Exploring Teacher-Chatbot Interaction and Affect in Block-Based Programming

Bahare Riahi
Computer Science
North Carolina State University
Raleigh, North Carolina, USA
Briahi@ncsu.edu

Ally Limke
Computer Science
North Carolina State University
Raleigh, North Carolina, USA
anlimke@ncsu.edu

Xiaoyi Tian
Department of Computer Science
North Carolina State University
Raleigh, North Carolina, USA
xtian9@ncsu.edu

Viktoriia Storozhevykh
Computer Science
North Carolina State University
Raleigh, North Carolina, USA
vstoroz@ncsu.edu

Sayali Patukale
Computer Science
North Carolina State University
Raleigh, North Carolina, USA
spatuka@ncsu.edu

Tahreem Yasir
Computer Science
North Carolina State University
Raleigh, North Carolina, USA
tyasir@ncsu.edu

Khushbu Singh
School of Education and Human Development
University of Virginia
Charlottesville, Virginia, USA
dcf2rk@virginia.edu

Jennifer Chiu
School of Education and Human Development
University of Virginia
Charlottesville, Virginia, USA
jlc4dz@virginia.edu

Nicholas lytle
Computer Science
Georgia Institute of Technology
Atlanta, Georgia, USA
nlytle3@gatech.edu

Tiffany Barnes
Computer Science
North Carolina State University
Raleigh, North Carolina, USA
tmbarnes@ncsu.edu

Veronica Catete
Computer Science
North Carolina State University
Raleigh, North Carolina, USA
vmcatete@ncsu.edu

Abstract

AI-based chatbots have the potential to accelerate learning and teaching, but may also have counterproductive consequences without thoughtful design and scaffolding. To better understand teachers' perspectives on large language model (LLM) based chatbots, we conducted a study with 11 teams of middle-school teachers using chatbots for a science and computational thinking activity within a block-based programming environment. Based on a qualitative analysis of audio transcripts and chatbot interactions, we propose three profiles: explorer, frustrated, and mixed that reflect diverse scaffolding needs. In their discussions, we found that teachers perceived chatbot benefits such as building prompting skills and self confidence alongside risks including potential declines in learning and critical thinking. Key design recommendations include scaffolding the introduction to chatbots, facilitating teacher control of chatbot features, and suggesting when and how chatbots should be used. Our contribution informs the design of chatbots to support teachers and learners in middle school coding activities.

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/26/04
https://doi.org/10.1145/3772318.3791823

CCS Concepts

• **K-12 Education**; • **Human Computer Interaction**; • **Natural Language Generation**;

Keywords

ChatBots, Teacher Professional Development, Computational Thinking, Large Language Models

ACM Reference Format:
Bahare Riahi, Ally Limke, Xiaoyi Tian, Viktoriia Storozhevykh, Sayali Patukale, Tahreem Yasir, Khushbu Singh, Jennifer Chiu, Nicholas lytle, Tiffany Barnes, and Veronica Catete. 2026. Exploring Teacher-Chatbot Interaction and Affect in Block-Based Programming. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3772318.3791823

1 Introduction

While integrating Artificial Intelligence (AI) applications in K-12 education has had a long and evolving history, the introduction of ChatGPT and other Large Language Model (LLM) powered technologies in 2022 accelerated the adoption of these tools [26, 43]. This rapid proliferation has created a litany of pedagogical and ethical concerns about problematic AI-usage in the classroom, with many stemming from uncertainty about how much of graded material (e.g., homework, assignments, exam answers) was completed by these LLMs rather than by the students themselves [29]. Products designed to detect LLM-generated text are not without error [79], and there are inconsistencies at the school, course, and teacher-level about what types of LLM interactions constitute plagiarism [58]. For instance, a student generating small sections of code or test cases for a larger coding project may or may not be seen as acceptable. Additionally, most of these AI-interactions are student-self directed and in environments outside of either the visibility or control of the teacher.

This type of use and misuse of AI in classrooms can potentially have impacts on learning outcomes. While our understanding of these effects are still evolving, early evidence points to the idea that while unrestricted student use can lead to faster and more accurate completion of activities in the short term, there can be long-term learning consequences. Research points to the idea that offloading cognitive effort to an AI may reduce the mental struggle necessary for deep learning, particularly in domains like programming where "productive struggle" is a key component of skill acquisition [14]. When students rely on an LLM to generate code, they may bypass the critical thinking and problem-solving processes that lead to lasting understanding [27]. Companies are responding to these challenges by creating LLM systems targeted towards more effective pedagogical practices (e.g. ChatGPTs StudyMode [60]). **Professional development workshops** will allow researchers and chatbot designers to capture the range of concerns teachers have about unwanted interactions with the chatbot as well as set standards for desired AI behaviors for both student-AI and teacher-AI interactions. Additionally, the range of learning environments and activities where these chatbots can be employed and differences in teachers' pedagogical style suggest that a 'one-size-fits-all' system with no instructor control over behavior is unacceptable. These complexities highlight the need for psychological frameworks such as Technology Acceptance Models (TAM) [19] and Theory of Planned Behavior (TPB) [2] to explain teachers' evaluations and adoption decisions, and we mapped our analysis to these frameworks to interpret how their emotions, perceptions, and sense of control shaped their intention to use the chatbot.

In the summer of 2025, we invited 25 middle school science teachers to participate in a professional development focused on integrating computational thinking (CT) and block-based programming activities into their science classrooms. As part of this training, we included a module in which teachers completed a lesson within a block-based environment that had an integrated 'chatbot' component. This chatbot had a range of features including the generation of code from written description as well as the ability to answer questions regarding programming concepts generally or questions related to how to use the block-based environment specifically. Our aim was to use data collected from both these learning modules and a set of follow up discussions with the teachers to guide our understanding of how teachers interact with these types of technologies in practice, and how teachers would want a more formally developed AI system to operate. This aim is characterized by the following research questions:

Research Questions:

(1) RQ1. How does interacting with chatbots for learning programming impact middle school teachers' affect and attitudes?
(2) RQ2. What are teachers' perspectives about the benefits and risks of using an LLM for block-based programming in their classrooms for students and for teachers?

2 Related Work

2.1 How Teachers Interact with LLMs in Pedagogical Environments

Kim et al. (2024) conceptualized teacher AI interactions as cognitive, socio-emotional, and artifact-mediated [46]. Cognitive interactions involve teachers' reasoning about AI outputs, selecting or modifying tasks, and making instructional decisions. Socio-emotional interactions reflect the trust, confidence, and attitudes of teachers toward AI, influencing when and how they use AI suggestions. Artifact-mediated interactions occur when teachers engage iteratively with AI-generated outputs, treating them as artifacts to support lesson design, instructional decision-making, and student learning. This framework, applied to various AI tools and classroom dashboards, can also be observed in studies examining the use of LLMs, such as ChatGPT.

Holstein et al. (2018) evaluated Lumilo, a wearable mixed reality tool for K-12 teachers, providing real-time insights into student learning and behavior [33]. Teachers engaged cognitively by planning interventions, socio-emotionally by evaluating trust and usability, and artifact-mediated through learning analytics. Lumilo also highlighted challenges with LLM use, such as information overload, anxiety, and the need to balance interpretability with accuracy.

Across these studies, teachers' interactions with LLMs and AI tools reflect cognitive, socio-emotional, and artifact-mediated dimensions. Teachers use AI outputs to reason, adapt, or reject; navigate trust in AI; and iteratively interact with AI-generated artifacts to shape instruction. They prioritize maintaining agency, seeking tools that reduce burdens while preserving pedagogical control and ethical responsibility [24, 82]. Overall, effective teacher–LLM interactions are best seen as collaborative partnerships where AI augments, not replaces, teacher expertise. LLMs and generative AI have quickly entered schools, generating excitement and concerns. Teachers view ChatGPT as a valuable tool for saving time and supporting computer science lesson planning [62], but also recognize challenges like academic integrity, student privacy, AI reliability, school restrictions, and ChatGPT's age requirement, which complicate classroom integration.

Teachers' confidence and competencies significantly impact their adoption of LLM tools [84]. For instance, Reichert et al. described a workshop for secondary teachers on ChatGPT that improved understanding and increased positive attitudes from 45% to 68% [65]. A survey of 102 STEM teachers in Germany found that teacher competence was the strongest predictor of ChatGPT use, with future use driven by perceived teaching benefits. Concerns about privacy, copyright, and reliability had minimal impact, suggesting that professional development and competence-building shape teachers' expectations for LLM design and use [8].

Key design considerations include supporting active learning, providing ethical guidance, facilitating prompt engineering, offering structured materials, overcoming school/age barriers, ensuring continuous support, and fostering critical thinking [83]. By addressing misconceptions, offering structured guidance, and building teacher competencies, AI can become a transformative tool for educators. Insights from a design-based research (DBR) study on human-centered AI in STEM classrooms highlight teachers' perspectives on AI's role in managing collaborative learning and scaffolding instruction [69]. Teachers valued AI that could reduce routine workload, offer structured guidance, and enhance classroom management without compromising their professional judgment [52]. Teachers expected AI to complement their expertise, provide timely support, and seamlessly integrate into workflows. Their expectations for autonomy, control, and practical support are crucial for designing AI tools that are both useful and adoptable [7].

Collectively, these findings suggest that teachers' perceptions, confidence, and expectations are critical for the successful adoption of AI [12]. LLMs and other AI tools should be designed in accordance with human-centered AI principles, aligning with teachers' pedagogical expertise, workflows, and classroom realities. Such tools should empower teachers by augmenting their professional judgment, providing structured guidance and timely support, and enabling creativity and active engagement, while maintaining human control and accountability in the learning process [74, 80].

2.2 Design/Use of LLM in Secondary and Coding Classrooms

In recent years, several professional integrated development environments (IDEs), such as GitHub Co-Pilot, Cursor, and Claude, have integrated LLM technology to assist developers. However, less attention has been given to IDEs for classrooms, which require additional design considerations. Secondary classrooms typically use block-based environments like Scratch [54, 66], MIT app inventor [61] and UC Berkeley's Snap! [28] to introduce programming concepts. The type of LLM support needed for students learning programming differs from that for professional developers. As LLM-integrated IDEs are still new, it's unclear what support students need and when to provide it for optimal learning.

Recent studies on large language models (LLMs) in secondary coding classrooms have explored a range of support, from code generation to structured scaffolding. Tools offering code adaptation and stepwise guidance, such as Parsons puzzles, minimal fix units, and next-step hints, yield stronger learning outcomes than tools that generate full solutions [22, 35–37]. These interactive approaches promote active problem solving, extending practice time, improving retention, and reducing over-reliance on automated answers [67]. Personalized scaffolding, tailored to address student errors or prior work, has shown to further enhance engagement and learning outcomes [21]. Feedback systems powered by LLMs can provide instant, precise, and individualized responses, sometimes outperforming human instructors in error detection [48, 73]. Yet, challenges persist, as LLM feedback can be misleading and lacks adaptability to classroom dynamics [30, 68].

For teachers, the effectiveness of LLM-based tools depends on integration into pedagogy and professional development. Educators value systems that encourage reflection and deeper understanding rather than direct answer-giving, often supported by guardrails that prompt students to engage in self-correction [18, 50]. At the same time, design tensions emerge between providing sufficient scaffolding and preserving learner agency. For example, in *Cognimates Scratch Copilot* [23], students leveraged AI suggestions for ideation, debugging, and asset creation but actively adapted or rejected outputs to maintain creative control. These findings highlight that the value of LLMs lies not in their raw generative capability but in their careful design as collaborative learning partners that balance efficiency, accuracy, and student autonomy. However, there is still limited understanding of teacher perspectives and perceptions regarding the use of LLMs in classrooms, with a noticeable gap in detailed evidence on how teachers experience these tools. Specifically, more research is needed to understand the challenges teachers encounter, their perceptions of the benefits and usefulness of AI tools, and the design features they prefer, as these factors strongly influence adoption decisions.

3 Methods

3.1 Participants

Our LLM chatbot study took place during a teacher professional development (PD) workshop on infusing computational thinking and block-based programming into middle grades science classrooms. The professional development had both experienced lead teachers (N=8) and novice participant teachers (N=17) (Table 1) . The lead teachers had previous experience with computing education and had taken part in professional development with our research group before. The participant teachers were newer to computing and hadn't been involved in our professional development programs. The lead teachers did the 1-hour study on day 0, during their pre-PD training, while the participant teachers did it on day 2, with the lead teachers facilitating the session.

