

CONJUROR: A GenAI-Powered Tool to Support Teachers Customizing a Scratch Curriculum

Minh Tran*
ngminhtran@uchicago.edu
University of Chicago
Chicago, IL, USA

David Gonzalez-Maldonado*
dagm@uchicago.edu
University of Chicago
Chicago, IL, USA

Diana Franklin
dmfranklin@uchicago.edu
University of Chicago
Chicago, IL, USA

Abstract

While many computer science (CS) curricula exist for K-8 classrooms (students ages 9-13), curriculum design has been largely led by research groups and organizations. To empower teachers to adapt CS curricular materials to their local contexts, recent research has explored ways to support teacher-led curriculum design.

This paper introduces CONJUROR, the first generative AI tool (to our best knowledge) that assists teachers in every step of customizing a fully-featured lesson (with a Scratch project and student-facing worksheets) for a CS curriculum. We describe how CONJUROR uses AI to scaffold and streamline the process while providing teacher agency to leverage their expertise and knowledge. We then present results from 44 elementary and middle school teachers using CONJUROR during a professional development workshop. Using interaction logs and survey data, we found that CONJUROR was time efficient and had high teacher-perceived output quality. In addition, features for teacher agency led to a variety of teacher use patterns.

CCS Concepts

• **Social and professional topics** → K-12 education; Computer science education; • **Human-centered computing** → **Interactive systems and tools**.

Keywords

K-8, curriculum customization, generative AI, Scratch programming

ACM Reference Format:

Minh Tran, David Gonzalez-Maldonado, and Diana Franklin. 2026. CONJUROR: A GenAI-Powered Tool to Support Teachers Customizing a Scratch Curriculum. In *Proceedings of the 31st ACM Conference on Innovation and Technology in Computer Science Education V. 1 (ITiCSE 2026)*, July 10–15, 2026, Madrid, Spain. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3803400.3809362>

1 Introduction

While there is a growing library of introductory computer science (CS) curricula that situate in learners' contexts, no single curriculum can resonate with *all* learners. Teachers, with intimate knowledge of their own classrooms [24, 27, 35], are best positioned to sustain existing curricula by customizing them to their local contexts. One

*Equal contribution.



This work is licensed under a Creative Commons Attribution International 4.0 License.

ITiCSE 2026, Madrid, Spain

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

ACM ISBN 979-8-4007-2634-7/2026/07

<https://doi.org/10.1145/3803400.3809362>

approach to customize a CS curriculum involves teachers choosing a theme that connects with their students, and incorporating it into the technical materials, such as sample Scratch projects and explanations of CS concepts [10, 12, 40]. However, customizing instructional materials can be time-intensive and challenging, given the many time constraints teachers face [7, 38].

This paper introduces CONJUROR, an AI-powered tool that helps teachers create customized materials that align with a Use-Modify-Create Scratch curriculum [13]. The tool provides structure and suggestions throughout the process (from brainstorming themes and project ideas to creating a functional Scratch project and student worksheets). Our overarching goal is to support teachers in (1) producing high-quality materials in a short time (less than 30 minutes) so it is feasible to complete during normal class preparation time, while (2) leveraging teachers' expertise about their students, schools, and local communities, and (3) allowing them to remain in control of how and where AI-generated content is adopted.

We present the design and evaluation of CONJUROR (RQ1, RQ2), as well as explore how teachers use such a tool (RQ3). Following a curriculum workshop, teachers spent 1.5 days learning about lesson customization and working with CONJUROR. The study was guided by the following research questions:

- **RQ1:** Was the customization process *time efficient*?
- **RQ2:** To what extent did teachers perceive the *quality* of AI-generated suggestions?
- **RQ3:** What were the *patterns* in how teachers interacted with the tool and its AI-generated suggestions?

2 Background and Related Work

We first situate our work at the intersection of K-8 CS curriculum development and the emerging uses of generative AI (GenAI) for CS class preparation. We then discuss the tension between teacher agency and automation before justifying our approach to design a sustainable AI tool to augment curriculum customization process.

2.1 K-8 CS Curriculum Development

The expansion of CS in K-8 has led to a growing library of fully-featured curricula designed for young CS learners [3, 13, 19]. However, much work on CS curriculum development has relied on expertise outside of the classroom, often led by researchers through research-practitioner partnerships [5, 6, 16, 20, 30]. While resulting in curricula that align to state standards and incorporate the cultures and experiences of the target populations, prepackaged materials cannot always resonate with *all* students. Teachers know their students, schools, and communities best and therefore can customize a curriculum to their local contexts [10, 14, 37]. However, curriculum customization can be challenging, especially in

CS where many teachers do not have both sufficient content and pedagogical knowledge [42]. Further, teachers typically do not have the bandwidth to fully use their localized expertise to inform the design of the materials they teach [7, 38].

To support teacher customization, some researchers have identified challenges teachers face when customizing Scratch-based materials from an introductory CS curriculum (e.g., excessive time investment, adding complexity) and co-designed scaffolds for them (e.g., step-by-step guidance, planning sheets) [40].

