being more lucid and accurate. In their thematic analysis of student responses to being asked about what qualities make a code explanation useful, the authors found that students enjoyed line-by-line explanations and engaged in more thorough adaptive help-seeking [51], such as "request[ing] examples, templates, and the thought process behind how the code was written". Balse et al. [1] found that GPT-3-turbo produced explanations about syntactically correct code with logical errors that undergraduate TAs could not distinguish from peer-created explanations and which correctly identified at least one logical error 93% of the time. Hoq et al. [27] compared the effectiveness of traditional and Abstract Syntax Tree-based machine learning models to detect ChatGPT-generated code in CS1 student submissions and found that both had above 90% accuracy. They also compared structural differences between novice students and ChatGPT-produced code and found that ChatGPT routinely creates optimized and compact code that has less variation in it compared to student solutions. At scale, Liu et al. embedded a suite of bespoke GPT-based assistants (CS50.ai) into the CS50 MOOC, demonstrating massive uptake but reporting only descriptive interaction analytics and no direct learning-outcome evaluation [37].

3.2 Student-AI Interaction and Help-Seeking Among Programming Students

LLM-based chatbots have rapidly emerged as significant learning resources in computer science education, with recent studies showing widespread adoption among programming students [28]. While researchers have begun examining these interactions, fundamental questions remain about how students effectively utilize these tools for learning. Prather et al. [46] conducted a systematic observational study of novice programmers using generative AI for code generation in CS1 assignments, documenting not only basic interaction patterns but also uncovering complex cognitive and metacognitive challenges students face when integrating AI tools into their learning process. Sheese et al. [55] deployed

an LLM-powered tool in an introductory computer and data science course for 12 weeks, collecting and categorizing over 2,500 student queries to analyze how students seek on-demand programming assistance. They found that most queries focused on immediate assignment help, with minimal information provided by students, and that tool usage was positively correlated with course success. Rogers et al. [50] conducted an online survey to gauge CS students' awareness, experiences, and attitudes regarding ChatGPT, using both quantitative and qualitative analyses. They found that most students use ChatGPT primarily as a study tool, while a notable minority admit to unscrupulous usage. Haindl and Weinberger [25] conducted a five-week study involving part-time undergraduate students who used ChatGPT for Java programming exercises and provided feedback through anonymous surveys. They found that their participants primarily used it for learning background knowledge and programming concepts, though some avoided using it for fear of not developing programming experience and receiving untrustworthy code outputs. Garg et al. [22] analyzed 411 Distributed Systems (non-novice) students' interactions with LLMs and found that students used a spectrum of prompting strategies with LLMs for tasks such as code generation, debugging, and conceptual inquiries. Xue et al. [59] carried out an experiment with 56 participants split into two groups, where the experimental group could use ChatGPT 3.5 and other online resources, while the control group had access to everything except ChatGPT. They found that ChatGPT access did not significantly improve learning performance but led students to rely heavily on the chatbot (reducing their use of other resources). Choudhuri et al. [9] conducted a between-subjects study with 22 students to assess ChatGPT's effectiveness for software engineering tasks compared to traditional resources. They found no significant differences in productivity or self-efficacy when using ChatGPT but observed higher frustration levels. Skjuve et al. captured early users' perceptions of ChatGPT across everyday tasks and found that pragmatic usefulness and surprise drive positive experiences, whereas hallucinations and prompt-craft burden erode trust [56]. Ouaazki et al. [41] investigated the impact of using tuned GPT-3 prompts to structure self-directed computational thinking labs. While this approach enhanced accuracy, student usability ratings decreased, and the effectiveness of the tutor diminished as students advanced through open-ended projects.

3.3 Help-seeking Avoidance and Social Factors

Reeves and Sperling [49] surveyed 226 students in an introductory educational psychology class about how comfortable they felt asking for help and how likely they were to use six different types of help sources (for example, face-to-face vs. online). Despite some anxiety about asking for help in person, most students still preferred face-to-face help (especially before or after class), and higher-performing students tended to use in-person methods more often, while lower-performing students leaned more on online options like discussion boards and virtual office hours. Qayyum [48] surveyed 438 college students about their help-seeking behaviors and applied factor analysis to uncover six motivational factors for seeking help from peers and instructors. They found that students overwhelmingly preferred asking classmates for help, and that "sense of vulnerability about their ability" and perception of instructors were key predictors of whether students sought help from instructors outside of class. This work also found that students enrolled in distance education courses felt less intimidated about seeking help than their in-person peers. Downing et al. [12] interviewed 29 community college science students to explore what made them feel anxiety in classes that emphasized hands-on, interactive learning. They discovered that students felt less anxious when active-learning activities helped them learn in multiple ways or from each other, while a fear of being judged by peers or instructors remained a key source of anxiety. Zander and Höhne [60] used a cross-sectional survey of 418 undergraduate students across 25 seminar and tutorial groups in computer science and education to measure perceived peer exclusion and self-reported help-seeking strategies. They found that computer science students reported lower autonomy-oriented help-seeking and higher

avoidance overall, with perceived peer exclusion significantly predicting increased help-seeking avoidance and reduced autonomy-oriented strategies.

3.4 Our Contribution

Our study advances the understanding of conversational AI in education by addressing several key limitations in prior work. While Skjuve et al. [56] provided valuable insights into user enthusiasm and pragmatic usability through survey-based research, their reliance on self-reported data limited direct observation of student-chatbot interactions. Our methodology enhances this understanding by capturing and analyzing actual conversational exchanges at a granular, turn-by-turn level, revealing detailed patterns in how students construct prompts and engage with AI-based tutors.

Building on Ouaazki et al.'s [41] investigation of prompt scaffolding in authentic course settings, our work introduces two critical dimensions: a systematic comparison with human tutoring and a comprehensive analysis of students' prompt formulation strategies. This comparative approach, absent in Ouaazki's methodology, allows us to identify unique characteristics of AI-based tutoring interactions while our detailed prompt categorization framework provides deeper insights into students' conversational strategies.

Liu et al. [37] demonstrated the scalability of GPT-powered educational assistants in MOOCs, primarily through descriptive analytics. While their work established feasibility at scale, our controlled laboratory study, coupled with in-depth qualitative analysis, offers a more comprehensive understanding of the conversational dynamics between novice learners and AI tutors. This methodological approach enables us to uncover subtle patterns in student-tutor interactions that may be overlooked in larger-scale, purely quantitative analyses.

This research also expands on existing literature by offering new insights into how students use AI chatbots to learn programming concepts. While prior experimental work has compared ChatGPT with various learning resources, our study specifically examines its similarities and differences relative to human tutoring in an online setting. Prior survey-based work by Rogers [50] and Haindl and Weinberger [25] indicates that most students already perceive ChatGPT primarily as a study tool, yet these studies do not investigate the detailed interaction dynamics or specific learning processes involved. Additionally, earlier research, such as studies by Prather [46] and Sheese [55], has focused heavily on ChatGPT's ability to generate functional code rather than on its effectiveness in helping students internalize programming concepts. Hou's study [28] provided qualitative student perceptions of generative AI tools, but did not focus explicitly on learning concepts and lacked rich, post-interaction reflections directly tied to student experiences. By integrating mixed methods to capture students' experiences and drawing explicitly from educational and psychological perspectives, we illuminate the social factors that inhibit help-seeking, show how LLM-based chatbots fit into these dynamics, and explore how conversational user interfaces can leverage these insights to better support learners.

4 RESEARCH DESIGN

To answer our research questions, we designed a four-phase lab study as presented in Figure 1. In Phase 1, we used a brief test and questionnaire to establish participants' baseline knowledge of the study topic and their trust in and sentiments about LLM-based chatbots. In Phase 2, the participants studied the topic using their assigned tutor (ChatGPT or human). In Phase 3, the participants' knowledge, trust, and sentiments are measured again via a test and survey. In Phase 4, participants underwent a debriefing interview. We received approval for this research design through our research institution's review board (IRB). All study artifacts are included in an artifact package[1].

