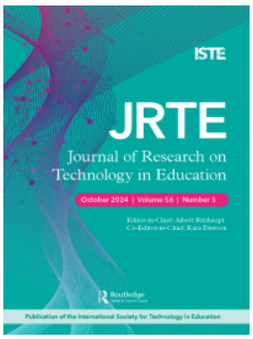

Routledge
Taylor & Francis Group

Journal of Research on Technology in Education

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ujrt20

Diving into the role: a multi-case study on supporting novice CS teachers' knowledge through co-teaching

Aman Yadav, Michael Lachney, Richard Hill, Andrew Lapetina, Anne Drew Hu, Hyein Jee & Madison C. Allen Kuyenga

To cite this article: Aman Yadav, Michael Lachney, Richard Hill, Andrew Lapetina, Anne Drew Hu, Hyein Jee & Madison C. Allen Kuyenga (25 Sep 2024): Diving into the role: a multi-case study on supporting novice CS teachers' knowledge through co-teaching, Journal of Research on Technology in Education, DOI: 10.1080/15391523.2024.2404125

To link to this article: https://doi.org/10.1080/15391523.2024.2404125

Published online: 25 Sep 2024.

JOURNAL OF RESEARCH ON TECHNOLOGY IN EDUCATION
https://doi.org/10.1080/15391523.2024.2404125

Diving into the role: a multi-case study on supporting novice CS teachers' knowledge through co-teaching

Aman Yadav[a], Michael Lachney[a], Richard Hill[b], Andrew Lapetina[b], Anne Drew Hu[a], Hyein Jee[a] and Madison C. Allen Kuyenga[a]

[a]Michigan State University, East Lansing, MI, USA; [b]University of Detroit – Mercy, Detroit, MI, USA

ABSTRACT

While computer science is widely recognized as crucial for all U.S. high school students, challenges persist for teachers' pedagogical, content, and culturally responsive-sustaining education. In this paper, we explore how a co-teaching model with an experienced CS instructor shaped three novice CS teachers' knowledge while implementing Advanced Placement Computer Science Principles courses. Using a multi-case study methodology, we analyzed semi-structured interviews, content knowledge test, and teacher created artifacts from the three teachers to compare their knowledge and experiences. We found that the teachers perceived the co-teaching model as valuable professional development, used pedagogical strategies to fill in content knowledge gaps, and struggled with culturally responsive-sustaining education. We discuss the implications of these findings for supporting novice CS teachers.

ARTICLE HISTORY
Received 30 March 2024
Revised 7 September 2024
Accepted 10 September 2024

KEYWORDS
K-12; computer science; teacher learning

Introduction

Since the growth of attention to computer science education in primary and secondary context, several countries, including the United States, have struggled to educate teachers to offer rigorous instruction. A recent landscape study found that several states in the US and countries like Ireland have challenges recruiting teachers into CS teacher preparation programs (Yadav et al., 2022). As a result, curriculum providers (e.g. Code.org and Project Lead The Way) and state and federal departments of education have used professional learning experiences to prepare teachers for implementing CS curricula. Still, the struggle to adequately educate CS teachers has led to varied student outcomes, including students from racially marginalized groups being less likely to successfully complete the course. For example, the latest available data suggests that 32% of Black students in AP Computer Science Principles (CSP) courses pass, despite the overall pass rate being 63% (Ericson, 2022). To address these disparities, there is a need for the long-term development of teacher knowledge and capacity for rigorous CS instruction.

Connected to the challenge of educating teachers, there is a growing amount of research that suggests current approaches to teaching CS, including but not limited to AP-CSP, may be exclusionary and even harmful to racially marginalized children (Margolis et al., 2017; Rankin et al., 2021). For example, Rankin et al. (2021) report on how anti-Black structures and attitudes of exclusion across the CS K-16 "pipeline" can be reproduced by teachers who are ignorant to the lived realities that, in their case, Black girls face in a highly racialized society like the United

CONTACT Aman Yadav ayadav@msu.edu Michigan State University, 620 Farm Lane, Rm 512, East Lansing, MI 48824, USA.

States. One way that CS education researchers have sought to overcome this challenge is scholarship and practices that focus on developing teachers' competencies in culturally responsive-sustaining pedagogies, which seek to center students' lived experiences within their classrooms and pedagogies (Kapor Center, 2021).

