Human–AI Interaction for Accessible CS Learning: Co-Designing AI with Blind and Visually Impaired Learners

Shi Ding
Georgia Institute of Technology
Atlanta, Georgia, USA

Jason Brent Smith
Northwestern University
Evanston, Illinois, USA

Kevin Gautier
University of North Texas
Denton, Texas, USA

Stephen Garrett
Georgia Institute of Technology
Atlanta, Georgia, USA

Brian Magerko
Georgia Institute of Technology
Atlanta, Georgia, USA

Jason Freeman
Georgia Institute of Technology
Atlanta, Georgia, USA

Bryan Pardo
Northwestern University
Evanston, Illinois, USA

Stephanie Ludi
University of North Texas
Denton, Texas, USA

Taneisha Lee
The Findings Group
Decatur, Georgia, USA

Tom McKlin
The Findings Group
Decatur, Georgia, USA

Abstract

Generative AI is increasingly used in learning environments, but most systems focus on generating responses rather than supporting interaction. This limitation is especially important for blind and visually impaired (BVI) learners, who rely on non-visual ways to navigate and understand programming tasks. We present findings from a co-design study with BVI high school students exploring expectations for AI assistants in expressive computer science learning. Participants described how AI could support accessible interaction, guide learning, and empower learner agency. We derive two design principles for inclusive AI assistants: (1) situated multimodal interaction and (2) scaffolded iterative interaction. These principles shift the focus from adapting outputs (e.g., audio) to designing interactions that support how learners navigate and engage in the CS learning context. Our work highlights the importance of interaction design in accessible AI and the value of involving BVI learners in the design of AI-powered educational technologies.

CCS Concepts

• **Human-centered computing → Human computer interaction (HCI)**; *Systems and tools for interaction design*; *Accessibility*; • **Applied computing → Education**.

Keywords

Generative AI, Human–AI Interaction, Accessible Computing, Educational Technology

ACM Reference Format:

Shi Ding, Jason Brent Smith, Kevin Gautier, Stephen Garrett, Brian Magerko, Jason Freeman, Bryan Pardo, Stephanie Ludi, Taneisha Lee, and Tom McKlin. 2026. Human–AI Interaction for Accessible CS Learning: Co-Designing AI with Blind and Visually Impaired Learners. In *Proceedings of the 25th Interaction Design and Children Conference (IDC '26), June 22–25, 2026, Brighton, United Kingdom.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3773077.3812141

ACM ISBN 979-8-4007-2283-7/2026/06
https://doi.org/10.1145/3773077.3812141

1 Introduction

Generative AI systems are increasingly integrated into learning environments, supporting tasks such as coding assistance [8, 15] and feedback generation [23, 34]. However, current LLM tools are primarily designed as response generators [1, 32], offering limited support for interaction-level needs such as navigation and contextual awareness. This results in a mismatch with how dynamic learning unfolds in practice, where learners engage in iterative, multi-step workflows. This gap is especially significant for blind and visually impaired (BVI) learners, who rely on structured, non-visual interaction through screen readers, keyboard navigation, and spatial descriptions. While accessibility is often approached as adapting output modalities (e.g., audio or Braille), less attention has been paid to how AI supports interaction in non-visual learning environments. To address this gap, we conduct a co-design study with BVI high school students in an expressive programming environment. We ask: *What design considerations emerge when co-designing AI assistants with BVI learners in CS learning environments?* This paper makes two contributions: (1) We derive key challenges and expectations for AI assistants through co-design with BVI learners. (2) We synthesize these findings into design principles that position accessibility as an interaction-level design problem that supports how learners navigate, understand, and engage with programming tasks.

2 Background and Related Work

2.1 Human–AI Interaction in CS Learning

Generative AI (GenAI) systems are increasingly used to support coding tasks such as code generation, explanation, and debugging

[15, 22, 23]. These tools can reduce cognitive load and support problem solving (e.g., editable prompts) [13], but also raise concerns about overreliance and reduced critical thinking [2, 21, 35, 43]. Prior work highlights the importance of interaction design in shaping learning outcomes, including AI-assisted debugging[5, 10, 28], and teachable agents [30]. However, most prior work focuses on general CS learning contexts and uses AI systems primarily as response generators [24, 27]. Less attention has been paid to interaction-level support, such as navigation support and contextual awareness. While recent studies in accessibility have explored improving AI responses in terms of precision, interpretability, and contextual relevance [3, 11, 19, 38, 47], these approaches remain underexplored in CS learning settings, particularly in non-visual interaction contexts. This gap motivates the need for design-oriented approaches that examine how AI assistants can support real learning workflows for BVI learners.