[1] https://osf.io/mb3du/?view_only=35496af031ab482aa62ca6937c77a32c

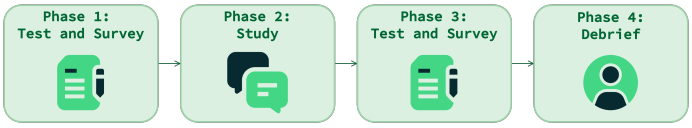

Fig. 1. Overview of the four lab study phases that each participant completed

4.1 Recruitment

Table 1. ChatGPT Group

ID	Level	Major	Age	Gender	English
P1	Ugrad	Comp Sci	18-20	M	L1
P2	Ugrad	English	21-30	M	L2
P3	Ugrad	Ecoinformatics	18-20	W	L1
P4	Ugrad	Comp Eng	21-30	M	L1
P5	Grad	Comp Sci	21-30	M	L2
P6	Grad	Comp Sci	21-30	M	L2
P7	Grad	Comp Sci	21-30	M	L2
P8	Grad	Comp Sci	21-30	M	L2
P9	Grad	Comp Sci	21-30	M	L2
P10	Grad	Comp Sci	21-30	M	L2

Table 2. Discord Group

ID	Level	Major	Age	Gender	English
P11	Grad	Comp Sci	21-30	M	L2
P12	Grad	Comp Sci	21-30	M	L2
P13	Grad	Comp Sci	21-30	M	L2
P14	Grad	Comp Sci	31-40	M	L2
P15	Ugrad	Comp Sci	18-20	M	L2
P16	Grad	Comp Sci	21-30	M	L2
P17	Grad	Comp Sci	21-30	M	L2
P18	Grad	Comp Sci	21-30	W	L2
P19	Grad	Comp Sci	21-30	M	L2
P20	Grad	Comp Sci	21-30	W	L2

We recruited 20 participants through volunteer and snowball sampling. We recruited all participants from a pool of 340 students enrolled across nine sections of three introductory programming courses at an American public university during the Spring and Fall 2024 semesters. We advertised between February and November 2024 using two approaches: (1) distributing study flyers through course instructors via email; and (2) direct in-class presentations by a researcher. Interested students completed a screening questionnaire assessing their prior experiences with and attitudes toward using chatbots for educational purposes. Of the 40 students who completed this initial survey, five were excluded for not being in the target classes at the time of the study.

A researcher then contacted all eligible respondents to schedule 30-60 minute digital study sessions via Zoom, with all 20 respondents who replied being included in the study. The final participant demographics are presented in Table 1 and Table 2. Graduate student participants were included in our introductory classification because they were enrolled in introduction to programming courses designed for students without prior programming experience, usually coming from non-computing-related majors. L1 and L2 indicate the participant knows English as a first and second language, respectively.

4.2 Study Sessions

For each phase of the study (see Figure 1, the proctor introduced the task and then disabled their camera and microphone—only the proctor did so—while remaining in the call to answer clarifying questions. Participants, in contrast, shared their screens and microphones during the whole session, so that we could observe and record their interactions with the assigned tutor. We allocated five minutes for each of phases 1 and 3, and 10 minutes for Phase 2. Clarifying questions did

not count against that time. Participants could not request assistance with the study tasks or revisit previous sections. All participant questions are documented in the artifact package[2].

Phase 1: Pre-assessment - Participants completed a quiz on programming topics aligned with their current coursework, followed by a sentiment questionnaire regarding human and AI tutors. All participants were studying the C programming language and were assessed on their understanding of pointers through three multiple-choice questions and one long-answer question. The sentiment questionnaire utilized a seven-point Likert scale to measure initial attitudes about the ease of use, usability, and trustworthiness of LLM-based chatbots, human tutors, and peers.

Our knowledge quiz and sentiment questionnaire were not drawn from standardized or previously validated instruments; instead, we constructed custom items that align with the learning outcomes of the targeted courses and our focus on novice understanding of pointers and attitudes toward human- vs. AI-based tutoring. Because this study is exploratory, we used these instruments to gather preliminary data on changes in knowledge and sentiment, rather than conducting formal psychometric validation. We designed the quiz at a difficulty appropriate for CS1 learners, and anchored the survey questions in established constructs such as usability and trust. Although not validated, these measures map directly to our core research questions and provide a starting point for further refinement in future studies.

Phase 2: Learning through the chat interface - Participants studied the topic with their assigned tutor, either ChatGPT (GPT-3.5, the free version available during data collection) or a human tutor communicating via Discord. The human tutor was identified to participants only as someone with formal computer science education. We refer to these as the ChatGPT and Discord groups, respectively. Participants were given the freedom to approach the learning task however they preferred, including asking for answers to the Phase 1 questions if they recalled them.

Phase 3: Post-assessment - Participants completed an identical quiz and sentiment questionnaire to measure knowledge acquisition and changes in attitudes toward their tutor after the Phase 2 interaction. This repetition was intentionally not disclosed during the study introduction to avoid biasing participant questioning strategies.

Phase 4: Debrief - We used a semi-structured interview protocol to explore participants' experiences, perceptions, and reflections on the tutoring process. This phase had no time limit and provided qualitative context for understanding participant experiences (see Table 5).

5 DATA COLLECTION AND ANALYSIS

To address our research questions, we collected and analyzed five sources of data, detailed in the following.

Quizzes: We were interested in seeing how each participant and each group gathered, consumed, and retained information while studying through their conversational interface. To that end, each study session included two quizzes (pre-study and post-study) on pointers, a programming concept relevant to the classes the participants were in but that the participants had not yet studied in the course. Identical quizzes were administered directly before and after the study period (Phases 1 and 3), enabling measurement of score changes within individuals and across groups. We checked the normality of the quiz scores using the Shapiro-Wilk test, then conducted a paired t-test to see if there were meaningful changes between the pre- and post-intervention scores. We also conducted Cohen's d to measure the magnitude of differences between scores, and Bayesian analysis to gain a direct probabilistic estimate of the likelihood that the true mean difference is greater than zero. Additionally, we used Levene's test, Student's independent t-test, and Bayesian analysis to compare the pre-to-post differences between the GPT group and the Discord group.

[2] https://osf.io/mb3du/?view_only=35496af031ab482aa62ca6937c77a32c

Questionnaires: As with the quiz data, we were interested in measuring changes in the participants' perceptions of trust, ease of use, and perceived usefulness concerning their tutor after studying with them. To measure these attitudinal factors, each participant completed two questionnaires, pre-study (Phase 1) and post-study (Phase 2), with items using a 7-point Likert scale. We asked participants in each group five common questions about their trust in and sentiments about ChatGPT (we will refer to this value as n_{GPT}). We asked the Discord group four additional questions for a total of nine questions (we will refer to this value as $n_{Discord}$). We analyzed individual changes in sentiment for each survey question using the Wilcoxon test. Because we performed multiple tests at once, we used a false discovery rate (FDR) approach to correct for multiple comparisons, ensuring control over the proportion of false positives among significant results. For the survey questions that were common between the two groups (the first n_{GPT} questions in the survey), we measured the changes in responses for each group and performed comparative analysis with these deltas using a Mann-Whitney U Test. Again, since we performed multiple tests, we applied a FDR.