In order to explore how we might achieve these goals, we adopted a co-teaching model with an experienced university instructor to co-teach Advanced Placement Computer Science Principles (AP-CSP) courses with three novice CS teachers. During the academic year, teachers were supported in their computer science instruction by a co-teacher with a background in computer science. Given prior research has suggested that novice CS teachers lack the capacity to learn and maintain an entirely new course subject while facing other demands and feel isolated in their schools (Yadav et al., 2016), the goal of the co-teaching model was to develop teacher capacity to offer rigorous CS instruction.

To understand how the teachers themselves experienced and were affected by the co-teaching model, this paper uses a multi-case study approach to highlight how three teachers with unique backgrounds, contexts, and motivation for teaching CS navigated their teaching the CSP course. Using interview data, content knowledge measure, and teacher create artifacts collected during the 2022-2023 school year and subsequent summer, we highlight the three teachers' co-teaching experiences, self-reported CS pedagogical strategies, and how they saw the role of culturally responsive-sustaining pedagogies within their CS instruction and classrooms. We discuss some implications of this work in supporting novice CS teachers in their pursuits to provide high quality instruction to their students.

Background

Efforts and desires to prepare high school teachers in the United States with comprehensive training and education that ensures they have deep knowledge and skills necessary to teach CS effectively are not new (Yadav et al., 2022). Astrachan et al. (2011) outlined the National Science Foundation funded CS10K program, which started in 2008 with the goal of training 10,000 high school teachers to teach high quality CS education by 2017. The CS10K community persists today but in the form of CSforAll Teachers, which aims to develop, support, and sustain professional learning communities for CS teachers across the country. Throughout this period there has been much research on what teachers need to know to enact quality CS instruction. For example, Goode and Ryoo (2019) have explained how pedagogical CS knowledge cannot just be reduced to technical skills but requires a synthesis of CS content with knowledge about what pedagogical practices are best suited to support student learning, students' cultural backgrounds and community contexts, and their school systems' policies and structure. This has made research on culturally responsive-sustaining computing (CRSC)—which seeks to connect CS curriculum content to cultural and community content (Eglash et al., 2013)—increasingly important for ensuring quality CS instruction (e.g. Lachney et al., 2021), leading to exciting efforts (e.g. Exploring Computer Science and Teacher Bootstrap) to develop a robust curricular and cultural foundation for CS teachers across the country.

However, given that states across the country have varying credentialing and licensure requirements and procedures, a nationwide unified effort to build this foundation has been slow going. A 2013 report by the Computer Science Teacher Association and National Science Foundation titled "Bugs in the System," found that there are few incentives for teachers to seek out CS education credentials because most states do not require any background in CS itself (Lang et al., 2013). Thus, it is not uncommon for students to be in classrooms with teachers who have little to no CS expertise and, as a result, encounter poor instructional quality. This is not helped by the fact that even when teachers do receive training or professional development that is explicitly about CS content, these programs tend to be short-lived or even one-off, with limited opportunities for follow ups (Yadav et al., 2016). Yadav et al. (2016) found that CS teachers face

several challenges that include teaching CS content as teachers have limited CS knowledge, isolation given there is only one CS teacher in a school, and lack of adequate professional learning opportunities.

Despite these challenges, access to CS courses for students is growing across the country. For example, the current state of CS education in the U.S. reveals growth in the amount of money states are allocating to CS offerings, as well as in the number of states that have adopted or updated policies for making CS more foundational to their school systems (Code.org et al., 2023). Indeed, in 2023 over 57% of high schools in the U.S. offer at least one foundational CS course, up from approximately 53% in 2022 (Code.org et al., 2023). And across 35 states that have reported enrollment data, approximately 5.8% of students are enrolled in a foundational CS course (Code.org et al., 2023). However, racial inequities persist despite this growth in access and state attention. For example, even though Black students make up approximately 15% of the total student population in the U.S., they are only 6% of students in Advanced Placement (AP) computer science courses (Goins et al., 2022). Furthermore, AP pass rates for Black students are significantly lower than overall pass rates and pass rates of White and Asian students (Ericson, 2022). What is more, there has been continued evidence that many White CS teachers lack the racial literacies to support Black students in their classroom (Margolis et al., 2017; Goode et al., 2020).