3.2 Chatbot-integrated programming environment: stax.fun)

The programming environment we chose to use for this study is *Stax.fun (Figure 1)* offers an AI copilot for block-based coders that can generate, debug, and troubleshoot Scratch programming code through four different prompt modes. The developers describe these modes as *Code tab (Figure 2)*: generate visual blocks using natural language prompts; *Q&A tab (Figure 3)* : get answers, insights, and creative ideas about visual block coding; *Coach tab (Figure 4)* : generate step-by-step guidance on building a project idea from scratch; *Prompts tab (Figure 5)* : generate refined prompts based on the user's prompts for more accurate and creative results. Stax also supports importing and exporting native Scratch projects (.sb3 files). Each participant logged in using a unique student account linked to the researchers' teacher account with our starter code preloaded.

3.3 Study Activities and Tasks

The study involved 11 key activities designed to engage participants with the chatbot, organized into two main phases: *Getting Familiar with the Chatbot* (Steps 1-5) and *Coding Wave Activities* (Steps 6-11).

(1) Getting Familiar with the Chatbot's interface and its various features (Steps 1-5):

Table 1: Demographics of Teachers in LLM study

Type	Gender	Subjects	Skills
Lead: n=8	8 female	6 Sci, 1 Math 1 CS, 1 ELA	8 Scratch+Snap! 7 code.org
Participants:n=17	12 female 5 male	10 Sci, 3 Math 1 CS, 3 other	11 Scratch 3 code.org

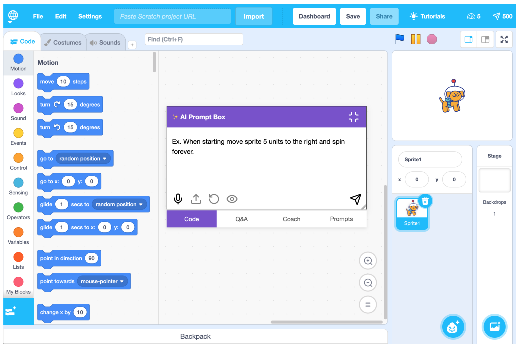
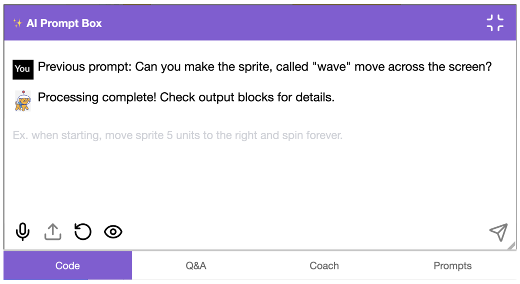


Figure 1: Stax.fun Platform overview and its chatbot tabs and features



You Previous prompt: Can you make the sprite, called "wave" move across the screen?

Processing complete! Check output blocks for details.

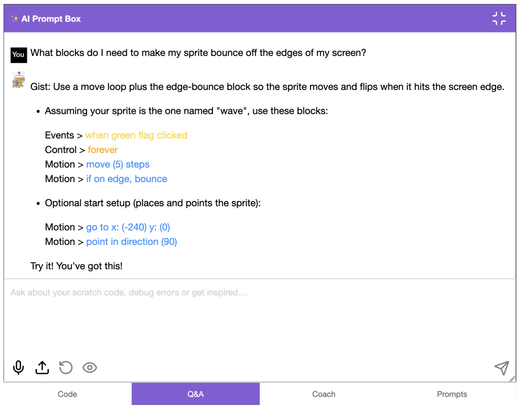
Ex. when starting, move sprite 5 units to the right and spin forever.
Code
Q&A
Coach
Prompts

Figure 2: Code Tab generating code snippets, providing error corrections

- **Step 1-5:** Tasks such as understanding the functionality of the chatbot and familiarizing themselves with different tabs (Code, Coach, Q&A, etc.).

(2) Coding Wave Activities (Wave interactions) (Steps 6-11):

- **Step 6 (Starter Code):** Initial task involving basic coding concepts.
- **Step 7 (Coding - Curtain):** Participants worked on coding tasks related to simulating the interaction of light waves with a curtain.
- **Step 8 (Coding - Glass):** Tasks focused on how light waves interact with glass.
- **Step 9 (Coding - Mirror):** Coding interaction between light waves and mirrors.



You What blocks do I need to make my sprite bounce off the edges of my screen?

Gist: Use a move loop plus the edge-bounce block so the sprite moves and flips when it hits the screen edge.

- Assuming your sprite is the one named "wave", use these blocks:

  Events > when green flag clicked
  Control > forever
  Motion > move (5) steps
  Motion > if on edge, bounce

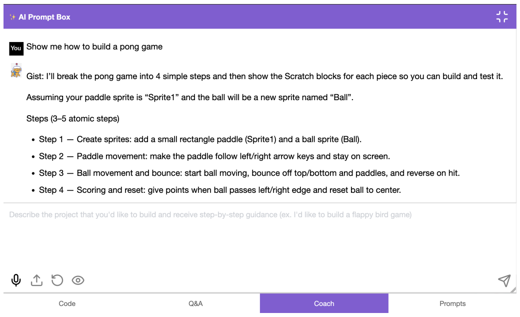
- Optional start setup (places and points the sprite):

  Motion > go to x: (-240) y: (0)
  Motion > point in direction (90)

Try it! You've got this!

Ask about your scratch code, debug errors or get inspired....
Code
Q&A
Coach
Prompts

Figure 3: Q&A Tab offering real-time answers, explanations, or clarifications



You Show me how to build a pong game

Gist: I'll break the pong game into 4 simple steps and then show the Scratch blocks for each piece so you can build and test it.

Assuming your paddle sprite is "Sprite1" and the ball will be a new sprite named "Ball".

Steps (3–5 atomic steps)

- Step 1 — Create sprites: add a small rectangle paddle (Sprite1) and a ball sprite (Ball).
- Step 2 — Paddle movement: make the paddle follow left/right arrow keys and stay on screen.
- Step 3 — Ball movement and bounce: start ball moving, bounce off top/bottom and paddles, and reverse on hit.
- Step 4 — Scoring and reset: give points when ball passes left/right edge and reset ball to center.

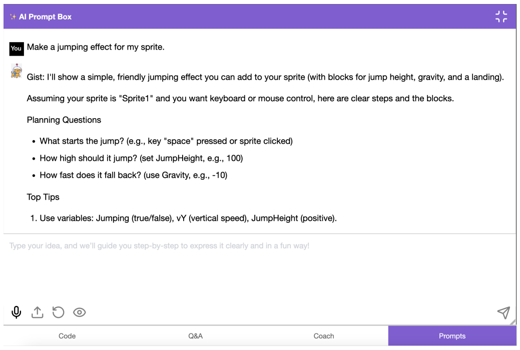
Describe the project that you'd like to build and receive step-by-step guidance (ex. I'd like to build a flappy bird game)
Code
Q&A
Coach
Prompts

Figure 4: Coach Tab offering hints, tips, and suggestions

- **Step 10 (Sound Wave Activity):** Participants applied the same principles to sound waves, coding how sound interacts with various materials.
- **Step 11 (Plastic-bag Activity):** Focused on coding the interaction of light waves with a plastic bag, using a ghost effect to represent partial transmission.

3.4 Procedures

Our data collection was conducted on Day 2, while Day 1 was dedicated to introducing the participants to block-based programming. *Day 1 (Introductory Activity):* Participants were introduced to the block-based programming language Snap!, which would be used



Make a jumping effect for my sprite.

Gist: I'll show a simple, friendly jumping effect you can add to your sprite (with blocks for jump height, gravity, and a landing).

Assuming your sprite is "Sprite1" and you want keyboard or mouse control, here are clear steps and the blocks.

Planning Questions

- What starts the jump? (e.g., key "space" pressed or sprite clicked)
- How high should it jump? (set JumpHeight, e.g., 100)
- How fast does it fall back? (use Gravity, e.g., -10)

Top Tips

1. Use variables: Jumping (true/false), vY (vertical speed), JumpHeight (positive).

Type your idea, and we'll guide you step-by-step to express it clearly and in a fun way!

Figure 5: Prompt Tab refine and structure the prompt to align with the desired outcome

throughout the PD. They then completed an activity in Snap!, which was adapted from a middle school science standard on wave interactions, requiring participants to program the behavior of waves. Slides guided participants step-by-step through the activity and included answer checks. Participants had 1 hour to complete the activity and worked in pairs on their laptops. This day was not part of the data collection procedure; it was only intended as an introduction for the participants. *Day 2 (Wave Activity):* We conducted a 30-minute think-aloud study with teachers in groups of 3-4. Lead teachers did not complete the introductory activity since they were already familiar with it and served as facilitators for the other teachers. During the study, participants were introduced to a large language model (LLM)-powered chatbot for block-based programming in a Scratch IDE, stax.fun. Participants spent approximately 10 minutes learning about and experimenting with the chatbot. They then completed the same programming activity as on Day 1. To facilitate the use of the chatbot, the guided slides were modified to include prompts encouraging participants to engage with the chatbot. Participants were also encouraged to use the chatbot as needed to assist them with the programming tasks. Participants were asked to record their screens and audio using Zoom during this activity.

At the conclusion of the activity, participants were given discussion questions for 15 minutes. Topics included how the chatbot impacted their coding process, their comfort with allowing students to use the chatbot, necessary modifications for classroom use, strategies for scaffolding student use, and their general interest in integrating AI tools into their classrooms (Figure 6).

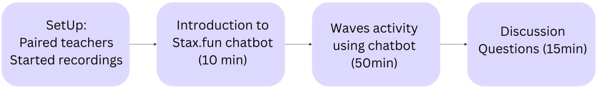


Figure 6: Process of the study with stax.fun chatbot

4 Analysis & Results

We conducted the analysis of the PD sessions data in two phases. The first phase focused on behavior, time stamping and affect and emotion tracking, while the second phase involved hybrid thematic analysis related to discussions and questions.

4.1 Analyzing Behavior, Interaction, and Emotion

We conducted a multi-source, time-anchored analysis of each 50-minute PD session using (a) Zoom video and screen recordings, (b) think-aloud transcripts, and (c) interaction traces (chatbot prompts, revisions, tab switches, and help-seeking). Each video was converted into a minute-level timeline, and using full meeting timestamps, we segmented the recordings into task-based intervals such as coding block-based activities, exploring the chatbot, debugging, addressing connectivity issues, and asking the instructor for help. These segments informed a structured codebook that included activity name, slide reference, timestamp, task description, duration. It also included analytic categories such as emotional responses, comprehension of instructions, interaction patterns with the chatbot, struggles with the interface, and question-asking. Two coders, trained by the lead author, collaboratively coded an initial sample and then independently coded the remaining sessions (each coder responsible for half). After initial annotation, the coders reviewed each session to verify alignment between timestamps, activity labels, and interpretations. Finally, the lead author finalized the tagging for consistency. To ensure reliability, we applied data triangulation by cross-verifying emotional codes across think-aloud transcripts and screen recordings. We first coded emotional expressions in the transcripts based on verbal cues and then compared these with corresponding actions observed in the screen recordings (e.g., hesitation or task changes). Any discrepancies were resolved through discussion among the research team, ensuring consistency and enhancing the reliability of our findings. Triangulation enhances validity by confirming consistency across multiple evidence sources [77] and aligns with recent CHI practices that integrate multimodal data streams for more rigorous interpretation [53]. During the transcript analysis, the lead author audited all codes against the transcripts and video, resolved discrepancies, and refined the final labels.

4.1.1 ***Timestamp Analysis.*** Time-stamped annotations captured the duration of completing each task and activity (getting familiar with the chatbot and different tabs, try prompt entry and run code in different wave activities), each behavior and feelings such as their moments of hesitation (silence/hovering), prompt responses and revisions, and whether participants sought help from the facilitator or referred to the written instructions. Timestamps were treated as bounded intervals (e.g., 24:00–27:00 for question-asking), allowing traceability between behavioral observations and original recordings.

4.1.2 ***Sentiment Categories.*** From the combined evidence, we identified three sentiment classes expressed during tasks: Positive, Neutral, and Negative. These classes were further divided into fine-grained emotion tags, such as "unsatisfied" (negative) for emotions like annoyed, sad, or upset, and "content" (neutral) for emotions like content, calm, or nonchalant.

- Positive: Excited, Confident, Curious, Exploring
- Neutral: Satisfied, Content, Bored

- Negative: Unsatisfied, Confused, Frustrated

We identified key moments that influenced participants' emotions and categorized the issues into three analytical lenses: (L1) UI/Interaction (system-level interaction issues), (L2) CT/Science (cognitive CT/science challenges), and (L3) (instruction-following challenges). These lenses helped distinguish the different sources of teachers' emotions: some arose from breakdowns in system-level interaction (L1), whereas others stemmed from teachers' cognitive engagement with CT and scientific reasoning (L2) or from challenges in following stepwise instructional guidance (L3). We elaborate on these distinctions in the Discussion. Examples summarized in the Table 2 for positive emotions, Table 3 for Neutral emotions and Table 4 for Negative emotions.