Think-aloud protocols: To understand participants' cognitive processes, we implemented think-aloud protocols [14] throughout the study sessions. Participants were instructed to verbalize their thoughts, questions, and decision-making processes while interacting with both assessment materials and tutors. When participants fell silent, proctors provided gentle reminders to continue vocalizing their thinking. This methodological approach revealed participants' reasoning strategies, moments of confusion, and engagement patterns that would otherwise remain hidden in conventional observation. We analyzed the collected data using a systematic approach of open coding to identify initial patterns, followed by axial coding to establish relationships between emerging concepts [10].

Tutor interactions: A central focus of our study was examining how novice programming students elicit information from LLM-based chatbots compared to human tutors through conversational interfaces. We recorded and transcribed all interactions between participants and their assigned digital tutors. The dataset captured multidimensional interaction elements: from participants, we documented messages, interaction pauses, scrolling behaviors, and paralinguistic cues (such as emoji reactions in Discord); from tutors, we collected all textual responses. This comprehensive data allowed us to analyze question formulation strategies, response interpretation patterns, and participants' adaptive behaviors throughout the tutoring sessions.

To systematically analyze participant questions, we leveraged established taxonomies while extending them to accommodate the conversational nature of digital tutoring. Namely, we employed the Graesser, Person, and Huber (GPH) taxonomy [24] to categorize factoid questions (FQs) alongside Bolotova et al.'s non-factoid question (NFQ) taxonomy [4], which builds upon GPH. One researcher categorized all participant questions, first distinguishing between factoid questions, which only require short answers on facts [24, 58], and NFQs, which typically require long-form answers containing explanations or opinions [4]. The categorizing researcher brought their code book back to the research team to discuss the fit of the codes they developed. During this process, we identified the need to add "command", "statement", and "greeting" categories to capture the full spectrum of communicative acts that extended beyond traditional question-answer exchanges. Complex participant messages often received multiple categorizations. We also categorized questions according to whether they pertained to the quiz questions, continued off of the tutor's previous response, and if the content the participant asked for was very explicitly covered in another response (Table 4). We did this because we were interested in seeing how participants from each group were engaged in the question-asking and response-reading process.

For the tutor responses, two researchers developed a novel taxonomy using the codes that emerged from the iterative open coding procedures. This was necessary since the existing literature lacked frameworks for categorizing tutor or LLM-based chatbot responses. We developed our code book (see Table 3) through four iterations, with meetings after

CODE	DEFINITION
Admission of Ignorance	An admission by the tutor to not knowing the answer.
Clarification	Clarification of the participant's misunderstanding or confirmation of something the participant asked; can be positive or negative; may be a response to a disjunctive-type factoid question. Distinct from procedure because procedure is step-by-step instructions on how to do something where clarification indicates what can be done and/or how to do it without step-by-step instructions.
Clarification Request	A request for more information by the tutor.
Code Example	A code-based example of something; may be part of a procedure, but doesn't always constitute a procedure.
Code Explanation	An explanation of a piece of code, either provided by the participant or by the tutor.
Comparison	Closely related to evidence-based; a comparison is an evidenced-based discussion of two things in juxtaposition.
Context	Context or relationships to other things; for example, how a concept relates to another concept; may be closely related to the "feature" tag. An indication that a snippet of a response is "context" is that it doesn't answer the original question but is relevant and tangentially related to the answer to or concept of the original question.
Diagram	A diagram given by the tutor to illustrate something, such as a concept or mechanism.
Evidence-Based	A definition of something, such as a term or a concept, or an elaboration of features or characteristics of something; what the thing is, can do, or be used for, risks or benefits, advantages and disadvantages.
Procedure	Step-by-step guidance or walkthrough of something, such as how to achieve a goal or what a piece of code is doing. May subsume other categories; for example, "code example", "code explanation", and/or "diagram" may constitute a procedure, but not always.
Reason	The reason for something, an explanation of the underlying mechanism of something; not a definition nor the features or characteristics of something, but an explanation of why or how something works.
Recommendation	A recommendation about what the user should do.
Refusal	A refusal to produce the content the participant asked for.

Table 3. Question Response Types

Category	Description
Pertains to Quiz? (Reason)	whether the question is related to the quiz, with a reason provided.
Builds off Last Tutor Response?	whether the question builds upon the last tutor response.
Covered? (Reason)	whether the topic has already been covered in a previous tutor response, with justification.

Table 4. Interaction Flow Categorizations

each of them to synchronize the understanding and reach a consensus on the codes applied. The first iteration was based on participants P1-P3, the second on P4-P6, and the third on the remaining participants. The fourth and final version incorporated refinements identified during the third iteration. We then retrospectively applied the completed codebook to all participant interactions for consistency. Response type saturation was reached midway through analyzing Discord participants.

Debriefing Questions
Can you describe how it felt using the digital tutor to learn the given computer science concept?
To what extent did you trust the tutor? Why?
Could you see yourself using this method for learning new concepts in the future? Why or why not?

Table 5. Standard debriefing questions

Debriefing interviews: After completing the study session, we engaged participants in semi-structured interviews to elicit their experiences and reflections on their tutor and the study session. We asked standard questions across all participants (see Table 5) and adapted follow-up questions based on individual participant actions or responses during the session. We employed member checking to clarify ambiguities in participant responses, increasing the robustness of our analysis.

The interviews were transcribed and the primary researcher conducted the first round of coding. During this phase, the researcher read the text, identified quotes with relationships to our research questions, and labeled the quotes with codes intended to capture their context and intent. After completing the initial coding, the research team convened to deliberate over, rewrite, and reorganize the researcher's codes. The researcher integrated these edits and continued with rounds of coding. In all, the research team met four times, each for two to three hours, to deliberate codes. After reaching stability and code saturation [26], the research team stopped meeting, and the primary researcher analyzed the rest of the debriefing data.

To quantify the detectable effect sizes afforded by our sample, we conducted a sensitivity analysis for 80% power ($\alpha = .05$, two-tailed). For between-group comparisons (independent-samples t), 10 participants per group allow detection of effects no smaller than Cohen's $d \approx 1.32$. For within-group comparisons (paired-samples t), the detectable threshold is Cohen's $d_z \approx 1.00$. Detecting a conventional medium effect (d=0.50) would require roughly 64 participants per group.

All participant quotes and the finalized code book are available in the artifact package[3].

6 RESULTS

Below, we present our findings related to whether ChatGPT-based sessions produce significant learning gains (RQ1), how students perceive ChatGPT as a resource (RQ2), and what strategies they employ when interacting with AI (RQ3).

6.1 Pre-Analysis Data Check

Both groups' pre- and post-intervention quiz scores were unimodal and approximately normally distributed, based on Shapiro–Wilk tests. For the Discord group (n=10), skewness was 0.46 and excess kurtosis was −0.67, with no significant deviation from normality ($W = 0.935$, $p = 0.193$). Similarly, the GPT group (n=10) had a skewness of 0.10 and excess kurtosis of −0.23, also showing no evidence of non-normality ($W = 0.952$, $p = 0.393$). Given that both group distributions satisfied normality assumptions and exhibited skewness and excess kurtosis values comfortably within recommended thresholds for parametric analyses [23], we proceeded with parametric statistical methods.

[3] https://osf.io/mb3du/?view_only=35496af031ab482aa62ca6937c77a32c

6.2 RQ1: Novice Programming Student Quiz Performance After Studying with ChatGPT Versus a Human Tutor over Discord

We analyzed the quiz data from the two groups, those who interacted with ChatGPT (GPT group) and those who interacted with a human via Discord (Discord group), assessing whether the differences between pre-test and post-test scores were statistically significant. We employed a paired t-test, Cohen's d to estimate effect size, and a Bayesian analysis.

The paired t-test for the GPT group indicated a non-significant result (p=0.111). Cohen's d (0.56) suggests a moderate practical effect. Additionally, a Bayesian analysis produced a 95% credible interval of (-0.059, 1.157) for the mean difference and indicated a 96.14% probability that the true effect is positive.