The combination of teachers who are underprepared in terms of CS content knowledge and the literacies required to support racially marginalized students has made the availability of quality CS pedagogy limited. This has meant that many teachers do not have the expertise or skills for bringing CRSC into their classrooms. And there is evidence to suggest that when CRSC content or technologies are brought into the CS classroom, it is all too easy for the traditional curricular content to overshadow the community and cultural content (Davis et al., 2019). Deep and dynamic CRSC requires teachers who are familiar with their students' family backgrounds and communities (Lachney et al., 2021), but teachers who are already struggling to grasp the curriculum may find it even more difficult to make connections to cultural and community content that appears meaningful and authentic to children in their classrooms.

To confront these challenges, we developed a co-teaching model for working in Advanced Placement Computer Science Principles courses. This model aims to help AP-CSP teachers develop CS knowledge and cultural expertise for rigorously equitable CS instruction. The co-teaching approach places a university instructor, in this case Dr. Luke (pseudonym), in the high school teacher's classroom for an entire academic year. As the university instructor has expertise in the technical content and its pedagogy, we believed that their presence would assists in developing the high school teachers' CS content knowledge as well as their pedagogical approach through sustained interactions. The university co-instructor would achieve this by serving as a model of high-quality CS instruction, while also observing and providing feedback on the high school teacher's delivery of the content. It was the goal that these interactions would be supplemented by a limited amount of out-of-classroom development activities that the university instructor would lead the high school teacher through. Since in this approach the university instructor takes much of the lead in terms of lesson planning, teaching, and grading, the high school teacher is also provided additional time and space to focus on their own learning and development. In turn, the presence of the high school teacher would, we hoped, provide significant benefit through their knowledge and understanding of the students and their community, as well as through their ability to keep track of school policies, schedules, and administration, aspects that would be challenging for the university instructor to navigate considering they are coming from the outside for a limited amount of time. While different approaches to co-teaching and their effectiveness have been investigated (Dubek & Doyle-Jones, 2021), we believe this specific model of co-teaching is unique as it uses a job-embedded professional learning (Althauser, 2015) to develop teachers' CS content and pedagogical skill.

Job-embedded professional development

Job-embedded professional development has been suggested as an important approach to enhance teachers' instructional practices by supporting their learning within their day-to-day work environment (Balta et al., 2023). Job-embedded professional development refers to professional learning that is grounded in day-to-day teaching practice and aims to improve teachers' content-specific instructional strategies with the goal of enhancing student learning (Croft et al., 2010). It is typically school based, integrated into the workday, and involves teachers continuously assessing and solving practical problems through a cycle of improvement (Croft et al., 2010).

High-quality job-embedded professional development involves active teacher participation in cooperative, inquiry-based activities (Croft et al., 2010). Some common kinds include coaching, mentoring, peer observation, lesson study, and professional learning communities. These activities are designed to be reflective, collaborative, and directly linked to teachers' work in classrooms (Croft et al., 2010). For instance, coaching can provide real-time feedback during classroom instruction, while professional learning communities foster a culture of collaborative problem-solving and shared practices (Semon et al., 2020; Skoretz, 2011).

Effective implementation of job-embedded professional development requires supportive school leadership, adequate time, and resources. School leadership plays a crucial role in facilitating job-embedded professional development by creating a conducive school culture, providing necessary support structures, and integrating professional development into the school's routine activities (Derrington & Kirk, 2016). Additionally, successful job-embedded professional development aligns with state standards and school improvement goals, ensuring that professional development activities are relevant and targeted toward specific instructional challenges (Croft et al., 2010).

Studies have shown that job-embedded professional development can significantly improve teaching practices and student learning outcomes. For example, participation in job-embedded professional development activities has been linked to better classroom management, enhanced instructional strategies, and increased teacher self-assessment capabilities (Ernst et al., 2016). Ernst et al. (2016) investigated the effectiveness of a flexible and job-embedded professional development system for in-service technology, design, and engineering educators. The authors found that job-embedded professional development significantly improved teachers' abilities to manage, monitor, and adjust learning environments, as well as contribute to a learning community and increase self-assessment. These results suggest that job-embedded professional development can effectively enhance instructional practices and teacher proficiency in STEM education. In another study, Althauser (2015) investigated the effects of a district-wide, job-embedded professional development program on elementary teachers' personal and general mathematics teaching efficacy. The author also examined the influence of teacher efficacy on student achievement. The author found that the professional development program significantly improved teachers' efficacy, which was also a significant predictor of student achievement. This finding suggests that job-embedded professional development can positively influence student outcomes.