2.2 Accessibility in AI for BVI Learners

BVI learners face persistent challenges in CS education, where visual representations such as diagrams are often inaccessible [29, 45]. While digital platforms provide features such as screen reader support and audio descriptions, many CS learning tools still lack sufficient accessibility, limiting their usability for BVI students [14, 17]. Prior work has explored accessible programming tools, AI-supported interfaces, and multimodal interaction techniques [6, 7, 20, 36, 37, 48], as well as context-aware AI systems for navigation and workflow support [18, 26, 39]. Other studies examine how AI supports information retrieval for curriculum alignment [4, 12, 46]. However, support for BVI learners in CS learning remains limited, with ongoing challenges in AI integration and assistive technology (AT) compatibility [16, 48]. While prior work explores accessibility features and AI-supported interfaces, fewer studies examine how these systems support interaction-level workflows in CS learning contexts. This gap is especially critical in CS classrooms, where non-visual interaction plays a critical role. To address this gap, we engage BVI learners in co-design to explore their needs in an AI-assisted programming context and derive design insights for inclusive human–AI interaction.

3 Method: Context and Study Design

We adopted a co-design approach to engage BVI learners in shaping AI assistant design, enabling us to capture situated needs and accessibility challenges [9, 17, 41, 42]. The study was conducted using the EarSketch expressive programming environment (Figure 1), a widely used CS learning tool with over 1.5 million learners that teaches Python and JavaScript through music [31]. EarSketch provides an expressive learning context where users engage in iterative workflows combining coding and creative exploration, making it a specific setting to examine interaction-level AI support in non-visual learning environments. Although EarSketch does not include generative AI features, it provides a baseline environment for examining how AI can be integrated to support interaction within existing learning workflows.

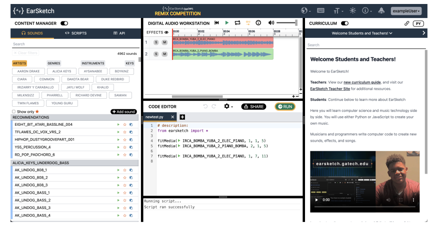

Figure 1: Overview of the EarSketch web client interface. Learners search for sounds using the Sound Browser (left), write code in the Code Editor (bottom), and play their projects through the Digital Audio Workstation (top), while accessing the Curriculum (right).

ID	Vis	AI	G	AT
BS1	LV	Heard	NB	Mag
BS2	Blind	Some	F	Braille
BS3	LV	Freq	M	Mag
BS4	LV	Some	F	Fusion
BS5	Blind	Some	M	SR

Table 1: BVI student participants. LV = low vision; G = gender; SR = screen reader, AT = assistive technology

We recruited five BVI high school students (three low-vision, two blind) with prior experience using EarSketch through the Microsoft TEALS curriculum [1]. Students used assistive technologies including screen magnifiers, Fusion (JAWS/ZoomText), or Braille keyboards. Two teachers (math and CS/TEALS) participated as contextual stakeholders to provide insight into classroom practices and AI use. Before the workshop, we administered a short pre-survey to teachers to understand students' programming practices and AI use in the classroom. We then conducted co-design sessions using speculative design and role-play activities to explore learners' expectations for a generative AI assistant in EarSketch. Table 1 summarizes participant demographics, two teachers also participated to support the co-design sessions and provide contextual insights.

We conducted a multi-stage qualitative analysis combining thematic coding and affinity mapping [33] of workshop transcripts, student artifacts, and researchers' field notes. An initial AI-focused codebook was developed using a mixed inductive–deductive approach [33]. Themes derived from affinity mapping closely aligned with those identified through coding, further supporting the consistency and reliability of the analysis. To assess coding reliability, three researchers independently coded a subset of 207 quotes, achieving strong agreement (Fleiss' κ = 0.83). Disagreements occurred on 6 instances (2.9%). Remaining disagreements were resolved through discussion and minor refinements to the codebook.

4 Themes and Findings

4.1 Context-Aware Multimodal Interaction

Participants emphasized that AI assistance should be *context-aware*, aligning with a learner's situated task [25, 40, 44], interface state, and curriculum context. Participants also noted that *limited-modality*

[1]https://tealsk12.github.io/2nd-semester-introduction-to-computer-science/

feedback (e.g., text-only responses) is insufficient for BVI learners, highlighting the need for multimodal interaction that supports non-visual navigation.