The Discord group's paired t-test yielded a p-value of 0.055, slightly above the conventional 0.05 threshold, indicating a borderline result. Cohen's d ($\approx$ 0.83) suggests that the observed difference between pre-test and post-test scores has a large effect size. The Bayesian analysis produced a 95% credible interval of (0.08, 1.33) for the mean difference and indicated a 98.63% probability that the true effect is positive. Taken together, these complementary methods provide evidence that the improvement in scores is perhaps substantively meaningful.

We compared learning gains by first computing each participant's gain score (post-test - pre-test). Levene's test indicated equal variances (p=.851), so an independent Student's t-test was appropriate; it revealed no significant between-group difference in gain scores (t=-0.34, p=.74). A Bayesian analysis agreed, giving a 95% credible interval of −1.02 to 0.72 and a 36.9% probability that the GPT group's gains exceeded the Discord group's. Thus, we found no evidence of a difference between groups in pre-to-post improvement. When read alongside our qualitative evidence that participants rarely refined their prompts and relied on human tutors for strategic clarification, the null quantitative result implies that chatbots alone are unlikely to boost learning; real gains will require explicit AI-literacy scaffolds and purposeful integration of human support.

Answer to RQ 1

In our lab study quizzes, participants in the GPT group (who had ChatGPT as the tutor) showed no statistically significant difference in performance after studying the computer science topic with ChatGPT. The Discord group participants (who had a human tutor) showed a marginal improvement after the conversation.

6.3 RQ2: Sentiments about ChatGPT as a Learning Resource for Programming Concepts

To assess changes within the pre- and post-intervention survey regarding ease of use, usability, trust, and reliability for each group, we employed the Wilcoxon signed-rank test. This non-parametric test is well-suited for paired or matched data, particularly when the underlying distribution may be non-normal or the data are ordinal, such as Likert-scale survey responses. Because we performed this test separately for multiple questions, we applied a false discovery rate (FDR) correction to control for the increased likelihood of Type I errors due to multiple comparisons. To compare the GPT group and the Discord group, we used the Mann-Whitney U test, another non-parametric method that compares the distribution of ranks between two independent samples. As with the Wilcoxon test, we also applied FDR corrections to the Mann-Whitney U results to ensure a more stringent control of Type I error rates across all survey items tested. These non-parametric methods were chosen to accommodate the relatively small sample size (n=20) and the ordinal nature of the questionnaire data, allowing for a more robust analysis that does not rely on assumptions of normality.

No statistically significant differences were found in our tests. It is important to recognize that these findings do not necessarily indicate an absence of any true effect; rather, with such a modest sample size, we may not have had sufficient power to detect subtle differences or changes. Consequently, our results highlight the possibility that any real differences between the groups might be small and require a larger sample or more sensitive methods to detect (see 5 for a power analysis). To analyze our participants' sentiments about ChatGPT as a learning resource, we turn to their think-aloud and debriefing expressions.

Participants consistently reported that ChatGPT's perceived lack of trustworthiness overshadowed its user-friendly features. They found the tool's output to be of uncertain veracity, forcing them to rely on external resources to verify accuracy. As P3 remarked, *"the only thing that I feel like would help build trust is just making sure that the information is as genuine as it could be."* In fact, most ChatGPT group participants indicated they would not use ChatGPT to learn an entirely new topic (n=6) or would not rely on it exclusively (n=2), whereas half the Discord group—who only knew their tutor had an academic background in computer science and prior tutoring experience—trusted that tutor enough to learn unfamiliar concepts. One Discord participant explained, *"If... I don't know anything about the concept... yeah, yeah, whatever he's saying, he's correct."* This disparity highlights how students' trust depends heavily on whether they feel equipped to evaluate correctness: P5 observed, *"You need knowledge so if you are... well versed with the topic before then you can exactly understand if it's giving you the right thing or not."* Even those open to using ChatGPT for unfamiliar material, like P15, stressed the importance of prioritizing accurate information over quick responses.

Some participants describe significant fear of judgment from human tutors, particularly around asking "silly" or basic questions. Despite their distrust towards ChatGPT because of its outputs, participants displayed more comfort with asking it questions, possibly because they do not feel judged by it. Participants' questions to ChatGPT were largely foundational, with no instances of them qualifying their prior knowledge (n=0). In contrast, those interacting with the human tutor tended to ask more advanced questions, often providing background on their understanding or clarifying their intent (nine instances across six participants). For example, nine of 10 ChatGPT participants asked *"What is a pointer?"* or a close variant as their first question to ChatGPT where participants of the Discord group most often started by asking about the disadvantages and dangers of using pointers first (n=5), which was the only long answer question on the quiz given. P11 expressed a related sentiment during the debriefing interview: *"If we take the example of ChatGPT, we can ask as many questions as needed, and we can ask any kind of questions regarding the subject, even we ask the silly questions. Sometimes, we cannot ask simple and silly questions to the tutor because he may think 'you don't know this... You don't know this small thing... So because of this insecurity, we ask the ChatGPT... easily'"*.

Social anxiety also appeared to be a strong motivator behind preferring AI tools over human interaction. P15 explicitly identified as *"socially awkward"*, describing how the fear of looking *"dumb"* in front of a live tutor led them to avoid in-person sessions entirely. P19 noted a stark difference between typing a question and receiving text replies versus physically standing in front of someone and making eye contact—an emotional barrier that can discourage further inquiries. Digital resources like ChatGPT provided a judgment-free space where students feel comfortable asking questions they might otherwise withhold due to embarrassment. While participants acknowledged that human tutors offer richer explanations and real-world context, the anonymity and low-pressure nature of AI interactions likely reduce social anxieties, allowing learners to experiment, clarify gaps in knowledge, and build confidence in fundamental concepts.

Answer to RQ 2

While our survey analysis did not reveal any statistically significant differences, think-aloud and debrief data indicated that participants distrust ChatGPT's accuracy and use it mainly for low-risk, foundational questions. In contrast, students trusted human tutors for learning new material, yet they hesitated to ask basic questions due to fear of judgment. These findings illustrate how students engage in reflective practice, both reflection in and on action, when interacting with AI versus human tutors. The lack of statistically significant sentiment differences suggests that students are not actively adjusting their engagement strategies (reflection-in-action) in response to ChatGPT's limitations, instead relying on external verification rather than refining their prompts. Additionally, their distrust of ChatGPT's accuracy but comfort in asking it questions points to a limited reflection-on-action, as they do not necessarily use past experiences to refine their AI interactions but rather compartmentalize ChatGPT as a low-risk, low-trust tool.

6.4 RQ3: Strategies Employed When Interacting with ChatGPT

Measure	GPT Group	Discord Group
Number of participants	10	10
Total messages	62	44
Non-factoid questions (NFQs)	37	26
Factoid questions (FQs)	9	9
Commands	10	3
Mixed (questions + commands)	6	6
Top NFQ subcategory	Evidence-based (18)	Evidence-based (9)
Top FQ subcategories	Disjunctive/Concept completion/Quantification (2)	Definition (4)
Top Response Type to NFQs	Context (29) (Evidence-based is close second (27))	Evidence-based (19)
Top Response Types to FQs	Code example/Context (4)	Evidence-based (6)
Avg. # of tag categories per response		
Overall	2.7	1.33
NFQs	2.8	1.4
FQs	3.1	1.64
Avg. words per question	11.1	13
Avg. words per response	295	56.6
Avg. number of questions per participant	≈6 (62 / 10)	≈4 (44 / 10)
Avg. words in questions per participant	66	52
Avg. words in responses per participant	1,770	226.4
Questions related to quiz content	38 of 62	30 of 44
Questions built off tutor's last response	34	26
Questions repeated from previous info	7	6

Table 6. Interaction classification

Participant interaction patterns. To understand the patterns of interaction that participants used to achieve their learning objectives, we applied mixed methods to the logs of their interactions with their tutor. We present the results of this analysis in Table 6.