Research questions

The goal of this study was to examine three teachers with limited prior CS knowledge and experience teaching Advanced Placement Computer Science Principles course. The goal was to see how they view the role of co-teaching in supporting their CS knowledge and pedagogy. Our research was guided by the following research questions:

1. How do three novice CS teachers with limited content and pedagogical knowledge navigate teaching AP-CSP course, including using pedagogical approaches?
2. What role do teachers believe co-teaching played in supporting their development as CS teachers?

Methods

Multiple case study research design

To address our research questions, we employed a multi-case study research design where qualitative data (i.e. pre and post semi-structured interviews), content knowledge test, and teacher created artifacts (i.e. apps created during a summer professional development program after a year of co-teaching) were collected, analyzed, and compared from three CS teachers. We will describe these teachers below as co-teachers and treat each of the three sources of data - also described below - that were collected from them as each constituting a single case (i.e. our bounded unit of analysis). Indeed, we aim to get at the locally textured epistemic and belief shifts that took place during a year of co-teaching through the pre- and post-interviews and pre- and post-surveys, and the culmination of any shifts from evaluating the apps they created the summer following their co-teaching experiences. The overall purpose is to understand the affordances and limitations of our co-teaching method.

According to Yin (2014), a case study is an empirical investigation of a phenomenon that is constituted by the context in which it exists. In other words, the boundaries between the context and the phenomenon are always ill-defined in case studies research (e.g. epistemologies and beliefs of co-teaching that takes place in a high school AP-CSP course). We were motivated by our research questions to use a case study design because we are interested in understanding nuanced similarities and differences across the co-teachers; specifically in terms of co-teaching AP-CSP and the teachers' own epistemologies and beliefs as they are constituted by the context of the AP course and processes of co-teaching. Thus, as opposed to seeking generalizable knowledge (a limitation of a case study design), we sought to highlight the contextualized epistemologies and beliefs of each teacher as a case to open our analysis up for cross-case comparisons. Hence, our choice of a multi-case study design.

Building on this understanding of a case study, the multi-case study design that we employ in this paper is about investigating a single phenomenon (i.e. shifts in AP-CSP co-teachers' beliefs and epistemologies) that occur across multiple contexts for the purposes of drawing contextually informed conclusions about the phenomenon (e.g. Dai et al., 2023; Tannebaum, 2016). While our inability to produce generalizable knowledge that can be applied to other populations is a serious limitation of our design, the affordance of a multi-case study is that it provides opportunities for reflective understandings of what worked and did not work within these particular instances, underscoring potential lessons for future researchers and educators who want to use a co-teach method in ways that honor and attend to the words and knowledge of the co-teachers themselves.

Participants

Research team's pedagogic interventions with co-teachers

It is important to note that the research team itself was actively involved in not only collecting and analyzing data but running professional development sessions and activity working with Dr. Luke and the AP-CSP teachers to support co-teaching and the CRSC curricular modifications. The first, second, third, and fourth authors helped to design the summer professional development program that two of the co-teachers attended, partially or in full, the summer before co-teaching began. This included introducing two of the three co-teachers to AP-CSP units, physical computing activities, and CRSC tools and resources. The professional development program is part of a much broader and longer effort by the third and fourth authors to develop PK-12 CS and engineering teachers' (who live in and around the large midwestern city) content knowledge in these areas by drawing on their own university resources and professional connections (Hill & Garraway, 2022). In addition, the second, fourth, and seventh authors worked with the co-teachers during the school year to develop and implement CRSC lessons for their

classrooms. The result was one lesson that was a modification to the code.org Unit 2: Digital Information, that focused on local issues around voting and political activism that were pertinent to students' own communities. There was also a CRSC modification to code.org Unit 4: Computer Systems and Network, which focused on the design and development of a decision making app for youth civic engagement.