2.2 Generative AI for CS Class Preparation

Preparing classroom materials is laborious, even for experienced instructors. These materials include both instructor-facing materials (e.g., sample solutions, test code) and student-facing materials (e.g., coding assignments, code explanations).

At the college level: GenAI has been used to help CS instructors create classroom materials in bulk, such as worked examples [22], programming exercises [21, 34], and code explanations [23, 26] for text-based languages. Another way CS instructors can leverage GenAI is in the creation of materials on demand by students, such as contextually personalized programming problem descriptions and code solutions [8, 25] or personalized Parson puzzles [18].

At the K-8 level: While GenAI has been an intriguing topic [32], very little work has explicitly studied *how GenAI can support K-8 CS teachers in preparing Scratch classroom materials*. Prior studies found that while GenAI can help teachers brainstorm creative Scratch project ideas, they often suffer from code complexity, syntax, grading inaccuracies, and cultural insensitivity issues when used for curriculum design or auto-grading [15, 39].

2.3 Teacher Agency vs Automation

Despite potential benefits, the integration of GenAI into classroom preparation presents a tension: as AI systems become more capable of automating routine pedagogical work, automation may undermine teacher agency [4, 28]. This provides a lens through which researchers and tool designers can situate their interventions: *AI tools should “complement” (not replace) teachers’ expertise and responsibilities* [17, 36]. Endorsing this design approach, Ravi et al. conducted a study where they co-designed GenAI tools with teachers and proposed positioning GenAI as a brainstorming partner and workflow scaffold [33]. Teachers should play an active role; they specify contextual anchors (e.g., learning goals, selected standards) for the AI to generate project ideas and lesson components that they would then revise rather than accept wholesale. The design of our tool is guided by this human-AI complementary approach.

2.4 Design-Based Implementation Research

Design-Based Implementation Research (DBIR) is an approach to organize research projects to address practical challenges that require a substantial solution or change in practices [31]. It emphasizes (1) a focus on practical problems from stakeholders’ perspectives, (2) iteratively co-designing with stakeholders, (3) development of knowledge and theory to support the solution implementation, and (4) promoting a sustainable and scalable change [11]. Guided

by DBIR principles, our work aims to address the persistent time-management challenges in curriculum customization with a sustainable and scalable technological solution by engaging teachers in the design and evaluation of CONJUROR (detailed in Section 3.2).

3 Introducing CONJUROR

We begin by describing the lesson customization process, followed by our approach to develop an AI tool to streamline the customization. We then detail CONJUROR components and functionalities.

3.1 Lesson Customization Process

We use Scratch Encore, a Use-Modify-Create Scratch CS curriculum for upper elementary and middle school classrooms (students ages 9-13) [13], as the base curriculum for CONJUROR. This curriculum was chosen because for each module, it provides three sample Use-Modify projects and accompanying student-facing worksheets for teachers to choose from. The three sample projects have nearly identical code with different themes. Each module, teachers first introduce the CS topic/concept. Students then explore and modify the code of the provided sample Use-Modify project before applying that concept to implement their own Create project.

The purpose of CONJUROR is to customize this sample project to a teacher’s local context. For example, Figure 1 depicts an original sample project in Scratch Encore featuring a cute race between a bee and a snake with a stationary animated monkey. A teacher customized this to be a winter bazaar (a local event and his students’ favorite holiday – Christmas) with a moving train and horse and carriage with a stationary animated snowman.



Figure 1: Original vs teacher-customized lesson materials

An efficient method for customizing a Scratch Encore module involves several steps. First, the teacher chooses a module (e.g., Animation) and a theme they want to integrate into the module’s lesson materials (e.g., Christmas, pest patrol on Minnesota corn farms). They then think of a scenario involving the new theme that has *the same movements and actions as the original/base sample projects provided by the curriculum*. Next, the teacher remixes a base project (using the ‘remix’ option in Scratch) and “reskins” it (swapping out sprites and backdrop with new ones related to the new theme), reusing the base project’s code with some minor changes such as image size, sprites’ starting location, and text/number inside some blocks. Finally, they update the student worksheets (e.g., change images, edit sprite names) to match their customized project.

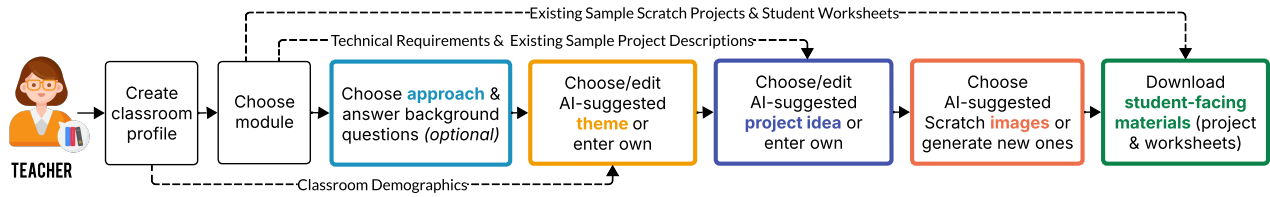


Figure 2: Lesson customization process with CONJUROR

3.2 Tool Design and Implementation

CONJUROR emerged from a multi-year research project that empowers K-8 educators to create customized CS materials for their classrooms [40]. Initial phases revealed that while teachers valued customization and were able to achieve it through scaffolding and professional development (PD) support, manual customization was unsustainable during the school year due to time constraints. Harnessing the breakthroughs in GenAI, this work has culminated in CONJUROR. The tool was designed to mitigate several drawbacks in using GenAI tools in the K-12 setting. Our key design goals are:

- **Retain Teacher Agency:** To address concerns of culturally insensitive results [1, 9], teachers should make final decisions; the CONJUROR interface should permit (1) directly using, (2) editing, and (3) replacing generated output.
- **Reuse Existing Code:** CONJUROR reuses the code of existing projects in the base curriculum to avoid errors in AI-generated Scratch code [15]. Generated project ideas closely match with the base project so that they can be implemented by merely swapping out graphical assets.
- **Require No Prompting:** To support novice GenAI users [2, 29, 41], AI prompts are integrated into the tool and specific to each Scratch Encore module¹. CONJUROR serves as an interface for teachers to provide details to refine the prompts.

We piloted early versions of CONJUROR with two cohorts of teachers, recorded their feedback, and iteratively improved the tool and implemented new features to support teachers' use patterns.

3.3 Tool Overview

Figure 2 depicts the workflow in CONJUROR. The tool guides teachers through each step, prompting them for inputs, offering suggestions, and asking them to accept, edit, or reject the suggestions.

(0) **Setup:** The first time a teacher uses CONJUROR, they create a classroom profile with geographical and demographic characteristics of their classroom and students.

(1) **Module:** The teacher selects a module to customize.

(2) **Approach:** The teacher selects from three customization approaches²: (a) local / school contexts or students' interests, (b) holidays or students' cultural traditions, and (c) an academic subject.

(3) **Theme:** CONJUROR combines the classroom profile and approach-specific questions to generate 10 theme suggestions.

(4) **Project Idea:** CONJUROR suggests four project ideas aligned with both the chosen theme and the technical requirements of the chosen module. Each idea includes a title, a description of the backdrop, and descriptions of each sprite's attributes and actions.

(5) **Images:** CONJUROR first suggests sprite and backdrop images from the Scratch asset library. The teacher may then choose to generate new images, and they are given the option to edit a pre-populated prompt. For human sprites, the teacher can also select modifiers for gender, age group, and skin, hair, and eye color to better reflect their students' demographics.

(6) **Student-Facing Materials:** CONJUROR integrates the selected images into a base project from Scratch Encore. The base project's code is reused; only images are swapped out. The teacher can upload the resulting project to Scratch and make additional minor edits as needed (e.g., add sprite costumes, adjust sprite size and starting position, or change values in code blocks). CONJUROR also integrates sprite images and names into the student worksheets (Google Docs and Forms) and emails them to the teacher.

In selecting AI models, we used OpenAI's GPT-4o mini for text and Stability AI's Stable Image Core for images. Both were cost-effective and offered the rate limits needed to support multiple teachers using the tool simultaneously at the time of our study.

4 Methods

To evaluate CONJUROR, we conducted an IRB-approved user study integrated into a PD workshop. This section details the participants, training, and specific tasks teachers performed using the tool.

4.1 Participants and Training Procedure

Our multi-day PD workshop was designed to ensure all participants had a foundational understanding of the base curriculum before introducing CONJUROR. The PD involved two cohorts of teachers from across the U.S. *Cohort 1:* 15 teachers who had previously taught Scratch Encore recruited via curriculum download registration. *Cohort 2:* 29 teachers who had taught Scratch, but not Scratch Encore, recruited via mailing lists for educators across the country.

Curriculum Workshop To establish a baseline of curriculum knowledge, 29 new teachers attended a 3-day intensive workshop covering the pedagogical approaches and materials of Scratch Encore, 15 experienced teachers received a half-day refresher session.

CONJUROR Workshop Both cohorts then combined for a 1.5-day workshop focused on using CONJUROR to customize Scratch Encore materials to reflect the characteristics of their students or existing curricular priorities at their school. The CONJUROR workshop was split into two portions: introduction and hands-on use.

Introduction (0.5 Day) We first defined and discussed motivating factors for lesson customization. Teachers then did a brainstorming exercise to generate potential themes and project ideas themselves, before being shown a step-by-step demonstration of CONJUROR.

Hands-on Use (1 Day) Finally, teachers customized two Scratch Encore modules with CONJUROR: Animation and Conditional Loops.

¹Some prompt details and evaluation results are published in a different paper [39].

²Approaches based on teachers' use patterns observed in early pilot studies.

Animation Module: After a live CONJUROR demo and tutorial on refining a generated Scratch project, teachers used CONJUROR to create and refine a customized Animation project. Next, they had a live tutorial on refining the generated student worksheets. Teachers then practiced refining their customized Animation worksheets.

Conditional Loops Module: In the final PD session, teachers applied the full workflow independently. They used CONJUROR to customize the Conditional Loops module. Teachers were tasked with refining both the generated Scratch project and the accompanying student worksheets to ensure they were classroom-ready.