(1) Positive:
   - **Excited (L1, L2):** Emerged when they expressed enthusiasm for chatbot functionality and responsiveness and participants could quickly try multiple variants and see immediate, meaningful outcomes (comparing "multiple waves' visual behavior", "different prompt tabs").
   - **Confident (L3):** Appeared when task framing was clear and participants anticipated the correct output and navigated options without hesitation.
   - **Curious / Exploring (L1, L2, L3):** Marked by prompt tinkering, tab switching to inspect features, and trial-and-error debugging without facilitator help.

(2) Neutral:
   - **Satisfied (L3):** Reported when steps were clear and tasks executed smoothly. Participants progressed but did not express strong feelings and affect.
   - **Content (L3):** Participants understood the interface and read/followed instructions together, working steadily without notable difficulty.
   - **Bored (L1, L2):** Observed during over-scaffolded or too-simple tasks, long pauses/low interaction, or when UX adjustments (e.g., resizing) felt irrelevant to their goals.

(3) Negative:
   - **Unsatisfied (L1):** Unsatisfiability of the interface prompts disappearing on tab switch, the chat box occupying screen real estate, or difficulty resizing the window.
   - **Confused (L1, L2):** Getting confused after several attempts, surfaced with ambiguous onboarding and task-framing and unclear system expectations. (L1: input mechanics — e.g., pressing Enter did not send).
   - **Frustrated (L1, L2, L3):** Characterized by repeated retries without progress, incomplete activities, reliance on researchers, or the sense that "the chat takes over the screen," undermining trust in bot-generated code.

4.1.3 ***Affective Responses Across Activities.*** Figure 7 visualizes per-activity emotions as bubbles (bubble size : minutes; color : valence from negative to positive). The chart shows excitement as the most prominent positive emotion, concentrated in Coach-tab, Coding (curtain), with a smaller peak at Plastic-bag activity, while curiosity is most visible in the initial Getting familiar with the chatbot step. Within the neutral emotion, content is most apparent during Coding (mirror). On the negative side, confusion dominates, appearing in Coach-tab, Coding (curtain), Coding (mirror), and the Sound-wave activity.

Time allocation follows a similar pattern: the largest bubbles occur which indicate the most time spent occurred at Coach-tab (for both positive and negative affect) and Coding (curtain), whereas Starter code and several Sound-wave and Coding (mirror) instances are marked by small bubbles, indicating brief engagement. Moreover, six activities show mostly small bubbles paired with neutral and negative sentiment, suggesting short time involvement of the participants. Several activities show mixed color within the same column, suggesting within-step fluctuation between positive and negative states. Neutral emotion appears where progress is steady but effect is weak or muted. The average time spent on each activity was derived by calculating the mean duration across all groups, reflecting the emotional engagement during each task. Figure 8 shows, for each activity, the counts of affect-coded utterances—negative plotted to the left, neutral located at the center, and positive to the right. Segment length indicates frequency (how often the affect occurred), not intensity. Coding (curtain) has the largest negative segment alongside substantial positive, marking it as the most challenging yet engaging stage; this aligns with Exploring groups, who probed features (positive) while encountering friction (negative). Coding (mirror) and Sound Wave skew positive, indicating smoother progress once participants were oriented. Coding (glass) shows low totals across all valences, suggesting limited verbalization/engagement. Early navigation stages (Code-, Coach-, Q&A-tabs) are net positive with modest negativity, consistent with low-stakes exploration. Overall, the pattern matches our persona analysis: Task-Focused groups advanced efficiently with mostly positive affect, while Exploring groups produced more utterances.

4.2 Learner Persona

Personas are widely used in HCI to identify meaningful user groups by capturing differences in goals, behaviors, and motivations [15]. Recent AI research emphasizes that personas are especially important in intelligent-system contexts because users' attitudes toward AI, such as confidence, trust, and willingness to engage directly shape their interaction behavior [34]. Following this perspective, our personas represent distinct behavioral and emotional orientations toward the AI-enabled chatbot, making them appropriate for interpreting teacher–AI interaction patterns.

Persona = (Time-on-task, Emotion trajectory, Strategy pattern, Help Seeking)

Based on this definition, we identified three personas:

(1) Explorer/Task-Focused
(2) Mixed
(3) Frustrated

Table 5 summarizes persona-level differences in affect trajectories, strategies, and help seeking.

To construct these personas, we aggregated the coded indicators from our multi-modal dataset help-seeking behavior, task strategies, time on task, and emotional valence into an integrated behavioral–affective profile for each of the 11 groups. This profile summarized both how each group worked and how they felt during the activity.

Table 2: Positive Emotions; Strategies/Behavior and related Quotes. Legend: ♦ L1 = User Interface & system-level interaction; • L2 = Computational Thinking & Scientific reasoning; → L3 = Instructional guidance.

Positive Emotion	Strategies/Behavior The pair of teachers . . .	Examples of Quotes
Excited	♦ Express enthusiasm about the chatbot's functionality ♦ Exhibit excitement by engaging in multiple trials very quickly • Discuss how the chatbot would be useful to them or students • Show marked interest in using the chatbot to promote student learning • Eagerness in seeing the running code • Show excitement in testing various features • Show interest by comparing multiple waves' visual behavior	♦ "The chatbot functions look very nice" • "I'm really excited that it worked!" • "If we were doing this with kids or teachers, we could use the Coach tab to walk them through the steps" • "I like that it helps focus the learning." • "We could add a third sprite so that when you click it, it gets absorbed by the third material."
Confident	→ Demonstrate a strong understanding of the task • See the correct and expected output with confidence ♦ Quickly navigate through options; follow prompts without hesitation	→ "What needs to be done here is. . . " → "I've got this.." • "Exactly what I was expecting.." • "Clearly light passes through the glass more easily" ♦ "This should be prompted with the coach tab. . . "
Curious	♦ Interacts and prompts the chatbot → Starts project smoothly; showed curiosity in using instructions and other times trying independently • Acts intuitively while exploring the chatbot's features • Shows enthusiasm and curiosity in materials • Shows curiosity by switching between the tabs • Indicates desire to understand the functionality	♦ "Loaded and ready, let's dive in" ♦ "Is it possible to tweak the access?" → "Like to know how my students use the bot with instructions" • "What's the difference between Coach and Q&A here?" • "Let's see how this next tab works?"
Exploring	♦ Delve into the different functions and buttons ♦ Debugs using trial-and-error • Engages with the outcome of replacing different materials • Finds driver-navigator coding interesting • Tests when conditionals are required • Struggles with typing out the message in the prompt box and try to solve the challenges • Compare the effectiveness of different materials • Trying different examples in prompting and observe the result • Show more goal oriented behavior • Debugs by the most experienced person, no help of chatbot (debugging by testing and adjusting the code structure until it worked)	♦ "I'll change the prompt and see how response will change?" ♦ "Which tabs have the best response?" • "I swap mirror for glass, does the wave still reflect?" • "If I flip the direction and it reverses, we've found the issue" • "Is 180 degrees the reverse direction in bot?"

Table 3: Neutral Emotions; Strategies/Behavior and related Quotes. Legend: ♦ L1 = User Interface & system-level interaction; • L2 = Computational Thinking & Scientific reasoning; → L3 = Instructional guidance.

Neutral Emotion	Strategies/Behavior The pair of teachers ...	Examples of Quotes
Satisfied	→ Find their steps in Coding are true ♦ Run the code smoothly ♦ Code reuse via duplication instead of re-prompting chatbot • Enjoyed watching the outcome of the chatbot	→ "This is cool. Look at me go!" → "We knew what we were doing, and the chatbot made it even more fun." → "I wanted to figure it out on my own—and I did.
Content	• Understands the interface → Struggles with the codes without any challenge → Reads/ follows the instructions together	• "No issues here, it's working as expected." → "Okay, that clicked right away." → "You scroll, I'll call out the next instruction."
Bored	♦ low engagement with chatbot ♦ Not struggling ♦ Long pauses / low interaction ♦ Low relevance in UX appearance resizing • Dragging of blocks with no new idea or logic • → Find following the instructions tedious	♦ "No need to ask the bot; I'll just copy." ♦ "I can do it myself." → "Instructions feels like a chore"

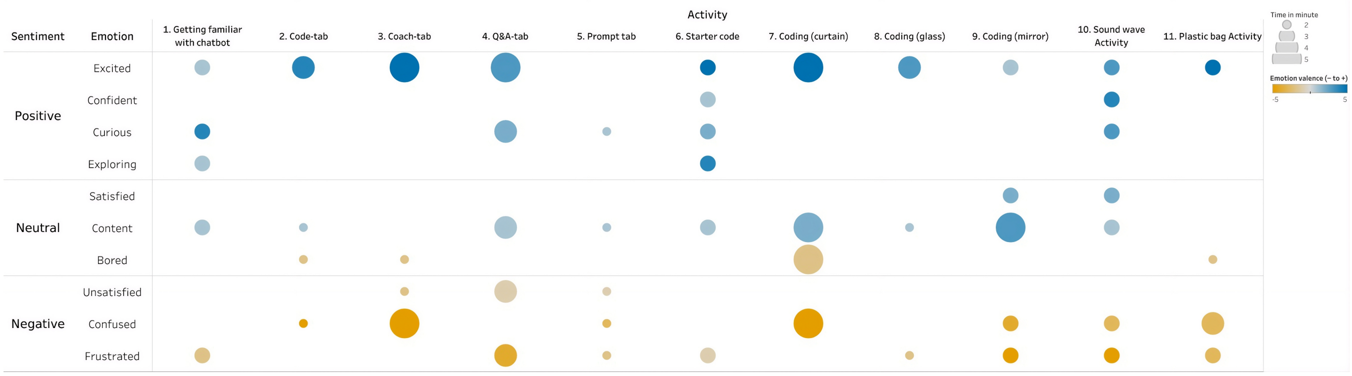


Figure 7: Bubbles encode the number of coded utterances (size); color shows valence (positive/negative/neutral). Large negative bubbles suggest intervention points.

Figure 9, visualizes time-aligned interaction flows for the three personas (Exploratory/Task-focused, Mixed, Frustrating). Each diagram is built from our time-stamped screen recordings and think-aloud: nodes show blocks of activities, Steps 1–5 (onboarding/ getting familiar with chatbot tabs) and Steps 6–11 (wave activities such as sound and light), plus decision points (e.g., Success?), help-seeking (Ask Facilitator), and outcomes (Save & Reflect). Edges encode the temporal path to the activities edge; edge style encodes valence (dashed = positive, solid = negative, neutral segments are explicitly labeled as content). Each edge is annotated with the emotion at that step (e.g., Curious, Confident, Confused, Frustrated). Loops indicate retries/backtracking (e.g., revise prompt & code / reread instructions), and Start/End are shown explicitly.

We visualized the aggregated coded features using a radar chart (Figure 10), plotting the frequency of key behaviors alongside the magnitude of time spent. The chart compares the four personas across seven dimensions: positive, neutral, and negative emotions, time spent, help-seeking, instruction-following queries, and refinement behaviors (e.g., prompt editing, tab switching, trial-and-error).

This visualization revealed clear clustering across groups and served as the analytic basis for our personas, illustrating how each group's emotional and behavioral patterns diverge in meaningful ways.

Table 4: Negative Emotions; Strategies/Behavior and related Quotes. Legend: ♦ L1 = User Interface & system-level interaction; • L2 = Computational Thinking & Scientific reasoning; → L3 = Instructional guidance

Negative Emotion	Strategies/Behavior The pair of teachers . . .	Examples of Quotes
Unsatisfied	→ Trouble finding code blocks → Need a minimize button for chatbot → Chatbot takes too much space; need to hide it → Interface problem: Struggling to resize chat window	→ “Where did the prompt go?, oh it was in other tab” → "It was not a good experience I won’t use it" → “I can’t resize the chatbox height"
Confused	• Not understanding unexpected results • Unclarity in the expected chatbot response ♦ Prompt disappears when switching tabs ♦ Pressed Enter; message didn’t send (must click Send) ♦ Confusion on onboarding and task-framing (Didn’t know the goal (“get to know the ♦ chatbot” vs wave activity) ♦ Unclarity in finding what the issue is after several attempts Unsure whether costume saved automatically	• “I bet the wave is supposed to stop at the mirror… is that what’s happening?” • “What happens if I pull . . . does it still work?” ♦ “Where did the prompt go?, oh it was in other tab” ♦ “Is there more text below? How do I scroll the Coach tab?” ♦ “Can I ask the chatbot where to find the block and still figure it out myself?” ♦ “Is this pseudocode for non-blocks too, or what am I supposed to follow?”
Frustrated	• Repeated retries without progress multiple time • Left coding activity incomplete? ♦ Enter didn’t send the prompt ♦ Reliance on researchers, undermining chatbot trust ♦ Couldn’t find needed blocks ♦ Teachers disliked chatbot code ♦ Lowered perceived usefulness of the chatbot/code for classrooms ♦ Chatbot UI got in the way, or generated code felt unusable → Following instructions and getting back was tedious	♦ "keep dragging this chat box out of the way—why won’t it stay put? I can’t even find the new code” ♦ “This is frustrating: the chat takes over the screen, the scripts are all over the place, and I’m hunting for code that should be right there. → “If a sixth grader saw this, they wouldn’t understand it, and honestly, with the chat in .way and no blocks showing, neither do I.”