Teachers & their contexts

Three participating teachers for the study were sought from local underserved high schools within and immediately adjacent to a large midwestern city. From this pool, CS teachers were selected that had a need and desire to develop their CS pedagogical content knowledge and had the support of their schools' administrative leadership. All three teachers selected had some experience teaching aspects of CS content but had never previously taught a full-year course, such as CS Principles, for AP credit; hence, we consider them to be novice CS teachers. Below are the teachers' backgrounds and teaching experiences as well as school demographics for where they taught.

Maya. Maya is a 52-year-old Black woman. She is a high school computer science teacher at a large urban school, where 87% of the students were eligible for free and reduced lunch. The school population included 99.1% Black students. Maya has been teaching for the past 28 years, including 10 years at the current school in a large urban district. Maya taught general computer classes that focused on students learning digital literacies, Office productivity and collaboration tools, and using applications to make yearbooks. Maya discussed how the yearbook project allowed her students to "make the connection between the tech side and creativity side." Beginning in 2021, Maya began to incorporate some material form the Advanced Placement Computer Science Principles (CSP) course in her own CS class.

Georgia. Georgia is a 55-year-old Black woman. She is a high school computer science teacher at a large urban school, where 72.7% of the students were eligible for free and reduced lunch. The school population included 99.8% Black students. She has been in the school district for the last 20 years, but only taught computer science for the past two years. Before beginning to teach the Computer Science Principles (CSP) course, Georgia taught English for 26 years. She also led students as a part of Games for Change for a couple of years before moving to teaching computer science. Games for Change is a national organization that uses game design and programming to engage middle and high students in social change within their community.

Matthew. Matthew is a 60-year-old White man with mixed ethnicity of Lebanese and Irish ancestors. Matthew was a teacher in a large suburban school where 87.5% of the students were eligible for free and reduced lunch. The school population included 50% White students, 34% Asian students, and 15% Black students. Matthew has been teaching for 25 years, first starting out as a Special Education teacher, and then taught cybersecurity, dual enrollment CS course with a local university for the last two years.

University co-instructor

Dr. Luke. Dr. Luke is a 38-year-old White man who was selected for the role of university co-instructor based on his PhD training combined with his unique experience as a high school teacher and his genuine desire to broaden participation of underserved youth in STEM fields. Dr. Luke's degrees are in fields of engineering where he took fundamental computer science courses. His PhD research involved the programming of computer models to simulate storm surge and inundation from hurricanes. Prior to becoming a university instructor, Dr. Luke participated in the Teach for America program, earned a provisional teaching certificate in integrated science, physics, and math,

and spent five years as a high school science teacher in two underserved urban high schools that are majority Black. Before the co-teaching experience discussed in this paper, Dr. Luke worked for three years at a university located in a large midwestern city. During this time, he collaborated as a co-instructor with local high school teachers, helping to enhance their ability to teach computer science and engineering-related courses. Dr. Luke participated in training on CRSC at the beginning of the project described in this paper, which when combined with his prior experience teaching diverse populations of high school students, are thought to be important assets.

Co-teaching model

For the three teachers who are focus of the case studies here, the co-teaching model relied on the premise that observation and participation in teaching by the instructional specialist (university co-instructor) would lead to knowledge growth by the high school teachers. Generally, either the high school teacher or university co-instructor led lessons completely, with gradual release of more days of instruction to the high school teacher over the course of the year. Because of scheduling constraints, one-on-one co-planning was infrequent and completed *via* email. Subsequent co-teaching has been modified due to reflections on these experiences, with co-instructed lessons featuring both partners as leaders occurring more frequently, and weekly meetings to cover content knowledge and pedagogy built into schedules.

Data collection and analysis

Teachers content knowledge

Multiple choice test. To measure teacher content knowledge, we used 20 multiple choice questions aligned to the content of the AP-CSP exam, with content covering both the programming and computational thinking portion, and the information technology portion of the exam. These questions were aligned to the AP-CSP exam in level of difficulty, content, and distribution. The assessment was created using questions from an AP-CSP exam preparation book published by a major test preparation company. Teachers were given this as both a pre-and post-test at the beginning and end of the school year. We examined changes in teachers' content knowledge by the number of correct answers on the multiple-choice test.

Teacher artifacts. Additionally, we assessed teachers' computer science content knowledge through their ability to create a computer program similar to the AP-CSP Create Task. After a year of co-teaching, teachers engaged in a week-long training session. During this training, they were assigned the task of writing a program using the same platform they had used to teach AP-CSP. The program was required to include five programming elements: a list, a function, a conditional, a loop, and parameters. An example from the AP-CSP course was provided for guidance. While teachers were offered as much support as needed, they were encouraged to work as independently as possible.