4.2 Data Collection and Analysis

CONJUROR was instrumented to log all teacher interactions with the tool, enabling complete reconstruction of each teacher’s customization workflow from initial login to project completion.

4.2.1 Logged Events and Attributes. The logging system captured the following information for each interaction:

Action Type: The specific action taken by the teacher (e.g., answering background questions, accepting or editing suggestions, generating new images, downloading a final project).

Content Origin: Whether the content presented was generated by CONJUROR or directly inputted by the teacher, allowing us to distinguish between teacher-provided ideas and AI-generated ideas.

Modification History: All edits and refinements made to AI-generated suggestions, including the number of revisions and the specific elements modified (e.g., theme descriptions, sprite attributes).

Image Generation: The frequency and context of image generation requests, including how many times a teacher regenerated image options, whether they selected from suggested Scratch images suggestions or generated new AI images, and any prompt modifications made to the image generation interface.

Time Metrics: Timestamps for all actions, enabling calculation of time spent on individual components and full start to finish use.

Navigation Flow: The sequence of steps taken, including any deviations from the standard workflow (e.g., revisiting prior steps).

4.2.2 User Feedback. Before downloading their projects, teachers were prompted to provide feedback on the quality of AI outputs in CONJUROR using three 5-point Likert items. They rated the (1) theme suggestions, (2) project idea suggestions, and (3) AI-generated images on a scale ranging from ‘Very Poor’ to ‘Very Good.’

4.2.3 User Flow Reconstruction. The complete interaction log for each teacher permits detailed analysis of their customization pathway. By replaying the sequence of logged events, we can reconstruct the exact state of each teacher’s work at any point in time, including which suggestions were retained, refined, or replaced, and how teachers navigated decision points within the tool. This fine-grained data enables analysis of user behavior patterns.

5 Results

This section presents findings from our evaluation of CONJUROR. We first evaluate the efficiency of the customization process (RQ1), followed by teacher perceptions of the quality of AI-generated content (RQ2). Finally, we describe emergent patterns in teachers’ interactions with the AI tool during our PD workshop (RQ3).

5.1 Time Efficiency

Finding 1: The majority of customization sessions (92.3%) were completed within 30 minutes (not counting offline refinements).

On average, teachers spent less than 15 minutes on from the moment they loaded the tool to when they downloaded the project and worksheets. Table 1 summarizes the detailed time-on-task. Specifically, 51/78 sessions (65.4%) took less than 15 minutes, 21/78 (26.9%) took between 15 and 30 minutes, and 6/78 (7.7%) took between 30 and 45 minutes. There were two main sources of longer sessions. First, the load caused slower responses to AI generation requests, especially image generation. Second, teachers could enter breakout rooms to ask questions and chat with facilitators during the PD.

Table 1: Distribution of total customizing time and average time teachers spent on major components of CONJUROR.

Total Time	Count	Component	Avg. Time
< 15 min.	51	Theme	2.05 min.
15-30 min.	21	Project Idea	3.29 min.
30-45 min.	6	Images	8.83 min.

A detailed look at the individual components shows that image generation took the most time (8.83 minutes). Creating the classroom profile only took about 25 seconds, and this step was required only in teachers’ first use of CONJUROR. Theme and project idea generation took an average of 2.05 and 3.29 minutes respectively.

5.2 Output Quality

Finding 2: Teachers generally perceived AI suggestions as meeting or exceeding expectations across all three components.

Figure 3 depicts the teachers’ ratings of AI suggestion quality across three major components of CONJUROR. Theme and project idea quality was more favorable than image quality. In particular, the majority of the ratings for generated themes was ‘Good’ (30/75, 40%), ‘Acceptable’ (28/75, 37%), and ‘Very Good’ (15/75, 20%). Only two themes (3%) were rated as ‘Poor’ and none was rated as ‘Very Poor’. Ratings for project idea quality followed a similar pattern, with the majority falling into ‘Good’ (28/78, 36%), ‘Acceptable’ (27/78, 35%) and ‘Very Good’ (19/78, 24%). Only four generated project ideas (5%) were rated as ‘Poor’ and none was rated as ‘Very Poor’. In contrast, ratings for generated sprite and backdrop images were more mixed. While many teachers still rated image quality as ‘Acceptable’ (29/78, 37%), ‘Good’ (19/78, 24%), or ‘Very Good’ (11/78, 14%), there was a larger proportion of negative ratings: 19% (15/78) was rated ‘Poor’ and 5% (4/78) was rated as ‘Very Poor’.

Additionally, our investigation of system logs revealed complementary patterns in teacher perceptions of AI output quality. Figure 4 shows the percentage of the time a teacher proceeded to a subsequent step or returned to a previous step when getting AI suggestions from each component in CONJUROR for the first time. Specifically, teachers returned more frequently when generating images (32% total), significantly higher than when generating project ideas (12% total) and themes (1% total). The most common returning path was from the images to the project idea step (31%).