We conducted independent-sample t-tests to determine if there was a significant difference between the lengths of messages and tutor responses across the GPT and Discord groups. This test was chosen because it compares means between two independent samples to evaluate whether observed differences are statistically significant. Both message length ($p < .001$) and response length ($p < .001$) showed significant differences, with messages to GPT shorter but responses from GPT substantially longer than those to and from the Discord group participants, respectively.

We applied the GPH and Bolotova question-asking taxonomies to the content of both group's questions, open and axial coding to the responses of the tutor, and open and axial coding to the relationship between the questions and responses, such as whether the question asked about material that was already covered or whether it related to the quiz questions (for elaborations of the existing question-asking taxonomies and our code books, please see Section 5).

The ChatGPT group exhibited terse prompts and received large amounts of diverse information. Despite statistically significantly shorter prompts that were, on average, the same type of question that was asked by the Discord group ("evidence-based"), they received responses that were seven times longer and covered almost 1.5 more response categories. As well, the top type of response to NFQs that ChatGPT gave was "context", followed closely by "evidence-based", where the human tutor primarily provided "evidence-based" responses. This pattern implies that ChatGPT spent many more words providing surrounding information and situating each answer in a broader context. Conversely, the human tutor kept explanations relatively succinct and targeted, evidenced by the lower word count and average number of tags. These divergent approaches suggest that while ChatGPT can enrich answers with ancillary details, the human tutor's content was generally more directly aligned with the question. Students who appreciate deeper contextual knowledge may find ChatGPT's style beneficial (P3 and P16 said in debriefing that they like example code without prompting for it), but others may prefer the clarity and brevity that come from more narrowly tailored, "evidence-based" guidance (P12 feels ChatGPT gives a lot of unnecessary information).

In contrast, the Discord group showed a more streamlined exchange where participants asked fewer overall questions and the human tutor responded with succinct, directly targeted explanations. Although more limited in scope—evidenced by fewer total tag categories per response and the top response type to NFQs being "evidence-based"—this style potentially mirrored a typical student–tutor interaction focused on clarifying immediate misunderstandings or directly addressing course objectives. Students seemed to gravitate toward specific, definition-oriented questions and received largely evidence-based replies, suggesting that the human tutor prioritized clarity and concise feedback with their limited study time. This pattern may benefit those who prefer straightforward, easily digestible information, and may explain the borderline increase in quiz scores seen in the Discord group. However, the data also show that GPT's tendency to embed different perspectives or related examples in each response could be valuable for students who want to explore beyond a direct answer, especially if they develop strategies (e.g., refining prompts) to navigate the AI's more verbose outputs effectively.

Additionally, the statistically significant increase in average words per question among the Discord group aligns with prior findings (RQ2), demonstrating that students interacting with the human tutor tended to provide more background or clarifying information. This behavior likely reflects a stronger motivation to supply context—either to reduce misunderstandings or ensure the tutor clearly understands their knowledge gaps. It might also reflect a social motivation, as students may provide more contextualizing information to human tutors to appear less ignorant, a concern mitigated by the social safety of interacting with generative AI. By comparison, the ChatGPT group's shorter prompts—paired with ChatGPT's more expansive responses—suggest these participants place less emphasis on elaborating their inquiries, likely counting on the AI to fill in the gaps. Thus, tutor style and student behaviors appear to reinforce each other: a human tutor's concise, direct approach elicits more precise question-framing from learners,

whereas ChatGPT's context-rich responses encourage students to depend on the system's elaborations rather than providing detailed initial context themselves.

We have established that a key difference between the two groups' interactions was how they phrased their questions and the type of responses they received rather than the questions' intent. From our coding of the think-aloud data and debriefing interviews, we find that our participants have considerable variation in their reactions to ChatGPT's output style. For example, while P3 valued ChatGPT's extensive contextual information, P12 criticized the same verbosity as excessive. Similarly, P2 described ChatGPT as providing superficial explanations without structured pedagogical guidance, likening it to a book that lists information rather than fostering understanding, while simultaneously appreciating code examples that P5 found *"too complex for novices"*.

Despite the diversity of preferences among participants, they rarely attempted to modify the AI's behavior through prompt engineering, demonstrating a lack of reflection-in-action. Instead of viewing ChatGPT's responses as malleable resources that can be shaped through strategic prompting, students tended to consider them as fixed in style, influencing both their engagement with the tool and their satisfaction with its responses. Prompt engineering exemplifies a way to have iterative experimentation and refinement central to reflective practice. Rather than adjusting their approaches when faced with unsatisfactory outputs, most of our participants relied on zero-shot prompting [34, 53], which are isolated questions without contextual scaffolding or specification of learning needs. Furthermore, the variation in reactions to ChatGPT's verbosity and structure suggests reflection-on-action could play a larger role, as students had opinions on the tool's usefulness but did not appear to use these reflections to modify their engagement strategies. This underscores the need for explicit instruction in AI literacy to help students develop metacognitive awareness and more strategic, adaptive approaches to AI-assisted learning.

A few exceptions highlight the potential for more sophisticated engagement. For example, P8's iterative prompting demonstrates a form of reflection-in-action by requesting a *"simple example, start from basics"* followed by a *"sample output"*. These infrequent cases underscore the missed opportunities for most participants to engage in the type of continuous experimentation and reflection that Schön identifies as essential for developing expertise. Additionally, the contrast between ChatGPT and human tutor interactions illustrates different modes of reflective practice. We could observe that students communicating with human tutors demonstrated greater intentionality and contextual framing in their questions. For example, P13, who studied with a human tutor, constructed a complex inquiry about pointers by comparing their difficulty to arrays: *"I believe using pointers is more difficult than using arrays, but even though we use pointers in real-time problems as because they are more like accessing the address of a variable, but why it is difficult compare to arrays?"* Similarly, P18 explicitly sought instructional strategies: *"What methods do you think we should implement in order to learn pointer[s]?"*. The differences in these interactions suggest that students perceived human tutors as adaptable partners in the reflection-in-action process, whereas they viewed ChatGPT's outputs as predetermined and static.

Answer to RQ 3

Although both groups asked similar question types, those interacting with ChatGPT tended to use zero-shot prompts and received significantly longer, more varied responses, yet rarely refined their prompts or attempted to modify ChatGPT's behavior. In contrast, students interacting with human tutors engaged in more reflective question-asking—providing context, seeking strategies, and actively adjusting their queries. Only one participant demonstrated iterative engagement with ChatGPT, evidencing the missed potential for deeper, more adaptive engagement. These findings

suggest that effective chatbot use in programming education requires reflective practice, particularly reflection-in-action, where students iteratively refine their prompts rather than treating AI responses as static. The lack of prompt engineering among participants indicates missed opportunities for adaptive engagement, highlighting the need for structured interventions that encourage students to experiment with different questioning strategies. Additionally, reflection-on-action could be better leveraged by helping students analyze their AI interactions, refine their inquiry approaches, and integrate chatbot use into a broader, strategic learning process. Overall, embedding AI literacy and metacognitive strategies into education can help students transition from passive consumers to active co-constructors of knowledge in AI-mediated learning environments.