Teacher interviews

We analyzed six interviews, three pre-interviews with each co-teacher and three post-interviews with each co-teacher. The interviews were conducted *via* Zoom and afterwards they were transcribed verbatim for analysis. The first interview occurred after the co-teachers had participated in a professional development for high school computer science teachers to learn about the AP-CSP curriculum. One of the teachers, Maya, also participated in a week-long PD that focused on physical computing and how to bring culturally responsive approaches to teaching into their CS classrooms. On average the three pre-interviews lasted approximately 50 min. The first interview protocol had five sections. The protocol was designed to provide researchers with

insights into the teachers' history, ideas about CS pedagogy, and their beliefs about culture and inequity in CS. The five sections included: 1) demographic and teaching background (e.g. "How would you describe your gender identity?); 2) history of CS and AP-CSP teaching ("How did you start teaching CS?"); 3) reflections on CS pedagogical content knowledge ("When a student struggles with a CS concept, what's your approach to helping them?"); 4) reflections about the concept of culture and its relationship to CS and CS pedagogy? (e.g. "When you think about the idea of culture in relation to yourself what comes to mind?"); and 5) reflections about inequities in CS education (e.g. Why do you think inequities (gender, race, class, etc.) persist in CS?").

We conducted a second interview with the teachers after the school year had ended to better understand how they viewed their co-teaching experiences and to reflect on their work with the research team, including but not limited to implementing culturally responsive and sustaining lessons. On average the three interviews lasted approximately 51 min. The second interview protocol had four sections. The protocol was designed to reflect on co-teaching and the CRSC lessons, as well as revisit their ideas about CS pedagogy. These four sections included: 1) reflections on teaching AP-CSP and the associated curriculum (e.g. What do you think a student needs to do to succeed in an AP-CSP course?"); 2) reflections about the process and practice of co-teaching (e.g. How did you and [Dr. Luke] prepare for co-teaching AP-CSP?"); 3) reflections about implementing culturally responsive and sustaining lessons in AP-CSP (e.g. "One lesson was about Internet Dilemmas where we looked at community networks, internet activism, and the digital divide. Students were asked to examine local examples of internet activism and local efforts to address computer access. What worked well with this modification?"); and 4) revising reflections on CS pedagogical content knowledge ("When you launch a new CS lesson in your classroom, how do you help students understand the objectives of the lesson?").

Using nVivo qualitative coding software, two researchers (first and third author) collaboratively open-coded the interviews line-by-line to capture teachers' ideas and reflections. After we had developed the open codes, we used the axial coding process to identify overarching themes within which the codes could be organized. During this process we used a constant comparison method (Glaser & Strauss, 2017) to ensure that individual codes still fit within a particular theme and new themes were not emerging based on the analysis. The collaborative coding processes allowed us to collectively interpret and make meaning of the data through discussion and consensus-building; thus, leading to a richer and more nuanced analysis (Cornish et al., 2013). Four themes emerged from this process that are described below in each case study.

Results

Teacher content knowledge

The multiple-choice test results showed that teachers did not exhibit significant overall improvement in their content knowledge. Specifically, Maya had a pretest score of 10 and scored 10 on the posttest, Georgia's scored 9 on the pretest, which increased to 13 on the posttest, and Matthew scored 11 on the pretest and 12 on the posttest. The scores also show that teachers did not reach a high level of content mastery, which is defined as achieving 75% correct answers.

Our analysis of the teacher artifacts showed that teachers developed applications with varying levels of complexity and used differing amounts of support. Maya created an app about the history of her African American sorority, utilizing a list but not incorporating other required programming elements. She completed her work independently. Georgia developed an app comparing different Motown artists, successfully integrating a list, function, conditional, and loop, but not parameters. She required a high level of outside support. Matthew created an app comparing food across different cultures but only included a loop, without incorporating the other required elements (lists, functions, loops, and parameters). He worked independently as well.