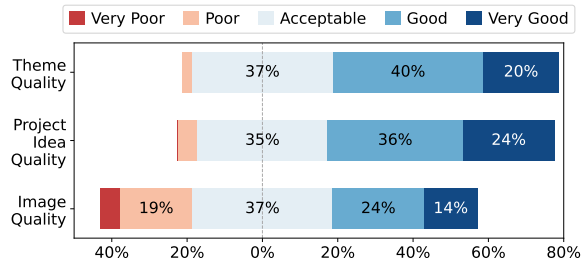


Figure 3: Teacher perceptions of AI suggestion quality

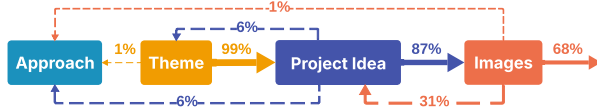


Figure 4: Overview of teacher progression when getting AI suggestions in each CONJUROR component for the first time

5.3 Usage Patterns

A major design decision was the level of teacher agency. For reasons described in Section 2, all steps in CONJUROR allowed teachers to accept, edit, or reject the AI suggestions, allowing full teacher agency. In this section, we explore how teachers used this agency.

5.3.1 Retain, Refine, or Replace? We first categorized teacher decisions relating to AI-generated suggestions into: (1) *Retain* – using the AI suggestion directly, (3) *Refine* – editing the AI suggestion, or (3) *Replace* – replacing the AI suggestion with their own idea or returning to a previous step to repeat the process. Figure 5 depicts the percentage of teacher decisions, separated by step.

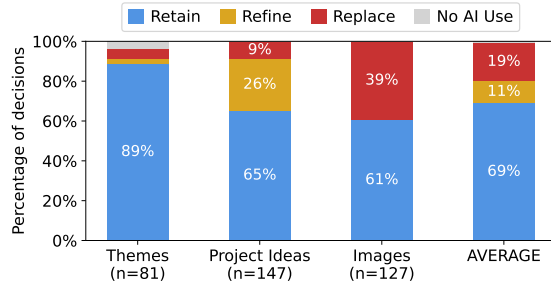


Figure 5: Percentage of accumulative decisions on AI uses teachers made within each step in CONJUROR

Finding 3: Teachers largely retained text-based AI suggestions, but they refined the more detailed project ideas more often than the high-level themes.

Theme and Project Idea Generation have similar interfaces, revealing text-based AI suggestions and allowing teachers to refine or replace them. For the Theme step, the teacher can also start with their own theme instead of using the AI. During theme generation, teachers overwhelmingly retained AI-suggested themes (72/81, 88.9%) rather than not using AI (3/81, 3.7%), refining (2/81, 2.5%), or replacing (4/81, 4.9%). Project ideas, however, are much

more detailed, with a “storyline” that places the theme into a project. Teachers refined (38/81, 25.9%) and replaced (13/147, 8.8%) many more AI-suggested project ideas than themes. However, they still most often retained the AI suggestions (96/147, 65.3%).

Finding 4: Providing image generation beyond Scratch assets is an important feature. However, there is room for improvement, as evidenced by teacher feedback presented earlier.

Image Generation uses AI in two ways. First, AI identifies the images from the Scratch asset library that most closely fit the description. A teacher can choose to retain one of those or seek a custom AI-generated image. Only if they reject the Scratch assets, do they go into the image generation interface and receive those images. *Within the tool, though, teachers can neither refine the images nor replace them with images from the Internet. However, they could do so in Scratch afterwards, changes we are unable to track.* Teachers retained the AI-suggested Scratch assets or AI-generated images 60.6% (77/127) of the time and replaced them (by returning to a previous step to repeat the process) 39.4% (50/127) of the time. Finally, though not depicted in the figure, in the final projects teachers downloaded, 54% (194/359) of the selected images were from the Scratch library and 46% (165/359) were AI-generated.

5.3.2 Customization Pathways. Not only do teachers make decisions at individual steps, they can also revisit previous steps if they are unhappy with the direction the suggestions have gone. By reconstructing user flows, we identified six approaches that influenced teachers’ customization pathways, split between *linear* (never revisiting a prior step – 60.3%) and *cyclic* (revisiting at least one prior step – 39.7%). Table 2 shows the classification of pathways teachers took, with illustrative flowcharts and statistics.

Finding 5: Teachers tended to follow a streamlined process without refining AI suggestions. However, it is important to provide the flexibility for those to want to actively curate content.

Linear AI-only was the dominant pathway (33/78 sessions, 43.3%). Teachers who followed this pathway never revisited a prior step and retained all AI suggestions. **Linear refinement** included teachers who never backtracked and refined AI suggestions or replaced them with their input every time there was an option to do so (in the Theme and Project Idea steps). No teachers in our study followed this pathway exclusively. Instead, 14/78 sessions (17.9%) followed a **hybrid linear** pathway – retaining some AI suggestions while refining or replacing others.