7 DISCUSSION

Our findings indicate a tension in how participants relate to human tutors versus AI-based chatbots like ChatGPT. On one hand, students generally trust the correctness of responses from human tutors, but they may hesitate to pose "basic" or "silly" questions due to fear of judgment or perceived social threat. On the other hand, despite recognizing ChatGPT as prone to inaccuracies, participants felt more at ease asking it foundational questions precisely because it is not human, equating to no perceived risk of embarrassment or negative evaluation. This dichotomy between high-trust, high-anxiety interactions with human tutors and low-trust, low-anxiety interactions with generative AI tools points us towards important considerations for designing GenAI chatbots for programming pedagogy with attention to how students reflect on their interactions with chatbots in learning spaces that are both technologically and socially mediated.

Adding nuance to these findings, novice programming students in our study demonstrated minimal prompt engineering—a key competency often referred to as "programmability" in AI literacy [38]. While effective prompting can elicit clearer, more tailored outputs from large language models, students tend to view the chatbot's responses as fixed rather than adjustable through iterative instructions. Prompt engineering is currently the most effective way that LLM-based chatbot users can improve the quality of inference outputs they receive [6, 34], necessitating users' awareness of the process [54] thorough reflection on action, granting increased efficiency as well as increasing trust in chatbots. Without recognizing that their prompts can shape ChatGPT's behavior, learners remain passive, attributing poor responses to the AI's inherent flaws rather than to a fixable breakdown in communication.

Participants' lack of prompt engineering reflects a broader issue of limited, goal-driven engagement with AI tools. Students' limited attempts to modify ChatGPT's behavior suggest an implicit acceptance of AI-generated content as static rather than something they can influence through strategic interaction. This lack of reflective practice, i.e., understanding the affordances and constraints of learning supports and using these judiciously, prevents students from fully leveraging AI's adaptability and limits their ability to position themselves as active learners in AI-mediated learning. Encouraging reflective engagement with AI tools and greater AI literacy could foster greater agency in students' learning processes. For example, structured interventions can prompt students to refine their queries iteratively, considering what additional information or context might enhance the output, which could shift them from passive consumers to active co-constructors of knowledge.

Ensuring that students (and all users) receive useful inference output from AI tools requires making prompt engineering more accessible. Pedagogical chatbot developers can enhance usability by integrating next-generation tools that optimize prompts, reducing the need for users to manually refine them. Long et al. highlighted AI Literacy challenges

before the widespread adoption of GenAI in late 2022, *"it is important for designers to keep in mind that prerequisite coding skills can be a barrier to entry"* to programming AI. While AI interaction has since shifted toward natural language, their argument remains relevant. Prompt engineering is the successor to manipulating AI through coding skill and, like early programming languages, it produces fragile artifacts meant to give instructions to particular hardware. In the case of prompts, they are fragile because of their specificity to a given model, sensitivity to formatting [54], and sensitivity to phrasing [15].

These factors can dramatically affect the quality of outputs and effective prompting methods are largely not discoverable per language model. For example, there is no popular means, such as documentation, for learning how to prompt most effectively for a given model and it is only recently that tooling has appeared for prompt engineering.[4] If prompts are our code in this metaphor, LLMs must be our hardware. Unlike hardware, though, LLMs are black boxes because they are not transparent or deterministic [42], making them unpredictable and unreliable. Like Assembly, though, prompts may be abstracted away by higher-level methods of specifying instructions, such as DSPy [33]. Chatbot developers can lower the barrier to student learning efficacy by wielding next-generation tools that optimize prompts for a given model, abstracting prompt engineering away from the user and no longer forcing them to communicate so directly with the model.

A further consideration is GenAI's anthropomorphism by students. When chatbots project more human-like qualities, students often find them easier to trust, given their familiarity with human conversation cues. At the same time, however, "acting human" may reintroduce the very social anxieties learners aim to avoid. Students' reflection on action might shift from prompt engineering and accuracy to positioning themselves as experts through the prompts. Our results suggest that the chatbot's perceived non-human nature, which frees students to ask questions without fear of judgment, is one of its biggest advantages. Thus, designers and educators face a balancing act: incorporating enough human-like features to foster trust and relational rapport, while preserving the "safe space" dimension that encourages uninhibited inquiry. Future research should investigate which specific chatbot social characteristics [8] can enhance learning outcomes while maintaining the psychological safety that facilitates open inquiry.

Overall, trust, social comfort, AI literacy, and the role of anthropomorphism—demands a multifaceted approach. Teaching prompt engineering in intuitive and streamlined ways can bolster learner confidence, while interface innovations may help students navigate the black box of LLMs. Meanwhile, thoughtful design of chatbot personas must balance fostering trust in AI outputs and capitalizing on the chatbot's unique capacity to mitigate social fears. By simultaneously addressing these practical and psychological factors, we can move toward AI-based learning environments that are both trustworthy and socially comfortable.

8 LIMITATIONS

Our study presents several methodological limitations that should be considered when interpreting our findings.

Our recruitment method relied on volunteer participation, which may have introduced self-selection bias. Students who chose to participate might have had specific interests or prior experiences with AI-based tools, which could influence their responses and engagement. To reduce this problem, we avoided mentioning AI in the recruitment materials. Additionally, since many participants were recruited through snowball sampling, some may have discussed the study with peers beforehand, potentially shaping their expectations and behavior.

[4] https://ide.x.ai

Participants were primarily self-selected volunteers from a single U.S. public university, with the sample comprising 18 men and 2 women, 17 non-native English speakers, and 18 computer science or computer engineering majors (with 2 from other disciplines). This overrepresentation of male, ESL speakers from technical fields limits the generalizability of our findings to broader CS1 student populations and to institutions with different instructional cultures and student demographics. While we aimed to mitigate recruitment bias, future research should include larger, stratified samples across multiple sites to assess whether the patterns observed here hold in more diverse settings.

Some participants may have been familiar with the studied topic in class. To reduce this risk, we chose a topic that had not been taught at the time of the data collection. Still, although we collected baseline knowledge data before the study session, prior exposure to the content may have affected how students engaged with the learning task.

During the study, students interacted with ChatGPT or a human tutor in a controlled setting. While allowing for systematic comparison, the controlled laboratory setting created an artificial learning environment that may not reflect authentic student behaviors. In natural learning contexts, students typically access multiple resources simultaneously and engage with tools like ChatGPT under different time pressures and motivational conditions.

We used a custom, unvalidated instrument specifically to gather preliminary data on change in knowledge and sentiment, recognizing that such an approach is appropriate for an initial, exploratory study. However, because our measures have not undergone extensive psychometric validation, the reliability and generalizability of our findings may be limited.

A significant proportion of participants were non-native English speakers attempting to learn a challenging programming concept while simultaneously verbalizing their thought processes through the think-aloud protocol. This triple cognitive burden likely affected their questioning strategies and engagement depth, potentially limiting the natural flow of their learning interactions. Nevertheless, such effects applied equally to both experimental groups.

Another factor to consider is the potential for unobserved behaviors. Although participants were asked not to use external resources during the study, we cannot rule out the possibility that some did since the study was conducted via teleconference. Anyway, since our goal was not to test their performance but rather to understand their learning interactions, we did not impose strict monitoring.

Our findings represent a specific snapshot of novice programmers' interactions with ChatGPT using the GPT-3.5 model available during data collection. The rapid evolution of LLM capabilities means that student interaction patterns may differ with newer models offering enhanced pedagogical features. Additionally, our focus on pointers in C programming may not generalize to other programming concepts or languages.

Finally, differences in how students approached their interactions with ChatGPT versus a human tutor could reflect personal preferences, prior experiences, or comfort levels rather than the effectiveness of the tool itself. Some students expressed hesitation in asking human tutors basic questions, which may have led to different engagement patterns between the two groups. While our findings highlight these tendencies, they should be interpreted with an understanding that individual learning styles and perceptions play a significant role. The study's small sample size (n=20) limits our ability to control for or systematically analyze these individual differences, which significantly influence tool selection and usage patterns in educational contexts.

