Mission

My research is in Pasteur's quadrant of research – foundational research with a large impact on practice. I choose research questions that serve this mission: *As computational thinking becomes critical for an increasing number of careers, all students, regardless of circumstances, have a right to high-quality, cohesive computer science instruction throughout K-12.* Performing research in this space requires contributions to many areas - defining learning trajectories that identify what and in what order to teach, researching methods for professional development so that teachers are knowledgeable and confident to teach the material, and exploring and developing tools for teachers and students that they can use during the teaching and learning process.

Major contributions since being appointed associate professor:

Scratch learning strategy: Our research showed that classroom-level performance in an elementary coding curriculum mirrored school-level performance on math and reading standardized tests. The *TIPP&SEE Learning Strategy* was designed to make explicit the steps to learn from example Scratch code, specifically designed for middle-school students [87]. A randomized control trial showed statistically significant improvements in performance by almost all student groups at academic risk (e.g., students with disabilities, students below grade level in reading, math, or both, and students in poverty) [91, 95, 100, 102]. This strategy has been integrated into multiple curricula and professional development workshops around the country.

CS culturally-relevant instruction: There is a tension in computer science instruction between providing enough curricular structure to result in concrete learning goals and connecting with student interests and background to increase engagement and identity. To help teachers and districts analyze curriculum for a diverse set of students, we created the *TEC rubric*[86]. We then created the *Scratch Encore* curriculum [87, 126], providing three choices for each structured project so the teacher could choose what would best resonate with their students. We then explored created guides and ran professional development workshops to help teachers customize projects for their students [128]. This work has culminated in *Conjuror*, a generative-AI-powered tool that interactively works with teachers to customize projects in about 15 minutes [119, 120, 129]. The TEC Rubric is being used at the teacher-, district-, and state-levels.

Quantum computing learning trajectories: Most quantum computing instruction occurs at the advanced undergraduate and graduate levels. I have explored what and how to convey QC concepts to novices at elementary through early college levels. We have found that using a visual representation provided scaffolding that resulted in no drop in an online course once mathematics was introduced [112]. We have also created various games, including *Qupcakery* [111], that embed QC concepts into the game mechanics. These and other learning resources became the basis of studying the learning goals and orderings for young learners, resulting in learning trajectories that can be used to inform other game and activity development. An activity on reversibility in a 3rd grade classroom revised a reversibility learning trajectory [97]. Our game development resulted in learning trajectories for superposition and measurement with three phases - game mechanics, general attributes of the QC concepts, and the concepts specifically applied to QC [127].

Algorithms metacognitive strategies: My newest project has begun to explore both culturally-relevant instruction and learning strategies in algorithms instruction. We found that, even without explicit instructions, students attempted to use four distinct strategies to implement dynamic programming problems [122]. This led to an intervention to explicitly teach two strategies, which is the subject of a study occurring right now. In addition, new work is using what we learned from Conjuror to create thematically different but technically identical algorithms word problems.

Philosophy / Historical Perspective

Because of computer science's role as an elective in K-12, much of the historical research in Computer Science Education Research (CSER) has occurred in informal learning environments. As a result, early curricula and educational tools focused on a "lifelong kindergarten," creative approach (Constructionism) in which students had agency in how they were learning and applying their computer science skills. Informal learning opportunities, however, are drivers of inequity because many students lack access to such opportunities. Among challenges (and drivers of equity) in formal learning environments are the diversity of learners to be supported and the need for assessable learning outcomes. My work contributes to knowledge on what, how, and when to introduce CS concepts in K-8. My work tackles both classical computer science concepts and quantum computing concepts. There are two defining features of my research agenda.

My research asks questions both related to **academic support and student engagement and belonging.**

How: What can we use from informal learning given the constraints of formal learning?

More specifically, how can we provide the engagement, agency, and belonging present in informal learning while providing more academic support for students who need it in a formal learning space?

What: What are the "pieces of knowledge" for CS concepts that we gloss over for older students?

When: What concepts are appropriate for students at different developmental levels?

These challenges bring up interesting research questions:

What CS concepts are accessible to children of various grade levels?
How do we balance structured tasks with student-driven projects?
How do we balance engagement and learning?
What supports are necessary and helpful for struggling students?

My group has a large focus on **creating learning resources and providing professional development for facilitators to use those resources**. This is for two reasons. First, learning is inextricably tied to the teaching resources and the facilitator. Because we are studying learning with novel approaches, we create the learning resources. We provide professional development so that we can improve the quality of teaching within our studies. Second, this substantially increases the impact of our work. For example, our curriculum, Scratch Encore, has been downloaded over three thousand times, and we have provided professional development for hundreds of teachers (including through an online EdX course that can be audited for free).