4.1.1 Context-Aware Curriculum Guidance.

Students valued AI support that aligns with the curriculum (e.g., unit goals, lesson objectives) rather than producing generic responses. Participants suggested that AI should reference instructional materials, identify misconceptions, and provide targeted examples that connect to ongoing lessons. As BS3 explained, *"it matters where the AI is getting its information from. It should understand the curriculum and summarize it, 'According to the curriculum. . . ' so then later students get verified information rather than random outputs."* Others (BS1, BS4) expressed that AI should act as a tutor by identifying misconceptions, referencing relevant concepts, and offering targeted examples rather than finished solutions. As BS3 noted, *"AI should be able to dig deeper into specific areas and provide examples, just like a good tutor would."*

Participants also highlighted the importance of *context-sensitive guidance* within EarSketch. Students suggested that AI assistants could provide code-aware debugging suggestions, explain programming concepts within the context of the current project, and support accessible navigation of the interface. For example, BS5 noted that *"it would be helpful in debugging... It would be cool to have an LLM read the screen and provide information about screen context."* Teachers expressed concern that current tools lack contextual awareness. Teacher BT2 discussed how an AI can *feed* on itself and degrade in quality: *"(AI in a CS class) is probably not gonna eat itself enough that it can't explain dictionaries anymore. But who knows? I think that it still has a long way to go as far as being able to contextualize information."*

4.1.2 Multimodal Interaction.

Students BS3 described using voice interaction to explore interface elements, request debugging assistance, and locate information within the coding workspace. Learners used voice queries such as asking what elements were located on different parts of the interface or requesting playback from specific measures. *"It can describe photos, which is great for visual learners and screen readers. I use AI voice (Google AI studio) to locate code, offer suggestions, debug, and navigate websites."* Students valued *integration of voice, keyboard shortcuts, and accessible text editor positions.* They used voice to build spatial awareness of the UI (e.g., "what's on the left side?"), request control playback (e.g., "play measure 10"), and receive precise feedback about where changes occurred.

Students (BS3, BS5) also emphasized the need for non-visual spatial references when interacting with AI systems. For example, BS3 described the use of voice interaction to explore the elements of the interface and understand the structure of the workspace: *"Voice feels more natural, like talking to a real person."* This interaction helps learners to articulate problems more naturally and supports real-time reasoning during coding tasks. Instead of cursor-based references, students preferred clear structural location descriptions aligned with the programming environment. As BS4 noted, *"Non-visual students don't use cursors — say the measure, track, and time."* These interaction observations helped learners maintain awareness of where changes occurred within their projects and supported independent navigation in EarSketch.

4.2 Scaffolded Human–AI Collaboration

Participants expressed a desire for AI assistants in EarSketch to provide teacher-like support while preserving learners' agency in the creative and problem-solving process. Rather than acting as an answer generator, participants described AI as a "thinking partner" that supports navigation, concept understanding, and debugging while allowing learners to construct coding solutions themselves. Their ideal interactions with an AI within creative learning environments require scaffolded interaction structures that support problem-solving processes while empowering learner agency.

4.2.1 Prompting and Debugging.

Participants highlighted the importance of scaffolded interactions that help learners articulate problems and reason through solutions. Teacher BT1 emphasized that effective prompting is an important part of learning how to collaborate with AI. For example, weak prompts such as "What is this?" provide little learning value, while structured questions such as "What Python concept should I use to make this repeat?" help students share their intent and receive more meaningful guidance.

Students also noted that AI explanations sometimes misalign with their level of understanding. For example, student BS2 described receiving explanations that were too complex for novice learners, suggesting the need for adjustable explanation levels that match students' current knowledge. *"I understood it, but I wasn't sure if my friend would. I could have asked it to explain it more simply."* At the same time, participants valued AI's ability to encourage exploration. As student BS3 noted, AI sometimes introduces new ideas that teachers may not always have time to explore in depth. BS3 added that they *"would rather get new concepts from AI because they give more encouragement at the expense of efficiency... in ways a teacher might not always have time for."*

Participants also preferred debugging interactions that follow a scaffolded dialogue structure. Instead of immediately applying fixes, they suggested that AI should first help localize the issue, explain the cause in plain language, and then offer possible solutions before applying changes. For example, students (BS2, BS3) preferred precise pointers as feedback such as: *"On line 13, the variable* `drm_loop` *doesn't exist, did you mean* `drum_loop`*?"* followed by an option to apply the correction. This explain-first approach supported both conceptual learning and independent problem solving.