Table 5: Observable Personas, Time Spent, Affect Trajectories, Strategies, and Help Seeking

Persona	# of Groups	Time Spent	Emotion Trajectory	Strategy/Behavior	Help Seeking
Exploring	6	Long	**Mostly Positive/Neutral** (Engaged, Curious, Satisfied)	• Instruction-led execution • Tab-switch exploration • Prompt refinement & output comparison • Few stalls	Low, after several self-retries
Task-Focused		Short			
Mixed	3	Average Different Time Across stages	**Variable** (shifts between positive and negative across stages)	• Strategy shifts within session • Following instructions and exploration	Occasional, to recover from stalls
Frustrated	2	Long dwell around failures	**Mostly Negative** (Confused, Unsatisfied, Frustrated)	• Repeated retries without progress, backtracking	Early/frequent; Signs of Fatigue or Abandonment

Interpretation of the Radar charts based on each persona:

Exploratory. Exploratory groups (1, 3, 7, 11) spent the longest time on task, showed the highest positive emotion, and frequently

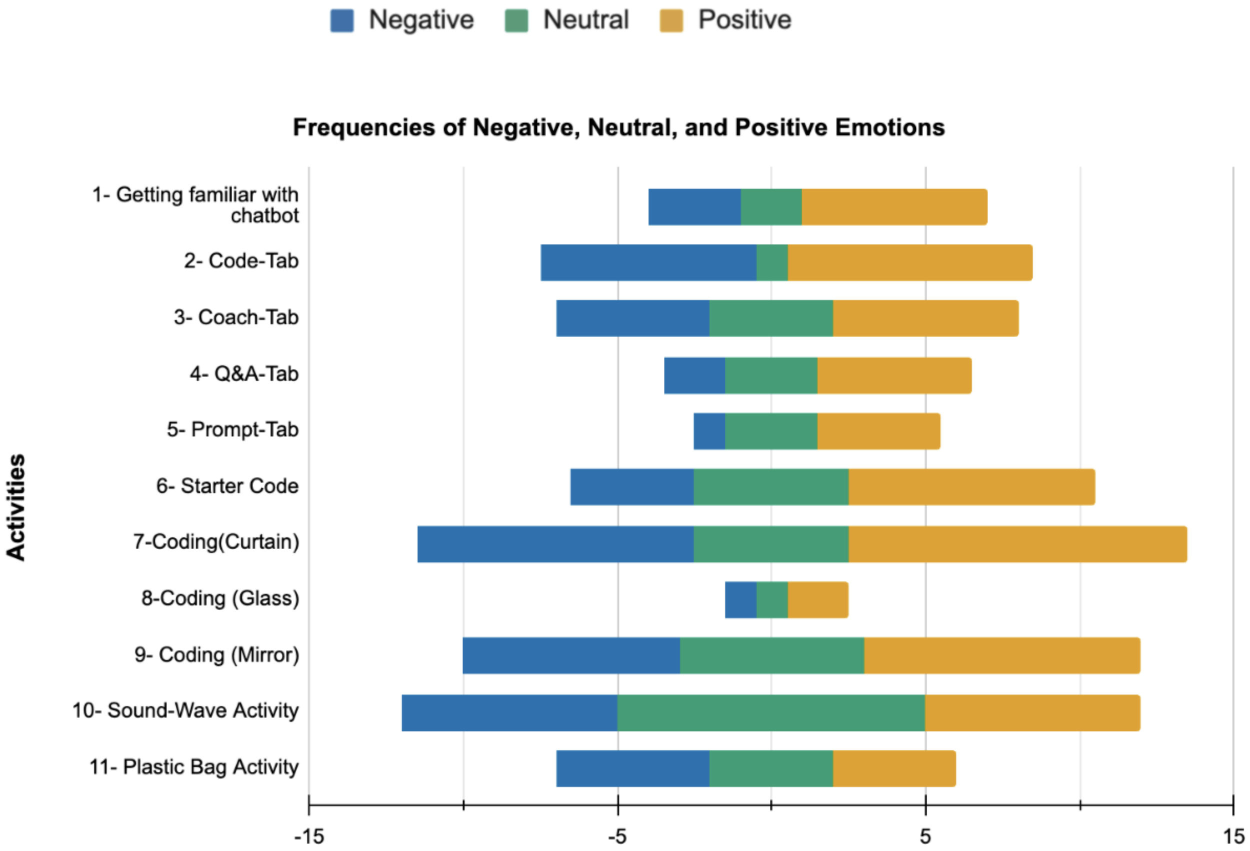


Figure 8: Counts of coded utterances per stage; negative (left), neutral (centered), positive (right), enabling comparison of affect balance across activities.

refined prompts, switched tabs, and experimented with bot features. They needed little help and demonstrated curiosity-driven experimentation.

Task-Focused. Task-Focused groups (2,8) showed high positive and neutral emotion with minimal negative affect. They followed instructions efficiently, interacted with the chatbot when needed, and completed tasks quickly with little help-seeking. Exploratory and Task-focused personas differed primarily in efficiency versus depth of exploration; we therefore report them together.

Mixed. Mixed groups (4, 5, 6) displayed moderate levels of positive, neutral, and negative emotion. Their engagement fluctuated across stages: they alternated between following instructions, brief exploration, and occasional frustration. They sought help periodically when stuck but generally continued independently.

Frustrated. Frustrated groups (9, 10) showed the highest negative emotion and the lowest positive emotion. Although they spent substantial time on the activity, much of it was directed toward repeated retries and heavy help-seeking. Their interactions reflected confusion, stalled progress, and difficulty navigating both the bot and the task.

4.3 Thematic analysis of Open-Discussion Data: Evaluating Perceptions and Planned Pedagogical Practices

In the second phase of the analysis, we conducted a thematic analysis of the post-question and discussion responses using a combined deductive and inductive approach [25]. We created deductive parent themes from the discussion questions and answers (confidence, teacher preparation, student use, required changes, scaffold changes) and inductive sub-themes emerged within those domains. Four researchers co-developed the themes and the codebook: Two trained researchers independently coded the open-discussion questions. Two additional researchers re-reviewed and refined the codes, grouping them into sub-themes and higher-order themes. They resolved the disagreements through discussion to agreement, and finalized the thematic structure and used them in the results. Our analysis resulted in 144 codes which two researchers sorted into 20 categories which can be found in Table 6. Those categories were

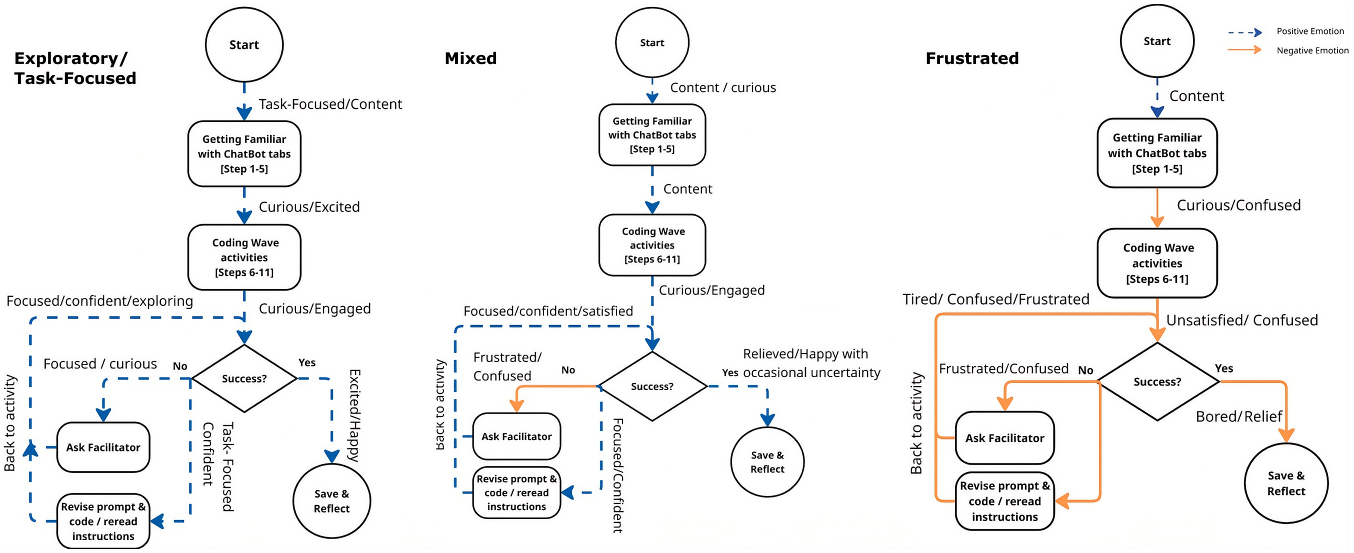

Exploratory/ Task-Focused
Start
Task-Focused/Content
Getting Familiar with ChatBot tabs [Step 1-5]
Curious/Excited
Coding Wave activities [Steps 6-11]
Focused/confident/exploring
Curious/Engaged
Focused / curious
No
Success?
Yes
Excited/Happy
Back to activity
Ask Facilitator
Task- Focused Confident
Revise prompt & code / reread instructions
Save & Reflect
Mixed
Start
Content / curious
Getting Familiar with ChatBot tabs [Step 1-5]
Content
Coding Wave activities [Steps 6-11]
Curious/Engaged
Focused/confident/satisfied
Frustrated/ Confused
No
Success?
Yes
Relieved/Happy with occasional uncertainty
Back to activity
Ask Facilitator
Revise prompt & code / reread instructions
Focused/Confident
Save & Reflect
Frustrated
Start
Positive Emotion
Negative Emotion
Content
Getting Familiar with ChatBot tabs [Step 1-5]
Curious/Confused
Coding Wave activities [Steps 6-11]
Tired/ Confused/Frustrated
Unsatisfied/ Confused
Frustrated/Confused
No
Success?
Yes
Bored/Relief
Back to activity
Ask Facilitator
Revise prompt & code / reread instructions
Save & Reflect

Figure 9: Different Personas' Reactions and Emotion in Each Stage of Activities

then sorted into 6 themes: 1) Benefits & risks to students, 2) pedagogical practices around chatbot, 3) assisting all learners, 4) chatbot allowance in classrooms, 5) benefits for teachers, and 6) usability.

4.3.1 ***Benefits to students.*** Teachers recognized that the chatbot could benefit students in a number of ways. First the chatbot could increase student self-efficacy. A teacher in Group 10 mentioned that some students lack self belief around programming and with the chatbot students may realize "hey, I can go to town on stuff like this... They may just really feel much more confident with the chat bot", meaning that students might be able to work faster and then feel better about their abilities. Additionally, they identified that the chatbot could help students build their computational thinking (CT) skills. For example, the chatbot can build students' CT vocabulary; Group 2 said that the chatbot could "reinforce that vocabulary, and then… you know, and after they see it and hear it, they start. They start speaking it."

Six of the groups mentioned that the chatbot could help grow their students' prompting skills. Group 1 mentioned that "I feel like I would use it because I would want my students to like, learn what prompts do you need to use and like get familiar with [prompting]". Groups 1 & 10 mentioned the balance between teaching programming basics and practical skills.

4.3.2 ***Risks to students.*** Group 10 mentioned that the chatbot changed the learning focus from "How do I use block coding" to "learning about, how do I write this prompt to get me to what I want?". Group 10 said that they do not mind the shift towards prompting, as some skills fade in importance, "coding is like becoming cursive, well now we type". Some of the groups were concerned about this shift. Group 6 reflected on the loss of productive struggle that students might lose out on, "I didn't feel like I was making headway in the same way that I do when I'm just messing around with blocks on my own, you know. Like, I felt like it was frustration that wasn't productive, because I couldn't figure out what it wanted me to do". Teachers from Group 4 expressed concern that students might not truly understand the code they generated. They worried that because the chatbot "gives them exactly what to do," students would just copy the code without understanding, questioning, "did they actually understand what was happening? Or did they just copy that?"

4.3.3 ***Pedagogical Practices Around Chatbot.*** All groups mentioned that they would scaffold student introduction to the chatbot through guided examples, a prompting worksheet, a slidedeck, a video tutorial, and explaining the tabs. A teacher from Group 6 highlighted the importance of establishing the chatbot's fundamental purpose, especially for students who have little experience with such tools. The teacher would frame the chatbot as "basically like a tutorial or a help," because many students "don't know that you use chatbots for all different things."