Brief Summary of Research Contributions prior to 2020 *(publications <= #86)*

I begin by summarizing my research contributions in computer science education research prior to earning tenure at the University of Chicago in January 2020.

Recruiting and Engaging Diverse Students My CSER work began as the lead PI on an NSF Broadening Participation in Computing (BPC) grant. Our goal was to understand what engages students not traditionally interested in computer science. Over three summers, we developed and refined an interdisciplinary (Mayan culture, animal conservation, art, and computer science) curriculum for middle school. Unlike most similar programs, 90% of the participants were under-represented minorities, and over 50% entered the program with no interest in computer science. When students completed the camp, there was a significant interest in computer science, and there was no statistical difference between interest of those who had prior interest in computer science and those who didn't [38]. In addition, our students demonstrated competence in several key computer science concepts, including sequential execution, event-driven programming, and message passing [47, 48]. To truly broaden participation, though, I saw the need to move to public schools.

Early CT Learning Progressions I was then the lead PI on a NSF CE21 grant to provide insight on what and when to introduce CT by exploring learning progressions for CT. Through our curriculum and development environment, our research found that students struggled understanding when initialization should occur (before or after)[68], analyzed the Scratch programming interface from an HCI perspective (and created an alternative) [59, 66], identified programming patterns that emerge within Scratch programming and the implications for future programming experiences [79], and performed a comparison of 4^{th}-6^{th} grade students to provide insight into learning trajectories for this age group[73]. My systems background was critical for this work. I blended qualitative educational research methods with systems tools to produce a feedback loop between qualitative and quantitative analysis. Hand inspection identifies code patterns that indicate conceptual understandings of a particular nature, and automated analysis answers how common such understandings were. This allowed us to identify patterns more sophisticated than in prior work.

Learning Trajectories for Everyday Computing (LTEC) (Lead PI of student learning effort). With Katie Rich, who used this project as a launching board for a PhD, I co-led the effort to create learning trajectories from existing literature, developing heuristics based on education theory for ordering learning goals that had no empirical evidence for their orderings. These are the first learning trajectories for basic CT concepts, serving as the starting point for future research to refine them. We published learning trajectories on sequence, repetition, conditionals, debugging, and decomposition [72, 74, 78, 82]. We used these in the design of an integrated Math (Fractions) / CS supplemental curriculum for 3rd and 4th grade [101].

Exploring Equity in CS Education At the time I earned my promotion, two projects related to equity in computer science were in their initial states. Our initial research showed that with a curriculum designed for informal learning environments (Creative Computing Curriculum) and modified to teach specific concepts, students' performance varied immensely, correlating with their mathematics and reading comprehension scores [83]. To combat this, we integrated a well-known pedagogical approach, Use->Modify->Create, and developed a new learning strategy, TIPP&SEE [88], to provide a roadmap for learning from existing code (the Use->Modify step). More details on the rest of that effort is below. In our Scratch Encore project, we explored how to create a more equity-driven curriculum for middle-school students. First, we developed a rubric for evaluating a computing curriculum for teacher accessibility, equity, and content [86]. We also developed activities in a participatory design session to draw out student ideas of themselves, their interests, and their communities to inform the design of our curriculum [85].

Research projects since Associate Professor appointment (publications >= 86):

TIPP&SEE Metacognitive Strategy (Lead PI). In collaboration with Cathy Thomas, a professor with expertise in reading comprehension strategies for students with disabilities from Texas State University, my team explored the relationship between non-CS skills and CS skills in order to develop learning strategies for students struggling with current curricular approaches. To combat reduce CS instruction's dependence on math and reading skills, we integrated a well-known pedagogical approach, Use->Modify->Create, and developed a new learning strategy, TIPP&SEE [88], to provide a roadmap for learning from existing code (the Use->Modify step). TIPP&SEE is inspired by reading comprehension strategies for struggling readers. A randomized control trial showed statistically significant improvements in performance by almost all student groups at academic risk (students with disabilities, students below grade level in read, math, or both, and students in poverty) [91, 95, 100, 102]. This strategy has been integrated into multiple curricula and professional development workshops around the country, well beyond our own initiatives. This effort also led to one of the first validated computational thinking assessments for upper-elementary students [98, 109] as well as an exploration of automated, personalized assessments [90], the relationship between project completion and conceptual understanding [93], and a better understanding of how and why teachers of different age groups used the TIPP&SEE strategy with their students [121].

The major impact of this work has been the development of and research into the effects of the TIPP&SEE metacognitive strategy. The TIPP&SEE strategy has not only been integrated into all curricula produced by our lab, but it is included in professional development that occurs in California and Utah by groups not affiliated with us. Viewing computer science instruction through the lens of more mature fields (e.g., reading comprehension), allows us to adapt existing effective strategies for use in computer science.