These limitations suggest opportunities for future work, including longitudinal studies in authentic learning environments, investigations with larger and more diverse participant samples, using standardized scales to measure change in knowledge and sentiment, and comparative analyses across different programming concepts and LLM implementations.

9 CONCLUSION

In this research, we investigated *what factors influence novice CS students' decisions to use ChatGPT over other learning resources and how these factors alter their resource selection process, how novice CS students perceive ChatGPT's pedagogical methods compared to other learning resources*, and *what strategies novice CS students employ when interacting with ChatGPT to study programming concepts.*

We found that students have a robust learning resource selection process influenced by many factors and that ChatGPT does not subvert this process but is subsumed by it. We also found that novice CS students do not currently perceive ChatGPT as suitable for studying new concepts and found behaviors that an LLM-based chatbot should employ to become more useful to them for this and other learning tasks. We also observed that students leverage LLM chabots accessibility for quick inquiries but they hesitate to rely on it for deeper conceptual learning. This hesitancy reflects a limited engagement in reflection-in-action, where students rarely refine their prompts iteratively, instead treating ChatGPT's responses as static outputs rather than adaptable learning aids.

From these results, we offered advice to students, instructors, and chatbot developers about how to interact with LLM-based chatbots, support their students in learning about and through such tools, and develop pedagogical chatbots to appeal to instructors and support students, respectively.

ACKNOWLEDGMENTS

This work was supported by grants 2236198, 2303042, 2235601, and 2303043 from the National Science Foundation. We thank the students who participated in this study for sharing their time, allowing us to observe them, and for patiently answering our questions.

REFERENCES

[1] Rishabh Balse, Viraj Kumar, Prajish Prasad, and Jayakrishnan Madathil Warriem. 2023. Evaluating the Quality of LLM-Generated Explanations for Logical Errors in CS1 Student Programs. In *Proceedings of the 16th Annual ACM India Compute Conference* (Hyderabad, India) *(COMPUTE '23)*. Association for Computing Machinery, New York, NY, USA, 49–54. https://doi.org/10.1145/3627217.3627233

[2] Brett A Becker and Thomas Fitzpatrick. 2019. What do cs1 syllabi reveal about our expectations of introductory programming students?. In *Proceedings of the 50th ACM technical symposium on computer science education*. 1011–1017.

[3] Jens Bennedsen and Michael E. Caspersen. 2019. Failure rates in introductory programming: 12 years later. *ACM Inroads* 10, 2 (apr 2019), 30–36. https://doi.org/10.1145/3324888

[4] Valeriia Bolotova, Vladislav Blinov, Falk Scholer, W. Bruce Croft, and Mark Sanderson. 2022. A Non-Factoid Question-Answering Taxonomy. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 1196–1207. https://doi.org/10.1145/3477495.3531926

[5] Chris Bopp, Anne Foerst, and Brian Kellogg. 2024. The Case for LLM Workshops. 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, 130–136. https://doi.org/10.1145/3626252.3630941

[6] Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).

[7] Chen Cao. 2023. Leveraging Large Language Model and Story-Based Gamification in Intelligent Tutoring System to Scaffold Introductory Programming Courses: A Design-Based Research Study. *arXiv preprint arXiv:2302.12834* (2023).

[8] Ana Paula Chaves and Marco Aurelio Gerosa. 2021. How should my chatbot interact? A survey on social characteristics in human–chatbot interaction design. *International Journal of Human–Computer Interaction* 37, 8 (2021), 729–758.

[9] Rudrajit Choudhuri, Dylan Liu, Igor Steinmacher, Marco Gerosa, and Anita Sarma. 2024. How far are we? the triumphs and trials of generative ai in learning software engineering. In *Proceedings of the IEEE/ACM 46th international conference on software engineering*. 1–13.

[10] Juliet Corbin and Anselm Strauss. 2014. *Basics of qualitative research: Techniques and procedures for developing grounded theory.* Sage publications.

[11] Paul E Dickson and John Barr. 2019. Bringing Reflection into Computer Science Education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 1249–1249.

[12] Virginia R Downing, Katelyn M Cooper, Jacqueline M Cala, Logan E Gin, and Sara E Brownell. 2020. Fear of negative evaluation and student anxiety in community college active-learning science courses. *CBE—life sciences education* 19, 2 (2020), ar20.

[13] Yrjö Engeström. 1999. Expansive visibilization of work: An activity-theoretical perspective. *Computer Supported Cooperative Work (CSCW)* 8 (1999), 63–93.

[14] K Anders Ericsson. 2017. Protocol analysis. *A companion to cognitive science* (2017), 425–432.

[15] Federico Errica, Giuseppe Siracusano, Davide Sanvito, and Roberto Bifulco. 2024. What Did I Do Wrong? Quantifying LLMs' Sensitivity and Consistency to Prompt Engineering. arXiv:2406.12334 [cs.LG] https://arxiv.org/abs/2406.12334

[16] James Finnie-Ansley, Paul Denny, Brett A Becker, Andrew Luxton-Reilly, and James Prather. 2022. The robots are coming: Exploring the implications of openai codex on introductory programming. In *Proceedings of the 24th Australasian computing education conference*. 10–19.

[17] James Finnie-Ansley, Paul Denny, Andrew Luxton-Reilly, Eddie Antonio Santos, James Prather, and Brett A. Becker. 2023. My AI Wants to Know if This Will Be on the Exam: Testing OpenAI's Codex on CS2 Programming Exercises. In *Proceedings of the 25th Australasian Computing Education Conference* (Melbourne, VIC, Australia) *(ACE '23)*. Association for Computing Machinery, New York, NY, USA, 97–104. https://doi.org/10.1145/3576123.3576134

[18] Gili Freedman, Melanie C Green, Mia Kussman, Mason Drusano, and Melissa M Moore. 2023. "Dear future woman of STEM": letters of advice from women in STEM. *International Journal of STEM Education* 10, 1 (2023), 20.

[19] Natasha Freidus and Michelle Hlubinka. 2002. Digital storytelling for reflective practice in communities of learners. *SIGGROUP Bull.* 23, 2 (Aug. 2002), 24–26. https://doi.org/10.1145/962185.962195

[20] Francisco José García-Peñalvo. 2018. Editorial computational thinking. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje* 13, 1 (2018), 17–19.

[21] Francisco J García-Peñalvo, Daniela Reimann, Maire Tuul, Angela Rees, Ilkka Jormanainen, et al. 2016. An overview of the most relevant literature on coding and computational thinking with emphasis on the relevant issues for teachers. (2016).

[22] Anupam Garg, Aryaman Raina, Aryan Gupta, Jaskaran Singh, Manav Saini, Prachi Iiitd, Ronit Mehta, Rupin Oberoi, Sachin Sharma, Samyak Jain, et al. 2024. Analyzing LLM usage in an advanced computing class in India. *arXiv preprint arXiv:2404.04603* (2024).

[23] Darren George and Paul Mallery. 2019. *IBM SPSS statistics 25 step by step: A simple guide and reference*. Routledge.

[24] Arthur C Graesser and Natalie K Person. 1994. Question asking during tutoring. *American educational research journal* 31, 1 (1994), 104–137.

[25] Philipp Haindl and Gerald Weinberger. 2024. Students' Experiences of Using ChatGPT in an Undergraduate Programming Course. *IEEE Access* 12 (2024), 43519–43529. https://doi.org/10.1109/ACCESS.2024.3380909

[26] Monique M Hennink, Bonnie N Kaiser, and Vincent C Marconi. 2017. Code saturation versus meaning saturation: how many interviews are enough? *Qualitative health research* 27, 4 (2017), 591–608.