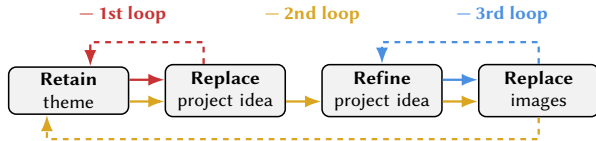
Finding 6: Teachers tended to backtrack to regenerate AI suggestions without refining their prior choices to get better results.

Cyclic repetition was the second most common pathway (22/78 sessions, 28.2%). Here, teachers revisited prior steps but always retained their original choice, only returning so that the AI would give a new set of suggestions in the later steps. **Cyclic refinement** was seen in 3/78 sessions (3.8%), and in all of them, the teachers returned to the Project Idea step and edited the project description. Finally, 6/78 sessions (7.7%) demonstrated a **hybrid cyclic** pathway – mixing both repetition and refinement when revisiting prior steps.

Table 2: Classification of teachers’ customization pathways in CONJUROR with simplified, illustrative flowcharts

Category		Illustrative Flowchart	Singular	Hybrid
Linear	AI-only	Retain AI theme → Retain AI project idea → Retain AI image suggestions	33 (43.3%)	14 (17.9%)
	Refinement	Refine AI theme → Refine AI project idea → Retain AI image suggestions	0 (0%)	
Cyclic	Repetition	Retain AI theme → Replace AI project idea → Retain AI image suggestions ↑ ----- ↓	22 (28.2%)	6 (7.7%)
	Refinement	Retain AI theme → Refine AI project idea → Replace AI image suggestions ↑ ----- ↓	3 (3.8%)	

In practice, teacher interactions with CONJUROR could look more complex than what shown in Table 2. Figure 6 illustrates a hybrid cyclic pathway taken by Rose (pseudonym), a teacher in our study. Rose revisited the Theme step twice – once from Project Idea (red) and once from Images (yellow). Both times, she attempted to generate more AI suggestions for later steps without changing her original theme (cyclic repetition). She finally revisited Project Idea once (blue) to edit the backdrop description before generating more Images (cyclic refinement). This session took about 37 minutes.

**Figure 6: A hybrid cyclic pathway taken by a teacher**

6 Discussion

Our goals for this study were to (RQ1) assess CONJUROR’s time efficiency, (RQ2) understand teacher perceptions of AI output quality, and (RQ3) explore patterns in teacher-AI interaction. We hope that this study will fill the gap and lead to further research into how GenAI can support K-8 CS teachers preparing classroom materials.

For RQ1, most teachers were able to customize within 30 minutes, even though it was their first time using CONJUROR. This time efficiency is very promising because curriculum customization is time-intensive and challenging, especially during the school year when teachers have many other responsibilities and time constraints [7, 38]. Specifically in CS, prior work has shown that even veteran teachers (with extensive Scratch experience and understanding of a curriculum) face technical and time-management challenges in customizing [40]. Having a streamlined tool that leverages both the computing powers of GenAI and the CS content and structure of an existing curriculum can significantly reduce customization time and effort, while ensuring the quality of resulting materials.

For RQ2, teachers were generally positive of the AI suggestions in CONJUROR, but favored text-based suggestions (themes and project ideas) over images. While teachers could use publicly-available GenAI models as a thought partner when preparing Scratch instructional materials, prior research has identified code complexity

and cultural insensitivity issues in AI outputs [15, 39]. CONJUROR mitigates these issues by combining expert-curated prompts with high-quality code from a structured curriculum. It also uses teachers’ classroom profiles as the context to guide the AI to generate locally relevant suggestions. Finally, the tool enforces a human-in-the-loop workflow, constantly reminding teachers to fact-check and allowing them to retain, refine, or replace AI suggestions.

For RQ3, our findings highlighted the necessity of allowing teachers to refine AI-generated content and backtrack. While linear customization pathways dominated, teachers increasingly intervened as the process got more detailed (refining project ideas – 25.9%, replacing images – 39.4%). However, advanced iterative behaviors remained rare, with few teachers using either the cyclic refinement (3.8%) or hybrid cyclic (7.7%) pathway to optimize results. This suggests that providing agency features within the tool is insufficient. In order to move away from passive repetition, teachers need explicit scaffolding to understand how editing upstream content in the workflow could effectively direct downstream AI outputs.

7 Conclusion

This paper introduces CONJUROR, the first GenAI tool that helps elementary and middle school teachers create customized lessons for a Scratch curriculum. With CONJUROR, we take a responsible first step to bring GenAI into K-8 classrooms, positioning it as a productivity tool for teachers rather than a student-facing solution. The tool both builds on teachers’ knowledge about their local communities and allows them to remain in control of how and where AI-generated content is adopted. Our study with 44 teachers demonstrated the tool’s time efficiency, positive teacher perceptions of output quality, and the value of teacher-in-the-loop features. In future work, we will conduct classroom studies to understand teacher practices when using CONJUROR in authentic settings and most importantly, its impacts on student engagement and learning.

Acknowledgements

This work is supported by the National Science Foundation under Grant No. DRL-2201313. We would like to thank Miles Brown, Jonathan Liu, and Erica Goodwin for contributing to the prompt evaluation and tool implementation. We are also grateful to David Weintrop, Jen Palmer, Andrew Chu, CS techstaffs at the University of Chicago, and the teachers for helping us run this study.