To ensure students still learned foundational skills and did not become overly reliant on the chatbot, teachers discussed several pedagogical safeguards they would implement. Several teachers expressed a desire to build students' foundational programming understanding before introducing them to the chatbot. A teacher from Group 8 felt that giving students the chatbot immediately would be "handicapping" them, likening it to "giving them a textbook that has all the answers." Instead, this teacher would prefer students to "just play" with programming first and then perhaps "have them use the chat on a bigger project." Similarly, a teacher from Group 2 felt that "trial and error is the best way versus giving [the chatbot] to them first." Teachers were concerned that students might use the chatbot without understanding the code it generated. To mitigate this, a teacher from Group 7 would ask students questions to evaluate their comprehension of the code, such as, "How did you do that?" or "So what did you do when the bot made the code for you? What did you do to make sure you understood what the code is?" To prevent students from becoming distracted or cognitively overloaded, teachers mentioned the need for clear structure in assignments.

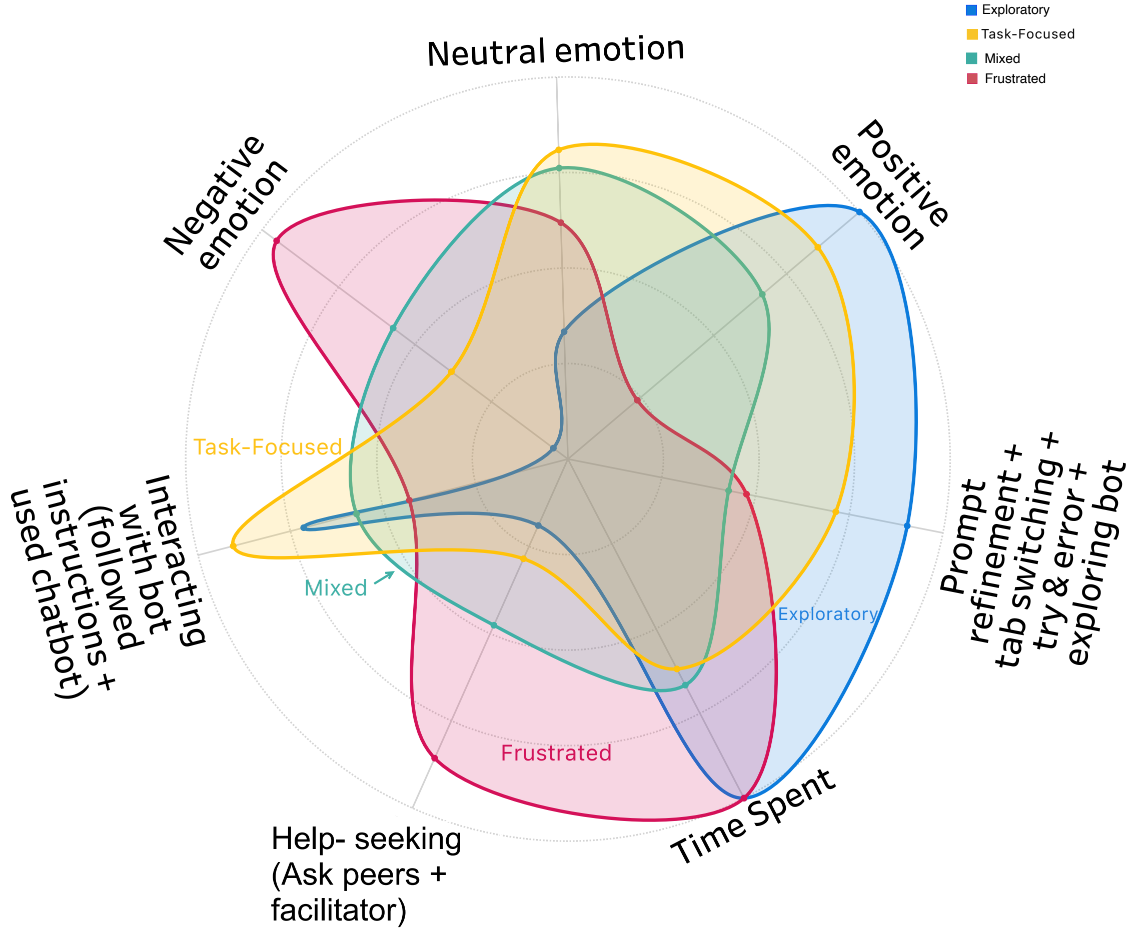


Figure 10: Radar chart comparing four personas—Exploratory, Frustrated, Mixed, and Task-Focused—across seven dimensions: three emotion states (negative, neutral, positive), time spent, help-seeking, instruction-following bot queries, and refinement-oriented interactions (prompt editing, tab switching, and trial-and-error exploration). Higher values indicate more frequent or intense presence of that dimension for a given persona.

A teacher from Group 3 noted that "kids like structure" and that allowing them to "tinker" without clear guidance might cause them to "shy away from... the task at hand" and said they would design activities so they focus on only one learning objective at a time.

Teachers expressed a desire for control over various chatbot features to better manage student use and learning. A primary request from seven of the groups was the ability to disable specific AI features for students. This control would allow teachers to tailor the tool to a particular lesson's goals. For example, a teacher from Group 5 stated it would be helpful if "the code button is off limits" while still allowing students to use the "QA" (questions and answers) function. Teachers also wanted control over the level of detail the chatbot provided in its responses. A teacher from Group 9 envisioned a "tiered approach" where the teacher could set the level of response. A Level 1 response, for instance, might "give me the code," whereas a Level 2 response would "prompt me a little bit, so that I understand what it is that I'm trying to do." Some teachers wanted the ability to limit the number of prompts a student could submit. A teacher from Group 1 suggested this as a way to create a challenge, stating that for students who are already comfortable, the teacher might "limit you to… two questions, you get to ask the chat."

Groups 10 & 3 mentioned that they could use the chatbot to support students in projects in which there is a high amount of student choice. Groups 10 & 11 also wanted the chatbot to have gamified features and awards to keep students motivated and engaged.

Table 6: Themes and categories produced from thematic analysis of teacher discussion groups.

Themes	Categories
Benefits & Risks to students	Chatbot may hurt learning to program Building student prompting skills Chatbot can increase student self-efficacy Chatbot can build student knowledge
Pedagogical Practices Around Chatbot	Build foundational learning first Scaffold intro to chatbot Control of chatbot features Chatbot and pair programming Feature requests for engagement
Assisting All Learners	Chatbot can support novice students Chatbot can provide challenge to advanced learners Chatbot not appropriate for advanced learners Chatbot can help with differentiation Feature requests for accessibility
Allow chatbot in classroom?	Would allow students to use the chatbot Would not allow students to use the chatbot
Benefits for teachers	Chatbot builds teacher confidence Chatbot can support teacher instruction Chatbot can support student independence
Usability	Usability problems

4.3.4 ***Assisting all learners.*** Teachers recognized the chatbot's potential to help a diverse range of students, from novices to advanced learners, and those with specific accessibility needs. Eight of the groups mentioned that the chatbot could be a powerful tool for differentiation. For example, a teacher from Group 8 felt that the chatbot would allow accelerated students to "move forward and then give them like more challenges" and "letting them explore a little further." However, a teacher from Group 5 also noted that some "of the higher students wouldn't want [the chatbot]," as they "like to figure it out on their own."

Teachers also saw the potential for the chatbot to assist students with specific accessibility needs. Teachers requested text-to-speech options to support different learning styles. A teacher from Group 10 asked if the chatbot could have "read aloud options... Is there a way that you can hear it?" to accommodate students who struggle with reading. Similarly, a teacher from Group 6 highlighted the need for language and reading support, explaining that some students are "Spanish speaking" and need to "hear it in Spanish, or read it in Spanish." This teacher also noted that some students "read at a second grade level" and would find the large amount of text "really overwhelming."

4.3.5 ***Benefits to teachers.*** Teachers in Groups 1, 2, and 9 felt that the chatbot could help them with leading activities in their classrooms. A teacher in Group 1 said that it would make them feel "more comfortable" working with students because every student's code is different, and having a chatbot that can interpret student code is like "having an answer key going into an activity which is nice." Another teacher from Group 9 commented that the chatbot "would make [them] feel really confident about going into a teaching situation, because some kids [ask] like, 'How do I do this?' I'm like 'I have no idea'". Another teacher in Group 9 felt like the chatbot had limited ability to support them in leading activities because, "It showed what kind of code I needed to use… But, It didn't explain why".

Five of the groups mentioned that the chatbot could support teachers with student interventions. The chatbot could take "pressure off of the teacher" (Group 7), as each student could get support without the teacher having to help everyone individually. Similarly, a teacher from Group 9 explained that "you have 28 kids, but you're only one person" and that helping each individually isn't always possible.

Teachers also saw the chatbot as a way to build student independence, which would free up their own time and energy. By providing students with a tool for debugging and troubleshooting, the chatbot could empower them to solve problems on their own. One teacher from Group 2 noted that the chatbot's questioning approach was similar to what they would use to guide students, stating, "I appreciated the questions that it asked, because those are questions that I would ask." Another teacher from Group 4 noted that the chatbot would be "handy for troubleshooting," especially when students need help with "little minor details" while a teacher is busy with other students.

4.3.6 ***Chatbot allowance in classrooms.*** All the groups reported that they would allow their students to use the chatbot, though many had specific ideas about which features to use and when. The teachers' decisions often depended on their primary pedagogical goal: whether they were focused on teaching core computational thinking (CT) skills or using the chatbot as a tool to teach other domain knowledge. Some teachers were less concerned about students learning CT concepts and more focused on using the chatbot as a medium to teach other subjects. For these teachers, using all of the chatbot's features, including the code generation function, was acceptable. A teacher from Group 1 explained that in their classroom, they wouldn't use the chatbot to necessarily teach coding, but rather to use it as a "medium to teach something about like insects." For example, they would have allowed students to use full chatbot functionality on the activity given during the co-design sessions, noting that even with the chatbot, students would still have to "think about that, because the prompt didn't work when you just said to make it."

In contrast, four of the groups (3, 5, 6, and 8) stated they would not allow students to use the code generation tab, at least in the beginning. They felt that this feature would hinder students' ability to learn foundational programming concepts. A teacher from Group 8 felt that the code generation feature "giv[es] you the answer too quickly." This teacher compared the process to "copying and pasting" and felt that it prevented students from engaging in the "cognitive ability of transferring information from one place to another."

5 Discussions

Without prompting, teachers have highlighted a suite of needs for CT-integrated instruction involving block-based programming that have been identified in prior research about novice programming and new educational tools [9]. The theme of pedagogical practices shows that teachers know both that students need to learn foundational skills before being offered shortcuts (like a code generator), but also that students must be taught how to use new tools like the chatbots, and this need is separate from learning the content being taught within the tool.

The theme of chatbot benefits and risks highlights the possibility of adaptive individual tutoring - since they learn from the chatbot embedded within the environment, students can build knowledge and self-efficacy. Furthermore, teachers identified that similar chatbots will likely be used in students' future careers, so practicing with prompting chatbots is important learning. The theme of assisting all learners addresses an important need teachers have expressed about learning to integrate new CT curricula, that part of the teacher's job is to differentiate the educational materials for students with differing prior preparation and needs. Teachers see the potential for the chatbot to reduce the need for teachers to prepare additional materials or adjustments ahead of time since the chatbot can meet some of the learner's needs for extra explanations, language translations, speech to text, or further challenges.

The theme of benefits for teachers highlights that teachers could immediately see impacts for themselves – building their own self confidence and knowledge about programming, and reducing the need for them to provide help to every student during programming, while also increasing student independence or ability to find help for themselves.

5.1 RQ1: Affect & Attitude

In this paper, we investigated how interacting with a chatbot for learning programming impacts teachers' affect and attitudes. Our analysis revealed that teachers' emotional responses and attitudes varied depending on their engagement with the chatbot [31].

Teachers showed excitement and curiosity when exploring the chatbot's features, while some felt neutral, such as contentment, when tasks were smooth to follow. However, there were also instances of boredom during repetitive or overly scaffolded tasks, which resulted in reduced engagement. Additionally, they experienced frustration and confusion when facing with interface issues, unexpected or mismatched responses. These emotional fluctuations highlight how teachers' experiences with the chatbot affect their attitudes.