Teacher BT1 further noted that BVI learners often develop computational thinking skills as they learn to navigate assistive technologies, meaning that instructional support must scaffold both programming concepts and interaction with tools: *"It's also true that a lot of that has to happen at the same time...they're learning how to use a computer to a different capacity when they're learning to code as well.*

4.2.2 Agency and Overreach in Human–AI Collaborative Learning.

Participants emphasized that AI should support, rather than replace, human-centered learning interactions. Students described how peer discussions often provide diverse strategies and encourage deeper understanding, while AI tends to provide more standardized responses. As BS4 noted, peers often explain how they arrived at a solution rather than simply providing an answer. *" Peer support is less get straight to the fact, they share their experiences and how they got to something".* Student BS5 also added, *"Talking to people* *your own age can be helpful. They can share how they approached the material, which might inspire you to try something similar, adapt it in your own way, or take a completely new path."*

Teacher BT1 similarly highlights the value of peer learning: *"In computer science, there's a strong emphasis on efficiency, but in learning, inefficiency is fine... I'd often prefer to learn a new concept from a peer rather than AI, because AI might skip foundational ideas."* In this context, participants suggested that AI should adapt to different roles during collaboration with accessible signifiers, for example, acting as a brainstorming partner during early creative exploration and as a tutor during skill-building tasks.

Students explained that an AI that provides complete solutions constitutes *pedagogical overreach*, diminishing learners' creative control and learning. As BS5 explained, *"Would you want AI to give you the entire song? No. You won't learn anything."* Similarly, BS4 noted that *"AI autofill of any kind could be as problematic as autocomplete."* Teacher BT1 emphasized that AI should be used as a supplement rather than as an authoritative source, encouraging students to use AI to generate ideas rather than code: *"I worry about AI replacing learning instead of enhancing it. Sometimes it spits out examples that are too close to the answer, and students miss the opportunity to engage with the problem-solving process."*

Beyond pedagogical overreach, participants raised concerns about the boundaries of AI assistance, including privacy and sensitive information. They emphasized that AI should clearly communicate its limits and use safeguards to maintain learner control. As BS3 suggested, *"AI should warn against sensitive requests and recognize private information automatically. For example, if someone asks, 'What's my username?' the AI should respond, 'I'm sorry, I can't share that.'"*

5 Design Principles for Accessible AI Assistants

Building on our analysis, we synthesize these findings into design principles for accessible AI assistants in learning environments. Our results point to a shift from response-level support to interaction-level design. Instead of only providing alternative outputs (e.g., audio or Braille), AI emerges as an interaction layer that supports navigation, understanding, and workflow in non-visual programming. While situated in the EarSketch context, these principles reflect broader interaction patterns and extend to accessible AI-supported learning systems beyond a single platform, with implications for more inclusive learning environments. Together, they capture complementary aspects of interaction design: how AI provides context-aware support (Principle 1) and how it structures the learning process over time (Principle 2).

Principle 1: Situated Multimodal Interaction AI assistants should provide guidance aligned with the learner's goals, code state, and interface context. Instead of generating generic responses, systems should offer feedback that reflects ongoing tasks and support accessible interaction for diverse learners (e.g., screen-reader). This is particularly critical for BVI learners, who rely on non-visual interaction to navigate and understand programming environments. Such support helps learners understand both the structure of the environment and how their actions relate to ongoing tasks.

Principle 2: Scaffolded Iterative Interaction AI assistants should support scaffolded collaboration rather than provide one-time answers. Systems should guide learners through iterative cycles of problem solving by localizing issues, explaining underlying concepts, and refine their approach step by step. This interaction maintains student agency and supports accessible workflows for programming tasks. Such behavior can be supported through interaction design strategies (e.g., prompting) that guide step-by-step reasoning and iterative refinement.

6 Conclusion and Future Work

We present findings from a co-design study with BVI learners, identifying key interaction needs for accessible AI-assisted CS learning. We contribute two design principles that shift the focus from output adaptation to interaction design. Although drawn from an EarSketch case study, these findings suggest implications for broader accessible AI-supported learning environments. This work suggests accessibility in AI as an interaction design challenge, complementing a focus on AI outputs and shaping how learners engage with and make sense of complex tasks. Future work will develop and evaluate an accessible AI assistant in real-world classroom settings.