Scratch Encore (Lead PI). In collaboration with David Weintrop at the University of Maryland (previously my post-Doc) and Chicago Public Schools, we built on my early work with the interdisciplinary summer camp to explore how to build an intermediate curriculum comprised of culturally-relevant projects. The distinguishing features of Scratch Encore are a) academic support in using TIPP&SEE and b) engagement / identity support by providing three projects for teachers to choose for their classrooms [87, 126]. Unfortunately, the pandemic interrupted this work, preventing a full evaluation of the effects of the design. However, we were able to use this effort as a research vehicle for a variety of foundational research questions. First, we developed a rubric for evaluating teacher accessibility, equity, and content for computing curricula [86]. We then shared our results of using participatory design sessions to inform our curriculum [85, 104]. Then we analyzed the effectiveness of the Use Modify Create pedagogical framework [96].we found that providing a starting point for open-ended projects significantly reduced the variety (and thus student-driven decisions) of completed projects [99, 113]. We also explored students' use of planning sheets [105, 107]. We also found that planning documents that were expressed vertically, like code and text flows, as opposed to horizontally, like one would draw a timeline, led to fewer teacher mistakes when planning their multi-object synchronization events [114]. Finally, we analyzed the effect the pandemic had on students [108] and teacher professional development [103].

There are three major impacts of this project. First, the TEC rubric has been integrated into state-level teacher professional development in several states. Providing a tool for teachers and districts to better analyze potential curricula for a broad range of learners allows them to make better decisions for their students. Second, the Scratch Encore curriculum and the associated professional development resources (including an EdX online course) provides teachers with both curriculum and education in Scratch, Scratch Encore, and pedagogical approaches useful for teaching CS. Finally, our research on Use-Modify-Create, starter projects, and the effects of the pandemic generalize to other curricula, providing curriculum designers more information on how to create curriculum that is both academically supportive and speaks to student engagement and agency.

Customized instruction While Scratch Encore was a step towards classroom-level customization, it was limited - three projects cannot capture the breadth of interests in classrooms around the world. Harnessing the latest breakthroughs in generative AI, we have built an interactive tool that a teacher can use to customize one of the structured Scratch Encore projects in just 15 minutes. In the future, we could imagine such a tool in the hands of learners, allowing them a customized project on the fly. As we pursue this vision, we contribute to the body of knowledge in the capabilities and effects of using generative AI to create teaching materials in middle-school classrooms.

We began by exploring the strengths and weaknesses of ChatGPT in generating project ideas for this purpose. Ideal project ideas have three sets of requirements. First, the new project needs to be able to be implemented using almost identical code to the original project - the same number of sprites, the same movement, etc. Second, the thematic idea needs to be age appropriate, authentic, and not reinforce stereotypes about a specific culture. Finally, the images need to be high quality and appropriate (again, not reinforcing stereotypes and reflect how people might look and dress were they drawn as a cartoon). We found that with careful prompt engineering, GPT was largely successful in proposing projects that aligned with the technical requirements - but not successful enough that teachers would not need to be taught how to spot hidden complexity in the proposed project [120]. Picture generation was very successful for inanimate objects, but there were a number of issues with pictures of people. First, when a country was included in the prompt (e.g., Chinese), many of the images produced included people with traditional clothing and exaggerated facial characteristics rather than a realistic, modern image. Therefore, we designed our image generator to use skin tone only, removing all references to a country of origin. In addition, people were unrealistically conventionally beautiful - thin, trendy clothing, and beautiful faces.

Our interactive tool, Conjuror, steps teachers through each step of customization. It first gathers information about classroom characteristics and uses that to suggest a number of themes (e.g., Chicago Air & Water Show). This draws on local events, holidays related to the cultural backgrounds of the students, hobbies, and school events. Once both the original template project and the theme are chosen, Conjuror suggests four editable project ideas, specifying each sprite, the background, and the action(s) of each sprite. Finally, teachers choose between existing pictures and new AI-generated images. We are using this tool to explore a number of questions:

What are the trade-offs between teacher agency and usability in controlling the interface to the generative AI tools? Our early work with teachers using generative AI tools motivated the need for the tool to facilitate genAI calls because teachers were unable to create prompts that would produce valid project ideas. However, as teachers gain more experience with genAI tools, they may want more control to specify prompts more directly. We are currently running user studies to better understand teacher needs.

How do students and teachers respond to customized projects? Are teachers more excited about the projects, causing them to be more animated and engaging in class, ultimately leading to more excitement and learning in students? Does this affect the number of successfully completed tasks?