Learners' emotions depend on interactions, and resolving barriers can shift feelings from negative to positive. This highlights the difference between LLM and human support, emphasizing when both are needed. Understanding this helps teachers anticipate and address students' needs in moments of confusion or frustration, whether due to lack of guidance, usability issues, or difficulty understanding. This insight empowers teachers to create a more supportive, adaptive learning environment more effectively. Our lens coding, (L1) UI/Interaction (system-level interaction issues), (L2) CT/Science (cognitive CT/science challenges), and (L3) Instruction-Following (instruction-following challenges)—clarifies which emotions were primarily driven by interface limitations versus teachers' own cognitive or instructional work. L1 emotions were tightly coupled to system-level interaction issues, such as disappearing prompts, the chat pane obscuring code, or confusion around input mechanics, all of which reliably produced confusion, unsatisfaction, or frustration even when teachers understood the underlying CT/science concepts. In contrast, L2 and L3 emotions were more often associated with cognitive and pedagogical demands: teachers' curiosity, confidence, or confusion when reasoning about wave interactions, debugging logic, or interpreting stepwise written instructions, even when the interface behaved as intended. This distinction helps separate emotions that signal breakdowns in the chatbot interface or coding environment (L1) from emotions that indicate productive or unproductive struggle with content or instructional framing (L2/L3).

5.2 RQ2: Perceptions of LLM use for classroom

Our findings highlight teachers' perspectives on both the benefits and risks of using large language models (LLMs) in block-based programming education. Teachers noted that LLMs can boost students' confidence and self-efficacy [47], helping them make faster progress, developing computational thinking skills, allowing students to build foundational programming knowledge step by step [32]. Teachers also noted that interacting with LLMs can enhance students' prompting skills, which is an essential skill for engaging with AI-powered tools [72]. Teachers raised concerns about students over-reliance on chatbot for programming, which could prevent students from understanding programming concepts to simply writing prompts, undermining the value of "productive struggle" in learning. Secondly, they cautioned that chatbots could diminish critical thinking, problem-solving, debugging, and troubleshooting skills. Chatbots also can help teachers by reducing workload, saving time and free teachers to help students with more complex concepts. However, they worried about the lack of transparency behind chatbot-generated code making it difficult to assess students' understanding [81]. Many teachers desired more control over chatbot features, such as limiting automatic code generation to guide, not replace student work. Teachers emphasized careful, scaffolded integration with clear control features to ensure chatbots enhance learning without undermining essential skills development.

One interesting finding in our results is that teachers differed on whether or not the chatbot would be useful for advanced learners, with some teachers feeling that advanced students would not need or like chatbot support, while others felt that advanced learners could use the chatbot to explore beyond a given assignment. These differing perspectives may be rooted in teacher experiences with students. For example, some experienced CT learners reject systems and supports built for novices. Other teachers may have struggled to come up with ideas for experienced CT learners to extend activities designed for novices, and perceive that the chatbot would enable independent exploration without needing teacher help on the programming language or how to code a new idea their students may have. Research has demonstrated that the high variance in prior experience and motivation for students with regard to CT and programming can be a challenge in the classroom, precisely for these reasons [49].

5.3 Mapping into Psychological-Educational Frameworks

To situate an understanding of teachers' behavioral and emotional dynamics within a broader decision-making framework, we turn to psychological frameworks such as Technology Acceptance Models (TAM) and Theory of Planned Behavior (TPB) which offer complementary lenses for understanding teachers' attitudes, perceived control, and intentions to use AI systems.

Technology Acceptance Models (TAMs), were first introduced by Davis [19, 20], to explain users' attitudes toward technology through four core constructs: perceived usefulness, perceived ease of use, attitude, and behavioral intention. In educational technology research, TAM has been extended to incorporate pedagogical and learning-related factors [44]. This framework has frequently been used to assess students' acceptance of AI and to interpret their experiences; for example, studies have shown that students report significantly positive perceptions of ease of use and perceived usefulness when receiving assistance from AI [5].

The Theory of Planned Behavior (TPB) [2, 4] provides an additional perspective for examining the factors that shape human actions within teaching and learning environments. TPB explains behavioral intention through attitudes, subjective norms, and perceived behavioral control, which reflects teachers' confidence in using AI tools in the classroom based on their perceived control over both technical and instructional challenges [3]. This makes TPB particularly useful for understanding how educators and students decide to engage with technology in educational and research contexts [39]. TPB has been widely used in studies of decision-making processes and has been applied extensively to predict K–12 teachers' intentions to adopt educational technologies [17, 70].

Scholars have also employed TPB investigate GenAI adoption in GenAI in education; for example, Ivanov et al. [39] found that perceived benefits of GenAI enhance attitudes and perceived behavioral control, which subsequently increase intention to use these tools.

Researchers such as Abdullah and Ward [1] have applied TAM-based models to e-learning and found that self-efficacy, social influence, enjoyment, anxiety, and experience significantly shape perceived usefulness and ease of use, while Kemp et al. [44] used a similar framework with university students in virtual classrooms and showed that comfort and well-being, cognitive engagement, and access and convenience strongly predict perceived usefulness and behavioural intention to use the technology. Smolinski et al. [76] demonstrates that LLMs can reliably approximate TAM measures, producing acceptance ratings that align closely with human expert coders. Similarly, TAM has been applied to LLM-driven educational chatbots, showing that students perceive these systems as useful and easy to use and report generally positive attitudes toward adopting them as learning support tools [57].

Since TPB captures behavioral and control-related factors while TAM focuses on technology-specific perceptions, TPM and TAM can not be used alone for explaining users' intention to use novel technologies such as AI systems and chatbots. Researchers must integrate the two for a more comprehensive foundation for explaining users' intentions to adopt AI-driven tools [41, 56].

In our study, we propose to integrate TAM and TPB to develop a preliminary conceptual foundation for understanding teachers' interaction, evaluation, and intention to use the chatbot (Figure 11). We conceptualized system and context factors—including the chatbot's interaction design, the structure of the coding tasks, teachers' prior experience with block-based programming and AI tools, and the broader pedagogical context, as external variables that may have shaped teachers' moment-to-moment experiences with the system. Emotions are inseparable from cognition and represent a necessary component of user-centered design [59]; emotional responses can influence engagement and behavior in interactive systems [78]. These encounters elicited affective responses such as confidence, curiosity, frustration, and confusion, alongside teachers' help-seeking, which may function as mediating processes linking system/context factors to teachers' evolving perceptions of the chatbot. These mediators may influence teachers' sense of control, how they navigate difficulties, and the degree to which they feel supported while working with the tool. Based on these mediated experiences, teachers appeared to form preliminary perception about the chatbot's perceived usefulness—particularly its benefits to students and teachers (Sections 4.3.1, 4.3.5)—and its perceived ease of use. Prior work suggests that perceived usefulness and enjoyment can contribute to more positive attitudes, which in turn may increase intention to use and continuance intention [42].

In turn, we hypothesize that these perceptions informed teachers' attitudes toward the chatbot as well as their perceived behavioral control, reflected in their confidence and self-efficacy when using and managing the tool. Together, this model explains our observed patterns in how teacher attitudes and perceived behavioral control (PBC) may have shaped teachers' intentions to adopt the AI-enabled chatbot in their instructional practice. These relationships collectively motivated our integration of TAM and TPB into a single explanatory framework for modeling teachers' behavioral intentions (Figure11).

We found that all groups expressed interest in using the chatbot, and we believe that their emotional experiences shaped how and why they intended to use it. For instance, Group 1 viewed the chatbot not for teaching coding, but rather to use it as a medium to teach science and computational thinking, whereas Groups 3, 5, 6, and 8 focused on student learning and preferred limited code-generation features.

These differences align with variation in perceived usefulness, perceived ease of use, and perceived behavioral control, which TAM and TPB identify as key factors shaping emerging intentions to use a technology, and they help explain why each group tends to cluster into distinct personas, as visualized in the radar chart. To make this relationship more concrete, we examine how the radar-chart patterns that define and articulate the personas map onto these theoretical constructs. For example, the Exploratory and Task-Focused profiles—marked by higher positive emotion, active engagement, and lower help-seeking—align with higher perceived usefulness, perceived ease of use, and perceived behavioral control. Their enthusiasm and willingness to try multiple prompts suggest that they see the chatbot as beneficial for achieving their instructional goals (high perceived usefulness); their smooth navigation of features and limited reliance on facilitator support indicate that they experience the system as straightforward to operate (high perceived ease of use); and their confidence in deciding when and how to use the chatbot, as well as their persistence in troubleshooting, reflects a strong sense that they can successfully control and manage the tool in their practice (high perceived behavioral control).

By contrast, the Frustrated profiles—characterized by high negative emotion, frequent requests for help, and repeated breakdowns in task progress—correspond to lower perceived ease of use and weaker behavioral control. Their confusion and dependence on external support signal that interacting with the chatbot feels effortful and unpredictable (low perceived ease of use), and their tendency to abandon or narrow their use of the tool suggests a diminished sense that they can effectively operate it in their own classrooms (low perceived behavioral control), which in turn undermines perceived usefulness. Mixed profiles fall between these extremes: their oscillation between curiosity and frustration, intermittent engagement, and selective help-seeking translate into ambivalent perceptions of usefulness, ease of use, and control. Collectively, the radar chart visualizes how these emotional trajectories and interaction styles may mediate the relationship between system/context factors and teachers' evolving attitudes and behavioral intentions toward adopting the chatbot.

Our study did not assess actual use of chatbot in classrooms, the findings are consistent with belated frameworks, indicating that emotional states may be playing a central role in technology acceptance, linking engagement with behavioral intention. We posit that these emotions dynamically shaped teachers' agency, confidence, and decision-making pathways regarding AI integration in educational settings.

In addition to acceptance-based models like TAM, teacher- knowledge frameworks such as Technological Pedagogical Content Knowledge (TPACK) provide a complementary lens for understanding how educators integrate technology into instruction [55]. Research shows that TPACK-informed professional development (PD) helps teachers develop effective pedagogical strategies for integrating GenAI alongside technological skills [75]. Their TPACK-aligned PD led to substantial gains in teachers' perceived ease of use, perceived usefulness, attitudes toward AI tools, intention to use them, actual usage, and self-efficacy—demonstrating the effectiveness of structured, TPACK-guided PD in supporting educators' adoption of emerging technologies.

Our findings can also be interpreted through the lens of the TPACK framework. Teachers' interactions with the chatbot and block-based activities unfolded on top of their existing technological, pedagogical, and content knowledge, meaning that our system and context factors (e.g., chatbot design, task structure, PD scaffolds) were experienced through their evolving TPACK. The affective reactions we observed—such as confidence, curiosity, frustration, confusion, and reliance on facilitator support—highlight moments where teachers were able to mobilize their technological–pedagogical knowledge or, conversely, where gaps in that knowledge became visible. These experiences shaped teachers' perceived usefulness and perceived ease of use of the chatbot, as well as their attitudes and perceived behavioral control, aligning with prior TPACK-informed PD work that reports gains in ease of use, usefulness, self-efficacy, and intention to use AI tools.

5.4 Design Implications

Many negative affective states emerged not because the chatbot produced incorrect content, but because teachers lacked Situation Awareness (SA) of how the system operated. While basic perception (SA Level 1) was sometimes hindered by interface issues like disappearing prompts or hidden code, a deeper breakdown occurred at 'Level 2 SA' (Comprehension). Even when teachers could see the output, they often lacked the transparency to understand its rationale or constraints [40].

To reduce this confusion, systems should surface clearer explanations of generated outputs. For example, automatically commenting generated code, highlighting specific changes (diffs), and directing users to where code appears can help teachers maintain situational awareness by bridging the gap between raw automation and human understanding. These transparency-oriented design choices address a major gap in teachers' mental models: knowing what the system did and why it did it (SA Level 2: Comprehension). Establishing this understanding allows teachers to accurately predict how the system will behave under different constraints (SA Level 3: Projection), a necessary step before they can confidently integrate the tool into their teaching practice [71].