Acknowledgments

This research was supported by the National Science Foundation awards 2300631, 2300632, and 2300633. The views expressed are those of the authors and do not necessarily reflect those of the NSF. EarSketch is available online at https://earsketch.gatech.edu/.

6.1 Selection and Participation of Children

This study involved five blind and visually impaired (BVI) high school students and two teachers from the Washington School for the Blind. Students had varying levels of visual acuity and prior experience with assistive technologies such as screen readers. Participants were recruited in collaboration with school teachers to ensure accessibility and equitable participation. All activities were designed to be inclusive, allowing the use of screen reader–compatible materials, verbal instructions, and flexible formats. Sessions were conducted with accessibility support (e.g., screen readers, Braille displays, and magnification tools), with two teachers present to support the learning environment. Participation was voluntary, with informed consent obtained from students and their parents. Participants were informed about the study purpose, procedures, and their right to withdraw at any time. The study was approved by the Institutional Review Board (IRB) at the Georgia Institute of Technology, and all participants received appropriate compensation. After the co-design sessions, we will maintain contact with participants and conduct a follow-up co-design evaluation in Summer 2026. Study findings will be shared in accessible formats with participants, educators, and the school community during these sessions. We collected observation notes, student artifacts, and audio recordings of teacher interviews during the co-design activities. No personally identifiable information was recorded. All data were stored securely and handled to ensure privacy and confidentiality. The co-design approach enabled participants to contribute to the design of AI tools for CS learning and to inform the development of accessible and inclusive learning environments.

References

[1] Vibhor Agarwal, Madhav Krishan Garg, Sahiti Dharmavaram, and Dhruv Kumar. 2024. " Which LLM should I use?": Evaluating LLMs for tasks performed by Undergraduate Computer Science Students. *arXiv preprint arXiv:2402.01687* (2024).

[2] Pedro Alves and Bruno Pereira Cipriano. 2024. " Give me the code"–Log Analysis of First-Year CS Students' Interactions With GPT. *arXiv preprint arXiv:2411.17855* (2024).

[3] LiYan Bu, YiZhuo Hou, WeiCheng Pan, Hong Chen, BoYu Cui, and HeSen Li. 2026. AI vs. traditional navigation systems: enhancing efficiency and reducing cognitive burden for visually impaired users. *Interactive Learning Environments* 34, 1 (2026), 67–97.

[4] Virgínia P Campos, Luiz MG Gonçalves, Wesnydy L Ribeiro, Tiago MU Araújo, Thaís G Do Rego, Pedro HV Figueiredo, Suanny FS Vieira, Thiago FS Costa, Caio C Moraes, Alexandre CS Cruz, et al. 2023. Machine generation of audio description for blind and visually impaired people. *ACM Transactions on Accessible Computing* 16, 2 (2023), 1–28.

[5] Elizabeth Carter and Glenn D Blank. 2013. An intelligent tutoring system to teach debugging. In *International Conference on Artificial Intelligence in Education*. Springer, 872–875.

[6] R Chandramma, Kunda Suresh Babu, KS Ranjith, Akash Kumar Sinha, Vikram Neerugatti, and D Somashekhara Reddy. 2025. Enhancing E-learning Accessibility through AI (Artificial Intelligence) and Inclusive Design. In *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*. IEEE, 1466–1471.

[7] Prashant Chandrasekar, Mariel Couvillion, Ayshwarya Saktheeswaran, and Jessica Zeitz. 2025. LLM impact on BLV programming. *arXiv preprint arXiv:2504.17018* (2025).

[8] Angxuan Chen, Yuang Wei, Huixiao Le, and Yan Zhang. 2026. Learning by teaching with ChatGPT: The effect of teachable ChatGPT agent on programming education. *British Journal of Educational Technology* 57, 1 (2026), 163–184.

[9] Sasha Costanza-Chock. 2020. *Design justice: Community-led practices to build the worlds we need.* The MIT Press.

[10] Taufiq Daryanto, Xiaohan Ding, Kaike Ping, Lance T Wilhelm, Yan Chen, Chris Brown, and Eugenia H Rho. 2026. Human-Human-AI Triadic Programming: Uncovering the Role of AI Agent and the Value of Human Partner in Collaborative Learning. *arXiv preprint arXiv:2601.12134* (2026).