[27] Muntasir Hoq, Yang Shi, Juho Leinonen, Damilola Babalola, Collin Lynch, Thomas Price, and Bita Akram. 2024. Detecting ChatGPT-Generated Code Submissions in a CS1 Course Using Machine Learning Models. 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, 526–532. https://doi.org/10.1145/3626252.3630826

[28] Irene Hou, Sophia Mettille, 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*. 39–48.

[29] Norio Ishii and Kazuhisa Miwa. 2005. Supporting reflective practice in creativity education. In *Proceedings of the 5th Conference on Creativity & Cognition* (London, United Kingdom) *(C&C '05)*. Association for Computing Machinery, New York, NY, USA, 150–157. https://doi.org/10.1145/1056224.1056246

[30] Sharin R Jacob and Mark Warschauer. 2018. Computational thinking and literacy. *Journal of Computer Science Integration* 1, 1 (2018).

[31] Breanna Jury, Angela Lorusso, Juho Leinonen, Paul Denny, and Andrew Luxton-Reilly. 2024. Evaluating LLM-generated Worked Examples in an Introductory Programming Course. In *Proceedings of the 26th Australasian Computing Education Conference* (Sydney, NSW, Australia) *(ACE '24)*. Association for Computing Machinery, New York, NY, USA, 77–86. https://doi.org/10.1145/3636243.3636252

[32] Fariza Khalid, Mazalah Ahmad, Aidah Abdul Karim, Md Yusoff Daud, and Rosseni Din. 2015. Reflective thinking: An analysis of students' reflections in their learning about computers in education. *Creative Education* 6, 20 (2015), 2160–2168.

[33] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. arXiv:2310.03714 [cs.CL] https://arxiv.org/abs/2310.03714

[34] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. arXiv:2205.11916 [cs.CL] https://arxiv.org/abs/2205.11916

[35] Sam Lau and Philip Guo. 2023. From" Ban it till we understand it" to" Resistance is futile": How university programming instructors plan to adapt as more students use AI code generation and explanation tools such as ChatGPT and GitHub Copilot. In *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1*. 106–121.

[36] Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. 2023. 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* (Turku, Finland) *(ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 124–130. https://doi.org/10.1145/3587102.3588785

[37] 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.

[38] Duri Long and Brian Magerko. 2020. What is AI literacy? Competencies and design considerations. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–16.

[39] Andrew Luxton-Reilly. 2016. Learning to Program is Easy. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (Arequipa, Peru) *(ITiCSE '16)*. Association for Computing Machinery, New York, NY, USA, 284–289. https://doi.org/10.1145/2899415.2899432

[40] Seyed Mehdi Nasehi, Jonathan Sillito, Frank Maurer, and Chris Burns. 2012. What makes a good code example?: A study of programming Q&A in StackOverflow. In *2012 28th IEEE International Conference on Software Maintenance (ICSM)*. IEEE, 25–34.

[41] Abdessalam Ouaazki, Kristoffer Bergram, Juan Carlos Farah, Denis Gillet, and Adrian Holzer. 2024. Generative AI-Enabled Conversational Interaction to Support Self-Directed Learning Experiences in Transversal Computational Thinking. In *Proceedings of the 6th ACM Conference on Conversational User Interfaces*. 1–12.

[42] Shuyin Ouyang, Jie M. Zhang, Mark Harman, and Meng Wang. 2024. An Empirical Study of the Non-determinism of ChatGPT in Code Generation. *ACM Transactions on Software Engineering and Methodology* (Sept. 2024). https://doi.org/10.1145/3697010

[43] Andrew Petersen, Michelle Craig, Jennifer Campbell, and Anya Tafliovich. 2016. Revisiting why students drop CS1. In *KOLI-CALLING* (Koli, Finland) *(Koli Calling '16)*. Association for Computing Machinery, New York, NY, USA, 71–80. https://doi.org/10.1145/2999541.2999552

[44] Tung Phung, Victor-Alexandru Pădurean, Jos é Cambronero, Sumit Gulwani, Tobias Kohn, Rupak Majumdar, Adish Singla, and Gustavo Soares. 2023. Generative AI for Programming Education: Benchmarking ChatGPT, GPT-4, and Human Tutors. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 2* (Chicago, IL, USA) *(ICER '23)*. Association for Computing Machinery, New York, NY, USA, 41–42. https://doi.org/10.1145/3568812.3603476

[45] James Prather, Juho Leinonen, Natalie Kiesler, Jamie Gorson Benario, Sam Lau, Stephen MacNeil, Narges Norouzi, Simone Opel, Vee Pettit, Leo Porter, et al. 2024. Beyond the Hype: A Comprehensive Review of Current Trends in Generative AI Research, Teaching Practices, and Tools. *arXiv preprint arXiv:2412.14732* (2024).

[46] James Prather, Brent N Reeves, Paul Denny, Brett A Becker, Juho Leinonen, Andrew Luxton-Reilly, Garrett Powell, James Finnie-Ansley, and Eddie Antonio Santos. 2023. "It's weird that it knows what i want": Usability and interactions with copilot for novice programmers. *ACM transactions on computer-human interaction* 31, 1 (2023), 1–31.

[47] Marc Prensky. 2008. Programming is the new literacy. *Edutopia magazine* (2008).

[48] Adnan Qayyum. 2018. Student help-seeking attitudes and behaviors in a digital era. *International Journal of Educational Technology in Higher Education* 15, 1 (2018), 1–16.

[49] Philip M Reeves and Rayne A Sperling. 2015. A comparison of technologically mediated and face-to-face help-seeking sources. *British Journal of Educational Psychology* 85, 4 (2015), 570–584.

[50] Michael P. Rogers, Hannah Miller Hillberg, and Christopher L. Groves. 2024. Attitudes Towards the Use (and Misuse) of ChatGPT: A Preliminary Study. 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, 1147–1153. https://doi.org/10.1145/3626252.3630784

[51] Allison M Ryan and Sungok Serena Shim. 2012. Changes in help seeking from peers during early adolescence: Associations with changes in achievement and perceptions of teachers. *Journal of educational psychology* 104, 4 (2012), 1122.

[52] Donald A Schon. 2008. *The reflective practitioner: How professionals think in action*. Basic books.

[53] Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. 2024. The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608* (2024).

[54] Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting. arXiv:2310.11324 [cs.CL] https://arxiv.org/abs/2310.11324

[55] 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* (Sydney, NSW, Australia) *(ACE '24)*. Association for Computing Machinery, New York, NY, USA, 49–57. https://doi.org/10.1145/3636243.3636249

[56] Marita Skjuve, Asbjørn Følstad, and Petter Bae Brandtzaeg. 2023. The user experience of ChatGPT: Findings from a questionnaire study of early users. In *Proceedings of the 5th international conference on conversational user interfaces*. 1–10.

[57] Lev Semenovich Vygotsky and Michael Cole. 1978. *Mind in society: Development of higher psychological processes*. Harvard university press.

[58] Mengqiu Wang et al. 2006. A survey of answer extraction techniques in factoid question answering. *Computational Linguistics* 1, 1 (2006), 1–14.

[59] Yuankai Xue, Hanlin Chen, Gina R. Bai, Robert Tairas, and Yu Huang. 2024. Does ChatGPT Help With Introductory Programming?An Experiment of Students Using ChatGPT in CS1. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training* (Lisbon, Portugal) *(ICSE-SEET '24)*. Association for Computing Machinery, New York, NY, USA, 331–341. https://doi.org/10.1145/3639474.3640076

[60] Lysann Zander and Elisabeth Höhne. 2021. Perceived peer exclusion as predictor of students' help-seeking strategies in higher education. *Zeitschrift für Entwicklungspsychologie und Pädagogische Psychologie* (2021).

[61] Barry J Zimmerman. 2002. Becoming a self-regulated learner: An overview. *Theory into practice* 41, 2 (2002), 64–70.