When teachers experienced frustration, it often stemmed from stagnation—moments where they tried several prompts but made no observable progress. These episodes were not limited to interaction and system breakdowns (Analytical Lense L1) ; they also occurred during (Analytical Lense L2) CT/Science (cognitive CT/science challenges) and (Analytical Lense L3) (instruction-following challenges),

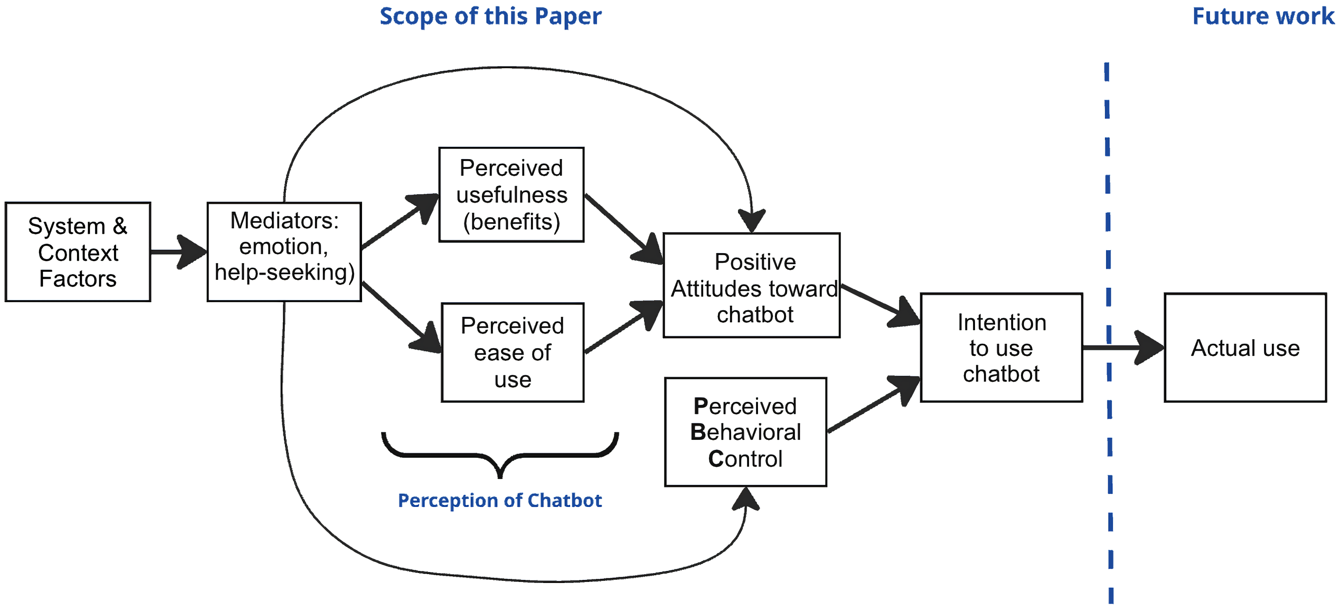


Figure 11: Integrated TAM (Davis, 1989) and TPB (Ajzen, 1991) model used to interpret teachers' intention to adopt the AI chatbot.

when teachers struggled with debugging logic, interpreting scientific explanations, or adapting stepwise guidance. In such situations, simply echoing empathic phrases is not enough to support teachers. Cuadra et al. (2024) showed that conversational agents can express emotional reactions, but struggle with deeper interpretation and follow-up, suggesting that their empathy can feel hollow or performative [16]. For teachers, therefore, effective support must go beyond emotional acknowledgment to provide instrumental scaffolding: the system should clarify goals, ask what the user is trying to achieve, or offer to restate the prompt. Moving beyond performative empathy requires equipping the system with actionable conversational repair strategies—such as offering explicit options or clarifying intent—rather than simply asking users to try again [10]. By coupling empathy with these goal-directed repairs, the system shifts from a passive observer to an active partner. This approach not only recognizes the user's emotional state but helps them resolve the underlying breakdown, thereby restoring their agency and competence in the face of automation [40].

5.5 Design Recommendations

Recommendation 1: Chatbot supporting differentiation in classrooms. Past research has shown that while teachers want to help every student in their classroom one-on-one, they do not have the time to do so, which results in them building pathways for students to independently problem solve [51]. The teachers in our study understood the chatbot to be something that could support differentiation in the classroom while also increasing student independence. To balance student agency and support based on prior skills, we recommend a feature that adapts to students' skill levels, adjustable by teachers or students [38]. The chatbot could offer more guided prompts initially, decreasing as students improve to encourage autonomy. Additionally, integrating an embedded pre-assessment or sample activity to calibrate prior knowledge and adjust support levels based on performance would be beneficial [11, 13].

Recommendation 2: pedagogical integration of chatbots, allowing teachers to enable or disable the code generation function [6]. This feature would enable teachers to toggle chatbot functionality based on classroom or individualized student needs, such as limiting it to answering conceptual questions or providing code suggestions instead of full code, these controlling features could ensure that students develop problem-solving skills rather than overreliance on chatbots. The teachers in our study feared that code generation would inhibit student learning - especially for students who have not yet mastered basic coding structures.

Recommendation 3: Adapt the chatbot to fit the needs of diverse learners. Our teachers expressed concern over the text-heavy output - especially for students who have lower reading levels or speak English as a second language. Modifying the chatbot to produce visual or auditory output may be another way to support these populations.

Additional recommendations are to allow users to modify the layout of their environment similar to professional IDEs, as our teachers had strong opinions about the layout which impacted their general attitudes about using the chatbot.

6 Conclusion

In this study, we explored how teachers perceive and react to using a Large Language Model (LLM) chatbot in the classroom, with a focus on their emotions and behaviors during interactions. Our key findings highlight the diverse teacher profiles and varying scaffolding needs, illustrating how teachers' attitudes towards the chatbot can shift based on their experiences. We identified both the benefits and risks of using the chatbot for teaching and learning. Our study demonstrates that chatbots can provide significant benefits to both teachers and learners. They can accelerate learning and troubleshooting, saving valuable time, and can boost self-confidence, particularly for novice learners. However, different users with varying levels of programming skills benefit from chatbots in different ways. Our findings contribute to the ongoing discussion about the use of LLMs in education [45, 64], highlighting the importance of fostering critical thinking, ensuring effective interactions, and providing valuable learning experiences [63]. We also identified key opportunities for future research, particularly in exploring how AI-driven tools can be designed to prevent over-reliance, avoid replacing teachers' roles, and mitigate the risk of diminishing critical thinking skills. Additionally, we emphasize the need for design recommendations that maximize the benefits of AI tools while ensuring they meet the diverse needs of users.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Award No. 2405854 and 2405855. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor.

References

[1] Fazil Abdullah and Rupert Ward. 2016. Developing a General Extended Technology Acceptance Model for E-Learning (GETAMEL) by analysing commonly used external factors. *Computers in human behavior* 56 (2016), 238–256.

[2] Icek Ajzen. 1985. From intentions to actions: A theory of planned behavior. In *Action control: From cognition to behavior*. Springer, 11–39.

[3] Icek Ajzen. 1991. The theory of planned behavior. *Organizational behavior and human decision processes* 50, 2 (1991), 179–211.

[4] Icek Ajzen. 2020. The theory of planned behavior: Frequently asked questions. *Human behavior and emerging technologies* 2, 4 (2020), 314–324.

[5] Mohammed AM Algerafi, Yueliang Zhou, Hind Alfadda, and Tommy Tanu Wijaya. 2023. Understanding the factors influencing higher education students' intention to adopt artificial intelligence-based robots. *Ieee Access* 11 (2023), 99752–99764.

[6] Manal A Almuhanna. 2024. Teachers' perspectives of integrating AI-powered technologies in K-12 education for creating customized learning materials and resources. *Education and Information Technologies* 30 (2024), 1–29.

[7] Renate Andersen, Anders I Mørch, and Kristina Torine Litherland. 2022. Collaborative learning with block-based programming: investigating human-centered artificial intelligence in education. *Behaviour & Information Technology* 41, 9 (2022), 1830–1847.

[8] M. Beege, C. Hug, and J. Nerb. 2024. AI in STEM education: The relationship between teacher perceptions and ChatGPT use. *Computers in Human Behavior Reports* 16 (2024), 100494. doi:10.1016/j.chbr.2024.100494

[9] Nigel Bosch and Sidney D'Mello. 2017. The affective experience of novice computer programmers. *International journal of artificial intelligence in education* 27, 1 (2017), 181–206.

[10] Anouck Braggaar, Jasmin Verhagen, Gabriëlla Martijn, and Christine Liebrecht. 2023. Conversational repair strategies to cope with errors and breakdowns in customer service chatbot conversations. In *International Workshop on Chatbot Research and Design*. Springer, 23–41.

[11] Veronica Cateté, Nicholas Lytle, Yihuan Dong, Danielle Boulden, Bita Akram, Jennifer Houchins, Tiffany Barnes, Eric Wiebe, James Lester, Bradford Mott, et al. 2018. Infusing computational thinking into middle grade science classrooms: lessons learned. In *Proceedings of the 13th workshop in primary and secondary computing education*. ACM, New York, NY, 1–6.

[12] Ismail Celik, Hanni Muukkonen, and Signe Siklander. 2025. Teacher–Artificial Intelligence (AI) interaction: The role of trust, subjective norm and innovativeness in Teachersacceptance of educational chatbots. *Policy Futures in Education* (2025), 14782103251348551.

[13] Yu Chen, Scott Jensen, Leslie J Albert, Sambhav Gupta, and Terri Lee. 2023. Artificial intelligence (AI) student assistants in the classroom: Designing chatbots to support student success. *Information Systems Frontiers* 25, 1 (2023), 161–182.

[14] Michelene TH Chi and Ruth Wylie. 2014. The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational psychologist* 49, 4 (2014), 219–243.

[15] Alan Cooper. 1999. The inmates are running the asylum. In *Software-ergonomie'99: design von informationswelten*. Springer, 17–17.

[16] Andrea Cuadra, Maria Wang, Lynn Andrea Stein, Malte F Jung, Nicola Dell, Deborah Estrin, and James A Landay. 2024. The illusion of empathy? notes on displays of emotion in human-computer interaction. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 1–high18.

[17] Charlene M Czerniak, Andrew T Lumpe, Jodi J Haney, and Judy Beck. 1999. Teachers' beliefs about using educational technology in the science classroom. *International Journal of Educational Technology* 1, 2 (1999), 1–18.

[18] Yun Dai, Ziyan Lin, Ang Liu, Dan Dai, and Wenlan Wang. 2024. Effect of an analogy-based approach of artificial intelligence pedagogy in upper primary schools. *Journal of Educational Computing Research* 61, 8 (2024), 1695–1722.

[19] Fred D Davis. 1985. *A technology acceptance model for empirically testing new end-user information systems: Theory and results*. Ph. D. Dissertation. Massachusetts Institute of Technology.

[20] Fred D Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly* (1989), 319–340.

[21] Andre del Carpio Gutierrez, Paul Denny, and Andrew Luxton-Reilly. 2024. Automating Personalized Parsons Problems with Customized Contexts and Concepts. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. ACM, ACM, New York, NY, 688–694. doi:10.1145/3649217.3653568

[22] Kai Deng. 2025. Evaluating the Effectiveness of Large Language Models in Solving Simple Programming Tasks: A User-Centered Study. (2025). doi:10.48550/ARXIV.2507.04043

[23] Stefania Druga and Amy J Ko. 2025. Scratch Copilot: Supporting Youth Creative Coding with AI. In *Proceedings of the 24th Interaction Design and Children*. ACM, New York, NY, 140–153.

[24] Yael Feldman-Maggor, Mutlu Cukurova, Carmel Kent, and Giora Alexandron. 2025. The Impact of Explainable AI on Teachers' Trust and Acceptance of AI EdTech Recommendations: The Power of Domain-specific Explanations. *International Journal of Artificial Intelligence in Education* (2025), 1–34.

[25] Jennifer Fereday and Eimear Muir-Cochrane. 2006. Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International journal of qualitative methods* 5, 1 (2006), 80–92.

[26] Bai Gao, Ruisi Liu, and Junjie Chu. 2025. Exploring Trends of Acceptance of Artificial Intelligence in Education: A Systematic Literature Review. In *Artificial Intelligence in HCI*, Helmut Degen and Stavroula Ntoa (Eds.). Springer Nature Switzerland, Cham, 196–213.

[27] Michael Gerlich. 2025. AI tools in society: Impacts on cognitive offloading and the future of critical thinking. *Societies* 15, 1 (2025), 6.

[28] Brian Harvey, Daniel D Garcia, Tiffany Barnes, Nathaniel Titterton, Daniel Armendariz, Luke Segars, Eugene Lemon, Sean Morris, and Josh Paley. 2013. Snap!(build your own blocks). In *Proceeding of the 44th ACM technical symposium on Computer science education*. ACM, NY, New York, 759–759.

[29] Emma Harvey, Allison Koenecke, and Rene F Kizilcec. 2025. " Don't Forget the Teachers": Towards an Educator-Centered Understanding of Harms from Large Language Models in Education. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 1–19.

[30] Arto Hellas, Juho Leinonen, Sami Sarsa, Charles Koutcheme, Lilja Kujanpää, and Juha Sorva. 2023. Exploring the Responses of Large Language Models to Beginner Programmers' Help Requests. In *Proceedings of the 2023 ACM Conference on International Computing Education Research V.1*. ACM, New York, NY, 93–105. doi:10.1145/3568813.3600139

[31] Sunyoung Hlee, Jaehyun Park, Hyunsun Park, Chulmo Koo, and Younghoon Chang. 2023. Understanding customer's meaningful engagement with AI-powered service robots. *Information Technology & People* 36, 3 (2023), 1020–1047.