[11] Shi Ding, Jason Brent Smith, Stephen Garrett, and Brian Magerko. 2024. Redesigning EarSketch for Inclusive CS Education: A Participatory Design Approach. In *Proceedings of the 23rd Annual ACM Interaction Design and Children Conference*. 720–724.

[12] Shi Ding, Jason Brent Smith, and Brian Magerko. 2025. Considering Large Language Model Integration in Expressive Computer Science Learning Environments for Blind and Visually Impaired Learners Through Co-design. In *International Conference on Artificial Intelligence in Education*. Springer, 472–480.

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

[14] Kimberly M Edwards and Satasha L Green. 2014. Working with learners with visual impairments in STEM. *STEM education: Strategies for teaching learners with special needs* (2014), 143–155.

[15] Ty Feng, Sa Liu, and Dipak Ghosal. 2024. Courseassist: Pedagogically appropriate ai tutor for computer science education. In *Proceedings of the 2024 on ACM Virtual Global Computing Education Conference V. 2*. 310–311.

[16] Paul DS Fink, Jessica A Holz, and Nicholas A Giudice. 2021. Fully autonomous vehicles for people with visual impairment: Policy, accessibility, and future directions. *ACM Transactions on Accessible Computing (TACCESS)* 14, 3 (2021), 1–17.

[17] Batya Friedman, Peter Kahn, and Alan Borning. 2002. Value sensitive design: Theory and methods. *University of Washington technical report* 2, 8 (2002), 1–8.

[18] Mohammed-Khalil Ghali, Abdelrahman Farrag, Daehan Won, and Yu Jin. 2025. Enhancing knowledge retrieval with in-context learning and semantic search through generative AI. *Knowledge-Based Systems* 311 (2025), 113047.

[19] Emma Goldenthal, Jennifer Park, Sunny X Liu, Hannah Mieczkowski, and Jeffrey T Hancock. 2021. Not all AI are equal: Exploring the accessibility of AI-mediated communication technology. *Computers in Human Behavior* 125 (2021), 106975.

[20] Ricardo E Gonzalez Penuela, Ruiying Hu, Sharon Lin, Tanisha Shende, and Shiri Azenkot. 2025. Towards Understanding the Use of MLLM-Enabled Applications for Visual Interpretation by Blind and Low Vision People. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–8.

[21] Hacer Güner and Erkan Er. 2025. AI in the classroom: Exploring students' interaction with ChatGPT in programming learning. *Education and Information Technologies* (2025), 1–27.

[22] Brian Harrington, Ahmad Zubair Alnoor, Pedram Haqiqi, Zahra Hoseininia, Kai Lin, Maliha Lodi, Asad Mirza, Leah Wolfe, and Kevin Zhang. 2025. A Systematic Literature Mapping of Early Generative AI Research is CS Education. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2*. 1467–1468.

[23] Peng He and Joseph Krajcik. 2026. Artificial intelligence in science education: global insights and future directions. *Disciplinary and Interdisciplinary Science Education Research* 8, 1 (2026), 6.

[24] Owen Henkel, Hannah Horne-Robinson, Nessie Kozhakhmetova, and Amanda Lee. 2024. Effective and Scalable Math Support: Evidence on the Impact of an AI- Tutor on Math Achievement in Ghana. arXiv:2402.09809 [cs.HC] https://arxiv.org/abs/2402.09809

[25] Wayne Holmes, Maya Bialik, and Charles Fadel. 2019. *Artificial intelligence in education promises and implications for teaching and learning.* Center for Curriculum Redesign.

[26] Ugochukwu Francis Ikwuanusi, Peter Adeyemo Adepoju, and Chinekwu Somtochukwu Odionu. 2023. AI-driven solutions for personalized knowledge dissemination and inclusive library user experiences. *International Journal of Engineering Research Updates* 4, 2 (2023), 052–062.

[27] Arif Jawaid, Memoona Batool, Waseem Arshad, Muhammad Ikram ul Haq, Parveen Kaur, and Sadia Sanaullah. 2025. AI and English language learning outcomes. *Contemporary Journal of Social Science Review* 3, 1 (2025), 927–935.

[28] Manu Kapur. 2008. Productive failure. *Cognition and instruction* 26, 3 (2008), 379–424.