References

- [1] Selin Akgun and Christine Greenhow. 2022. Artificial intelligence in education: Addressing ethical challenges in K-12 settings. *AI and Ethics* 2, 3 (2022), 431–440.
- [2] Farhan Ali, Doris Choy, Shanti Divakaran, Hui Yong Tay, and Wenli Chen. 2023. Supporting self-directed learning and self-assessment using TeacherGAIA, a generative AI chatbot application: Learning approaches and prompt engineering. *Learning: research and practice* 9, 2 (2023), 135–147.
- [3] Karen Brennan, Christan Balch, and Michelle Chung. 2014. *Creative Computing*. <https://creativecomputing.gse.harvard.edu/guide/>
- [4] Ismail Celik, Muhterem Dindar, Hanni Muukkonen, and Sanna Järvelä. 2022. The promises and challenges of artificial intelligence for teachers: A systematic review of research. *TechTrends* 66, 4 (2022), 616–630.
- [5] Cynthia E Coburn and William R Penuel. 2016. Research–practice partnerships in education: Outcomes, dynamics, and open questions. *Educational researcher* 45, 1 (2016), 48–54.
- [6] Merijke Coenraad, Jen Palmer, Donna Eater, David Weintrop, and Diana Franklin. 2022. Using participatory design to integrate stakeholder voices in the creation of a culturally relevant computing curriculum. *International Journal of Child-Computer Interaction* 31 (2022), 100353.
- [7] Vivienne Collinson and Tanya Fedoruk Cook. 2001. “I don’t have enough time”-Teachers’ interpretations of time as a key to learning and school change. *Journal of educational administration* 39, 3 (2001), 266–281.
- [8] Andre Del Carpio Gutierrez, Paul Denny, and Andrew Luxton-Reilly. 2024. Evaluating Automatically Generated Contextualised Programming Exercises. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 289–295.
- [9] Melissa Diliberti, Heather L Schwartz, Sy Doan, Anna K Shapiro, Lydia Rainey, and Robin J Lake. 2024. *Using artificial intelligence tools in K-12 classrooms*. RAND.
- [10] Deborah A Fields, Debora Lui, and Yasmin B Kafai. 2017. Teaching computational thinking with electronic textiles: High school teachers’ contextualizing strategies in Exploring Computer Science. In *Conference proceedings of international conference on computational thinking education*. 67–72.
- [11] Barry J. Fishman, William R. Penuel, Anna-Ruth Allen, Britte Haugan Cheng, and Nora Sabelli. 2013. Design-Based Implementation Research: An Emerging Model for Transforming the Relationship of Research and Practice. *Teachers College Record* 115, 14 (Dec. 2013), 136–156. doi:10.1177/016146811311501415 Publisher: SAGE Publications.
- [12] Diana Franklin, Phillip Conrad, Gerardo Aldana, and Sarah Hough. 2011. Animal Tlatoque: Attracting Middle School Students to Computing Through Culturally-relevant Themes. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE ’11)*. ACM, New York, NY, USA, 453–458.
- [13] Diana Franklin, David Weintrop, Jennifer Palmer, Merijke Coenraad, Melissa Cobian, Kristan Beck, Andrew Rasmussen, Sue Krause, Max White, Marco Anaya, et al. 2020. Scratch Encore: The design and pilot of a culturally-relevant intermediate Scratch curriculum. In *Proceedings of the 51st ACM technical symposium on computer science education*. 794–800.
- [14] Wilson González-Espada, Pablo Llerandi-Román, Yaihara Fortis-Santiago, Giovanna Guerrero-Medina, Nicole Ortiz-Vega, Mónica Feliú-Mójer, and Daniel Colón-Ramos. 2015. Impact of culturally relevant contextualized activities on elementary and middle school students’ perceptions of science: An exploratory study. *International Journal of Science Education, Part B* 5, 2 (2015), 182–202.
- [15] David Gonzalez-Maldonado, Jonathan Liu, and Diana Franklin. 2025. Evaluating GPT for use in K-12 Block Based CS Instruction Using a Transpiler and Prompt Engineering. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*.
- [16] Joanna Goode and Jane Margolis. 2011. Exploring computer science: A case study of school reform. *ACM Transactions on Computing Education (TOCE)* 11, 2 (2011), 1–16.
- [17] Kenneth Holstein and Vincent Aleven. 2022. Designing for human–AI complementarity in K-12 education. *Ai Magazine* 43, 2 (2022), 239–248.
- [18] 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*. 51–62.
- [19] Garima Jain, Fred Martin, Bernardo Feliciano, Hsien-Yuan Hsu, Barbara Fauvel-Campbell, Gillian Bausch, Lijun Ni, and Elizabeth Thomas-Cappello. 2022. CS pathways: A culturally responsive computer science curriculum for middle school. In *2022 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–9.
- [20] Mengying Jiang and Kristin A Searle. 2024. Developing Culturally Responsive-Sustaining Computer Science Curricula: A Design-Based Research Study. In *Proceedings of the 18th International Conference of the Learning Sciences-ICLS 2024*, pp. 1630–1633. International Society of the Learning Sciences.