[32] Sebastian Hobert. 2023. Fostering skills with chatbot-based digital tutors–training programming skills in a field study. *i-com* 22, 2 (2023), 143–159.

[33] Kenneth Holstein, Gena Hong, Mera Tegene, Bruce M. McLaren, and Vincent Aleven. 2018. The classroom as a dashboard: co-designing wearable cognitive augmentation for K-12 teachers. In *Proceedings of the 8th International Conference on Learning Analytics and Knowledge* (Sydney, New South Wales, Australia) *(LAK '18)*. Association for Computing Machinery, New York, NY, USA, 79–88. doi:10.1145/3170358.3170377

[34] Andreas Holzinger, Michaela Kargl, Bettina Kipperer, Peter Regitnig, Markus Plass, and Heimo Müller. 2022. Personas for artificial intelligence (AI) an open source toolbox. *IEEE Access* 10 (2022), 23732–23747.

[35] Xinying Hou, Zihan Wu, Xu Wang, and Barbara J. Ericson. 2024. CodeTailor: LLM-Powered Personalized Parsons Puzzles for Engaging Support While Learning Programming. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale*. ACM, New York, NY, 51–62. doi:10.1145/3657604.3662032

[36] Xinying Hou, Zihan Wu, Xu Wang, and Barbara J. Ericson. 2025. Personalized Parsons Puzzles as Scaffolding Enhance Practice Engagement Over Just Showing LLM-Powered Solutions. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2*. ACM, New York, NY, 1483–1484. doi:10.1145/3641555.3705227

[37] Tong Hu and Songzan Wang. 2025. From Generation to Adaptation: Comparing AI-Assisted Strategies in High School Programming Education. (2025). doi:10.48550/ARXIV.2506.15955

[38] Weijiao Huang and Khe Foon Hew. 2025. Facilitating Online Self-Regulated Learning and Social Presence Using Chatbots: Evidence-Based Design Principles. *IEEE Transactions on Learning Technologies* 18 (2025), 56–71. doi:10.1109/TLT.2024.3523199

[39] Stanislav Ivanov, Mohammad Soliman, Aarni Tuomi, Nasser Alhamar Alkathiri, and Alamir N Al-Alawi. 2024. Drivers of generative AI adoption in higher education through the lens of the Theory of Planned Behaviour. *Technology in Society* 77 (2024), 102521.

[40] Jinglu Jiang, Alexander J Karran, Constantinos K Coursaris, Pierre-Majorique Léger, and Joerg Beringer. 2023. A situation awareness perspective on human-AI interaction: Tensions and opportunities. *International Journal of Human–Computer Interaction* 39, 9 (2023), 1789–1806.

[41] Jinchuan Jiao and Xiangnan Cao. 2024. Research on designers' behavioral intention toward artificial Intelligence-Aided design: Integrating the theory of planned behavior and the technology acceptance model. *Frontiers in Psychology* 15 (2024), 1450717.

[42] Hyeon Jo and Do-Hyung Park. 2023. Affordance, usefulness, enjoyment, and aesthetics in sustaining virtual reality engagement. *Scientific Reports* 13, 1 (2023), 15097.

[43] Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences* 103 (2023), 102274.

[44] Andrew Kemp, Edward Palmer, Peter Strelan, and Helen Thompson. 2024. Testing a novel extended educational technology acceptance model using student attitudes towards virtual classrooms. *British Journal of Educational Technology* 55, 5 (2024), 2110–2131.

[45] HYEJI KIM, Jongyoul Park, Hyeongbae Jeon, Sidney S. Fels, Samuel Dodson, and Kyoungwon Seo. 2025. Augmented Educators and AI: Shaping the Future of Human-AI Collaboration in Learning. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 769, 6 pages. doi:10.1145/3706599.3706720

[46] Jinhee Kim. 2024. Leading teachers' perspective on teacher-AI collaboration in education. *Education and information technologies* 29, 7 (2024), 8693–8724.

[47] Harsh Kumar, Ilya Musabirov, Mohi Reza, Jiakai Shi, Xinyuan Wang, Joseph Jay Williams, Anastasia Kuzminykh, and Michael Liut. 2024. Guiding Students in Using LLMs in Supported Learning Environments: Effects on Interaction Dynamics, Learner Performance, Confidence, and Trust. *Proceedings of the ACM on Human-Computer Interaction* 8, CSCW2 (2024), 1–30.

[48] Dong-Kyu Lee and Inwhee Joe. 2025. A GPT-Based Code Review System With Accurate Feedback for Programming Education. *IEEE Access* 13 (2025), 105724–105737. doi:10.1109/access.2025.3581139

[49] Chien Hsiang Liao, Chang-Tang Chiang, I-Chuan Chen, and Kevin R Parker. 2022. Exploring the relationship between computational thinking and learning satisfaction for non-STEM college students. *International Journal of Educational Technology in Higher Education* 19, 1 (2022), 43.

[50] Mark Liffiton, Brad E Sheese, Jaromir Savelka, and Paul Denny. 2023. CodeHelp: Using Large Language Models with Guardrails for Scalable Support in Programming Classes. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*. ACM, New York, NY, 1–11. doi:10.1145/3631802.3631830

[51] Ally Limke, Saminur Islam, Bahare Riahi, Xiaoyi Tian, Marnie Hill, Veronica Cateté, and Tiffany Barnes. 2025. What Does It Take to Support Problem Solving in Programming Classrooms? A New Framework from the K-12 Teacher Perspective. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 591, 7 pages. doi:10.1145/3706599.3719763

[52] Phoebe Lin and Jessica Van Brummelen. 2021. Engaging teachers to co-design integrated AI curriculum for K-12 classrooms. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. 1–12.

[53] Sanzida Mojib Luna, Jiangnan Xu, Garreth W Tigwell, Nicolas LaLone, Michael Saker, Alan Chamberlain, David I Schwartz, and Konstantinos Papangelis. 2025. Exploring Deaf And Hard of Hearing Peoples' Perspectives On Tasks In Augmented Reality: Interacting With 3D Objects And Instructional Comprehension. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. ACM, NY, New York, 1–14.

[54] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. 2010. The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)* 10, 4 (2010), 1–15.

[55] Punya Mishra and Matthew J Koehler. 2006. Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers college record* 108, 6 (2006), 1017–1054.

[56] Svenja Mohr and Rainer Kühl. 2021. Acceptance of artificial intelligence in German agriculture: an application of the technology acceptance model and the theory of planned behavior. *Precision Agriculture* 22, 6 (2021), 1816–1844.

[57] Alexander Tobias Neumann, Yue Yin, Sulayman Sowe, Stefan Decker, and Matthias Jarke. 2024. An llm-driven chatbot in higher education for databases and information systems. *IEEE Transactions on Education* (2024).

[58] Quynh Hoa Nguyen. 2023. AI and Plagiarism: Opinion from Teachers, Administrators and Policymakers, In Proceedings of the 20th AsiaCALL International Conference (AsiaCALL2023). *Proceedings of the AsiaCALL International Conference* 4, 75–85. doi:10.54855/paic.2346

[59] Don Norman. 2007. *Emotional design: Why we love (or hate) everyday things*. Basic books.

[60] OpenAI. 2025. Introducing study mode. https://openai.com/index/chatgpt-study-mode/. Accessed: 2025-09-10.

[61] Evan W Patton, Michael Tissenbaum, and Farzeen Harunani. 2019. MIT app inventor: Objectives, design, and development. In *Computational thinking education*. Springer Singapore Singapore, 31–49.

[62] W. Powell and S. Courchesne. 2024. Opportunities and risks involved in using ChatGPT to create first grade science lesson plans. *PLOS ONE* 19, 6 (2024), e0305337. doi:10.1371/journal.pone.0305337

[63] Snehal Prabhudesai, Ananya Prashant Kasi, Anmol Mansingh, Anindya Das Antar, Hua Shen, and Nikola Banovic. 2025. "Here the GPT made a choice, and every choice can be biased": How Students Critically Engage with LLMs through End-User Auditing Activity. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 1015, 23 pages. doi:10.1145/3706598.3713714

[64] Prerna Ravi, John Masla, Gisella Kakoti, Grace C. Lin, Emma Anderson, Matt Taylor, Anastasia K. Ostrowski, Cynthia Breazeal, Eric Klopfer, and Hal Abelson. 2025. Co-designing Large Language Model Tools for Project-Based Learning with K12 Educators. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 138, 25 pages. doi:10.1145/3706598.3713971

[65] H. Reichert and F. OtherAuthors. 2024. Empowering secondary school teachers: Creating, executing, and evaluating a transformative professional development course on ChatGPT. In *2024 IEEE Frontiers in Education Conference (FIE)*. IEEE, Washington, DC, USA, 1–9. doi:10.1109/FIE61694.2024.10893106

[66] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.

[67] Bahare Riahi and Veronica Cateté. 2025. Comparative Analysis of STEM and Non-STEM Teachers' Needs for Integrating AI into Educational Environments. In *International Conference on Human-Computer Interaction*. Springer, 125–140.

[68] Lianne Roest, Hieke Keuning, and Johan Jeuring. 2024. Next-Step Hint Generation for Introductory Programming Using Large Language Models. In *Proceedings of the 26th Australasian Computing Education Conference*. ACM, New York, NY, 144–153. doi:10.1145/3636243.3636259

[69] Ido Roll and Ruth Wylie. 2016. Evolution and revolution in artificial intelligence in education. *International journal of artificial intelligence in education* 26, 2 (2016), 582–599.

[70] Sallimah Salleh and Peter Albion. 2004. Using the theory of planned behaviour to predict Bruneian teachers' intentions to use ICT in teaching. In *Society for Information Technology & Teacher Education International Conference*. Association for the Advancement of Computing in Education (AACE), 1389–1396.

[71] Lindsay Sanneman and Julie A Shah. 2022. The situation awareness framework for explainable AI (SAFE-AI) and human factors considerations for XAI systems. *International Journal of Human–Computer Interaction* 38, 18-20 (2022), 1772–1788.

[72] Ghadeer Sawalha, Imran Taj, and Abdulhadi Shoufan. 2024. Analyzing student prompts and their effect on ChatGPT's performance. *Cogent Education* 11, 1 (2024), 2397200.

[73] Niklas Scholz, Manh Hung Nguyen, Adish Singla, and Tomohiro Nagashima. 2025. Partnering with AI: A Pedagogical Feedback System for LLM Integration into Programming Education. (2025). doi:10.48550/ARXIV.2507.00406

[74] B. Shneiderman. 2020. Human-centered artificial intelligence: Three fresh ideas. *AIS Transactions on Human-Computer Interaction* 12, 3 (2020), 109–124. doi:10.17705/1thci.00131

[75] Shristi Shrestha and Jiyeong Yi. 2025. TPACK-based Professional Development for the AI Era: Fostering Pre-service Teachers' Acceptance of Generative AI in Mathematics Classrooms. (2025).

[76] Paweł Robert Smolinski, Joseph Januszewicz, and Jacek Winiarski. 2024. Scaling technology acceptance analysis with Large Language Model (LLM) annotation systems: A validation study. (2024).

[77] Veronica A Thurmond. 2001. The point of triangulation. *Journal of nursing scholarship* 33, 3 (2001), 253–258.

[78] Stefano Triberti, Alice Chirico, Gemma La Rocca, and Giuseppe Riva. 2017. Developing emotional design: Emotions as cognitive processes and their role in the design of interactive technologies. *Frontiers in psychology* 8 (2017), 1773.

[79] Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics* 51, 1 (2025), 275–338.

[80] S. J. H. Yang, H. Ogata, T. Matsui, and N. S. Chen. 2021. Human-centered artificial intelligence in education: Seeing the invisible through the visible. *Computers and Education: Artificial Intelligence* 2 (2021), Article 100008. doi:10.1016/j.caeai.2021.100008

[81] Yaxuan Yin, Shamya Karumbaiah, and Shona Acquaye. 2025. Responsible AI in Education: Understanding Teachers' Priorities and Contextual Challenges. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency.* 2705–2727.

[82] Habeeb Yusuf, Arthur Money, and Damon Daylamani-Zad. 2025. Towards reducing teacher burden in Performance-Based assessments using aivaluate: An emotionally intelligent LLM-Augmented pedagogical AI conversational agent. *Education and Information Technologies* (2025), 1–45.

[83] Olaf Zawacki-Richter, Victoria I Marín, Melissa Bond, and Franziska Gouverneur. 2019. Systematic review of research on artificial intelligence applications in higher education–where are the educators? *International journal of educational technology in higher education* 16, 1 (2019), 1–27.

[84] N. Zhou, H. Nguyen, C. Fischer, D. Richardson, and M. Warschauer. 2020. High school teachers' self-efficacy in teaching computer science. *ACM Transactions on Computing Education* 20, 3 (2020), Article 23. doi:10.1145/3410631