[29] Karen E Koehler and Kimberley M Picard. 2024. Making STEM accessible for students with visual impairments: Implications for practice. *TEACHING Exceptional Children* (2024), 00400599241231211.

[30] Qianou Ma, Hua Shen, Kenneth Koedinger, and Sherry Tongshuang Wu. 2024. How to teach programming in the ai era? using llms as a teachable agent for debugging. In *International Conference on Artificial Intelligence in Education*. Springer, 265–279.

[31] Brian Magerko, Jason Freeman, Tom Mcklin, Mike Reilly, Elise Livingston, Scott Mccoid, and Andrea Crews-Brown. 2016. Earsketch: A steam-based approach for underrepresented populations in high school computer science education. *ACM Transactions on Computing Education (TOCE)* 16, 4 (2016), 1–25.

[32] Bradley McDanel and Ed Novak. 2025. Designing LLM-resistant programming assignments: Insights and strategies for cs educators. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*. 756–762.

[33] Matthew B Miles, A Michael Huberman, and Johnny Saldana. 2014. Qualitative data analysis: A methods sourcebook. *(No Title)* (2014).

[34] Carrie Anne Philbin and Sue Sentance. 2025. Student Use and Teacher Practice: A Scoping Review of Generative AI in Computing Education Using PICRAT. In *Proceedings of the 2025 Conference on UK and Ireland Computing Education Research (UKICER '25)*. Association for Computing Machinery, New York, NY, USA, Article 13, 7 pages. doi:10.1145/3754508.3754513

[35] Mitchel Resnick. 2024. Generative AI and creative learning: Concerns, opportunities, and choices. (2024).

[36] Clark Saben, Jessica Zeitz, and Prashant Chandrasekar. 2024. Enabling Blind and Low-Vision (BLV) Developers with LLM-Driven Code Debugging. *Journal of Computing Sciences in Colleges* 40, 3 (2024), 204–215.

[37] Abir Saha, Thomas Barlow McHugh, and Anne Marie Piper. 2023. Tutoria11y: Enhancing Accessible Interactive Tutorial Creation by Blind Audio Producers. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.

[38] Mazen Salous, Daniel Lange, Timo Von Reeken, Maria Wolters, Wilko Heuten, Susane Boll, and Larbi Abdenebaoui. 2025. Semi-Automatic BVI Human-Centered Image Conversational Descriptions: Leveraging LLMs and Expert Refinements for Inclusive Visual Accessibility. *IEEE Access* (2025).

[39] Saumik Shashwat, Parth Barthwal, Deep Sharma, Richa Gupta, and Shourya Pathak. 2024. Talktile: Accessible Educational Tool to Augment Independent Learning for Students with BVI. In *International Conference on Human-Computer Interaction*. Springer, 159–173.

[40] Kristen Shinohara and Jacob O Wobbrock. 2016. Self-conscious or self-confident? A diary study conceptualizing the social accessibility of assistive technology. *ACM Transactions on Accessible Computing (TACCESS)* 8, 2 (2016), 1–31.

[41] Ben Shneiderman. 2022. *Human-centered AI.* Oxford University Press.

[42] Marc Steen. 2013. Co-design as a process of joint inquiry and imagination. *Design issues* 29, 2 (2013), 16–28.

[43] Arne Styve, Outi T Virkki, and Usman Naeem. 2024. Developing critical thinking practices interwoven with generative AI usage in an introductory programming course. In *2024 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 01–08.

[44] Lucille Alice Suchman. 1987. *Plans and situated actions: The problem of human-machine communication.* Cambridge university press.

[45] Sarah E Wegwerth, Gianna J Manchester, and Julia E Winter. 2023. A feasibility study of the kasi learning system to support independent use of stem diagrams by students with visual impairments. *Journal of Visual Impairment & Blindness* 117, 2 (2023), 162–174.

[46] Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Hong Lin. 2023. Ai-generated content (aigc): A survey. *arXiv preprint arXiv:2304.06632* (2023).

[47] Haoning Xue, Yoo Jung Oh, Xinyi Zhou, Xinyu Zhang, and Berit Oxley. 2026. Users' prompting strategies and ChatGPT's contextual adaptation shape conversational information-seeking experiences. *Scientific Reports* (2026).

[48] Crystal Yang and Paul Taele. 2025. AI for Accessible Education: Personalized Audio-Based Learning for Blind Students. *arXiv preprint arXiv:2504.17117* (2025).