What types of themes do teachers choose for their classrooms? Existing teachers have expressed a desire to customize for upcoming holidays (e.g., Thanksgiving), integration with other subjects (e.g., U.S. History), or to include minority groups (e.g., reduce bullying of new Afghan refugees). We will explore these three uses and how they affect teacher and student engagement and learning.

Quantum Computing Education In 2018, I embarked on a new research area - quantum computing education research. This allowed me to combine my background in quantum computer architecture (including a Nature paper [75], among others [13, 51, 53, 61, 76, 81, 84, 92]) with the computer science education skills I had built since earning my PhD. With this work, our goal is to better understand what are age-appropriate quantum computing learning goals for middle-school learners and how we can convey those in age-appropriate, engaging ways. As with all of my research, I have contributed both through research and the creation of classroom-reading teaching resources. First, in 2020, I was a lead writer on the QIS K-12 Key Concepts workshop and document that identified 13 key concepts that are accessible and useful for pre-college students to learn. This represented the first guidance on what might be reasonable in K-12. I also produced a CS-specific version of the key concepts that chose content and wording more appropriate for computer science classrooms.

Second, my group produced a number of learning resources, including 18 zines, 1 video, 11 hands-on activities, Quander (an online game world), Collapsing Qubits (a card game), an EdX course, and a two-week python / qiskit high school module.

These practical efforts have fed into a research agenda in quantum computing education that has mirrored early research in computer science education. However, computer science education was in existence for decades before we produced learning trajectories for K-12, whereas we are designing such research-based knowledge for quantum computing from the beginning. The design process for the zines led to draft learning trajectories for several quantum concepts, which are being refined through the design and use of later activities (e.g., reversibility hands-on activity [97]). We currently have a paper in submission for learning trajectories for superposition and measurement related to the Quander game world. In addition, we explored the drop-out rate and performance for students in the EdX course based on the introduction of visual representation and mathematical representation, finding that there was no steep drop-off when mathematics was introduced, suggesting the possibility that visual representation may have provided useful scaffolding prior to introducing mathematics [112]. In addition, we analyzed middle-school and high-school performance in Qupcakery, a game in which students make short quantum circuits that match the provided input/output sets. We identified a number of challenges (such as students believing that SWAP was a double-NOT gate and confusion about the CNOT gate) [111].

Because quantum computing is not a required part of K-12 education and is unlikely to get required minutes in the near future, we are exploring the use of in-game and out-of-game incentives to encourage students to play Quander games. Essentially, we are creating an eco-system of Quander-related activities. By pairing traditional in-game rewards (e.g., daily challenges, streaks, and reward coins) with out-of-game activities such as arts and crafts, hands-on activities, physical games (e.g., Collapsing Qubits), we are hoping that more students will continue game-playing sessions outside of the facilitated activities. Museum design has struggled with how to engage participants in learning past the initial visit - our work will add to this important informal learning question.

Computer Science Theory Education I have just begun a new research program in college-level algorithms and discrete instruction for computer science students (lead PI with UIUC and Utah State University). Although the area is new, I am bringing what I learned from the Scratch Encore into college-level education - students need to be supported both academically and through personalized themes for engagement and belonging. Therefore, my research agenda has two thrusts.

The first goal is to create interventions that support students academically. We began by performing a comprehensive survey of research in the area [125] and analyzing what meta-cognitive strategies students used when they solved dynamic programming problems and where they struggled [122]. We are currently piloting explicit instruction of a meta-cognitive strategy in UChicago algorithms classes. In addition, we are breaking down complete algorithm problems (dynamic programming, divide and conquer, and graph layer) into a series of discrete, assessable steps for use in future interventions.

At the same time, we are analyze ChatGPT's effectiveness at creating technically-equivalent word problems with different themes. The goal is for a student to be able to express some of their hobbies or where they grew up and have the system generate algorithmic word problems that they can relate to. We want to understand what problem genres go well with what types of problems (e.g., one class of problems uses linear distance or time - can GPT generate time-based or distance-based problems for any theme, or are there specific theme types (like active hobbies) that are better suited for it)? We also want to identify the specific technical mismatches there are between the original problems and proposed problems.

Summary

I feel that this research is my life's calling, and I want to dedicate my time to establishing the science that will allow more learners to succeed in learning computer science at all ages. Earning tenure at the University of Chicago provided the environment that has allowed me to thrive - recruiting graduate students, teaching courses that supported my research program, and increasing the value of my letters of recommendation. During my post-tenure period (2020-present), my group has published at double the rate of my career prior to being hired in a UChicago tenured faculty position. In addition, my group does more than produce publishable results that answer fundamental questions about how students learn computing (CT, CS, and QC) concepts at K-8, as well as college-level algorithms; My research provides direct benefits to our local and national community through professional development workshops, curriculum, and curricular tools.