- [21] Mollie Jordan, Kevin Ly, and Adalbert Gerald Soosai Raj. 2024. Need a Programming Exercise Generated in Your Native Language? ChatGPT’s Got Your Back: Automatic Generation of Non-English Programming Exercises Using OpenAI GPT-3.5. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. 618–624.
- [22] Breanna Jury, Angela Lorusso, Juho Leinonen, Paul Denny, and Andrew Luxton-Reilly. 2024. Evaluating llm-generated worked examples in an introductory programming course. In *Proceedings of the 26th Australasian Computing Education Conference*. 77–86.
- [23] Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. 2023. Comparing code explanations created by students and large language models. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. 124–130.
- [24] Miranda Lin and Alan Bates. 2014. Who Is in My Classroom? Teachers Preparing to Work with Culturally Diverse Students. *International Research in Early Childhood Education* 5, 1 (2014), 27–42.
- [25] Evanfiya Logacheva, Arto Hellas, James Prather, Sami Sarsa, and Juho Leinonen. 2024. Evaluating Contextually Personalized Programming Exercises Created with Generative AI. In *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*. 95–113.
- [26] Stephen MacNeil, Andrew Tran, Arto Hellas, Joanne Kim, Sami Sarsa, Paul Denny, Seth Bernstein, and Juho Leinonen. 2023. Experiences from using code explanations generated by large language models in a web software development e-book. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 931–937.
- [27] Eleonora Marucci, Beau Oldenburg, and Davide Barrera. 2018. Do teachers know their students? Examining teacher attunement in secondary schools. *School Psychology International* 39, 4 (2018), 416–432.
- [28] Joseph Mintz, Wayne Holmes, Leping Liu, and Maria Perez-Ortiz. 2023. Artificial Intelligence and K-12 education: Possibilities, pedagogies and risks. 325–333 pages.
- [29] Jiyeon Park and Sam Choo. 2024. Generative AI prompt engineering for educators: Practical strategies. *Journal of Special Education Technology* (2024).
- [30] William R Penuel. 2017. Research–practice partnerships as a strategy for promoting equitable science teaching and learning through leveraging everyday science. *Science Education* 101, 4 (2017), 520–525.
- [31] William R Penuel, Barry J Fishman, Britte Haugan Cheng, and Nora Sabelli. 2011. Organizing research and development at the intersection of learning, implementation, and design. *Educational researcher* 40, 7 (2011), 331–337.
- [32] James Prather, Juho Leinonen, Natalie Kiesler, Jamie Gorson Benario, Sam Lau, Stephen MacNeil, Narges Norouzi, Simone Opel, Vee Pettit, Leo Porter, et al. 2025. Beyond the Hype: A Comprehensive Review of Current Trends in Generative AI Research, Teaching Practices, and Tools. *2024 Working Group Reports on Innovation and Technology in Computer Science Education* (2025), 300–338.
- [33] 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, 1–25. doi:10.1145/3706598.3713971
- [34] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic generation of programming exercises and code explanations using large language models. In *Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1*. 27–43.
- [35] David Scheidecker and William Freeman. 2015. *Bringing out the best in students: How legendary teachers motivate kids*. Simon and Schuster.
- [36] Kyoungwon Seo, Mina Yoo, Samuel Dodson, and Sung-Hee Jin. 2025. Augmented teachers: K-12 teachers’ needs for artificial intelligence’s complementary role in personalized learning. *Journal of Research on Technology in Education* 57, 4 (2025), 876–893.
- [37] Kurt D Squire, James G MaKinster, Michael Barnett, April Lynn Luehmann, and Sasha L Barab. 2003. Designed curriculum and local culture: Acknowledging the primacy of classroom culture. *Science education* 87, 4 (2003), 468–489.
- [38] Nani Teig, Ronny Scherer, and Trude Nilsen. 2019. I know I can, but do I have the time? The role of teachers’ self-efficacy and perceived time constraints in implementing cognitive-activation strategies in science. *Frontiers in psychology* 10 (2019), 1697.
- [39] Minh Tran, David Gonzalez-Maldonado, Elaine Zhou, and Diana Franklin. 2025. Can GPT Help? Supporting Teachers to Brainstorm Customized Instructional Scratch Projects. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*.
- [40] Minh Tran, Heather Killen, Jen Palmer, David Weintrop, and Diana Franklin. 2024. Harmonizing Scratch Encore: Scaffolding K-8 Teachers in Customizing Culturally Responsive Computing Materials. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. 1335–1341.
- [41] Yoshija Walter. 2024. Embracing the future of Artificial Intelligence in the classroom: the relevance of AI literacy, prompt engineering, and critical thinking in modern education. *International Journal of Educational Technology in Higher Education* 21, 1 (2024), 15.
- [42] Aman Yadav and Marc Berges. 2019. Computer science pedagogical content knowledge: Characterizing teacher performance. *ACM Transactions on Computing Education (TOCE)* 19, 3 (2019), 1–24.