Interview results

Maya

Journey into computer science. Maya's journey into teaching computer science started during college where she initially began a major in CS at a small university in the Midwest when "the population of African Americans [in CS] was low" and "[you are] trying to find your place and navigating that I was in, compet[ing] in a computer science major that was pretty much highly populated with white males." Maya stated that she found being a CS major "challenging because in computer science, of course, when you're working in COBOL, and Fortran and using all these programming languages, you work in groups, and I was I found myself self-isolated from the group activities, which impaired my ability to successfully and proficiently progress, you know, progress through the program … And I struggled for a year and a half like, what am I going to do? I got to get out. I can't survive like this. It's like I'm by myself."

Given her poor experience as a Black woman in CS, she switched her major, as she stated, "with some insight and support from an African American advisor, in the School of Education, I was encouraged and motivated to switch my major to business education with a minor in Computer Science." After finishing college, Maya joined the urban school district she is currently employed at about 30 years ago and started teaching computer applications that involved "how to use Microsoft Office programs, teaching them [students] how to use various communication and collaboration tools used in the workplace." However, about "seven or eight" years ago, Maya was pursuing her PhD and decided to engage her students in "more rigorous CS" as she called it and her research led her to pick and choose parts of the CS Principles curriculum from Code.org in her computer applications course. Then two years ago her high school decided to formally offer the CS Principles course and Maya was "asked to take on that challenge" and she "was like, let's do it."

Co-teaching experience and impact. Maya discussed her experience co-teaching the CS principles course with Dr. Luke as a mentoring experience, stating, "It's like a mentorship you know…it's like co-teaching, but he's teaching me." She further elaborated that this mentoring is providing her with CS content and pedagogical skills as highlighted by the following statement:

> Really modeling for me. What needs to be shared and provided [for students]. And, you know, yeah, just all-around examples of excellent teaching strategies, teaching AP Computer Science.

Maya discussed the importance of communication and collaboration with Dr. Luke, supporting her and students in the classroom. She said:

> Well, that we would, we would collaborate on the lessons, you know, weekly. We would chop up the lessons prior to a unit, he would engage us [the high school co-teachers] in a session where we could communicate and highlight any concerns that we might have about anything in the curriculum prior to us approaching that unit, you receive regular texts, and email communication. Awesome, even from the…wow, wow…even from the virtual teams environment, making certain that resources were uploaded and generated for students to be able to, to support, you know, support the learning.

The opportunity to co-teach with a more experienced CS teacher gave Maya an opportunity to reflect on her own practice before the experience as well as what she needed to do to continue to develop as a teacher. In particular, she stated that "it [the co-teaching] has changed [me] because I now know that I need to up my professional development game in computer science. It also shows me that a lot of things I was doing, I've been doing right prior to this experience."

It also gave her opportunities to learn the curriculum and provided her with new teaching strategies. This is reflected by the following comment, "I mean, I just, it's just been an all-around learning experience, just with learning the curriculum myself. Managing the classroom environment. With the multitude of learning abilities, in one classroom, create leaders out of students

to even help the job make the job even easier. Being intentional, with certain decisions and grouping. And just, I could just go on and on."

While Maya believed that having Dr. Luke in the classroom was valuable for her own learning, she also raised the need to devote additional work on her behalf to support students in the classroom. She stated:

> Because one thing that I discovered with having [Dr. Luke] to support me in this first year, is that I didn't do a good job. I didn't give him I didn't give the energy that I needed. And that's another story. We're gonna talk about that later. But it takes more time outside of the classroom, I discovered that to successfully help these kids be successful with the AP exam, outside learning opportunities need to be available to students. I'm tutoring.

Pedagogical approaches to CS teaching. During the interview with teachers, one of the themes that emerged related to teachers' use of pedagogical approaches to teach CS and support their students. Given that teachers themselves had limited knowledge of CS content, it is important to examine how they adapt teaching strategies that help them. Maya discussed how students know when you as a teacher do not know what you are teaching. She said, "I mean, the thing about it is your students can respect you more your kids know, when you don't know something… So instead of trying to, you know, shield that or be fearful of allowing kids to be leaders? It's a waste of time because we're developing leaders. Okay. Because if your kid could train you, they're going to be bad."

She further elaborated that she was herself learning alongside her students and was open with her students about her own lack of knowledge and experiences teaching CS. She also asked students for help when teaching, this is highlighted by the following statement, "I'm also still learning alongside my students, as well. And I'm not. I'm not ashamed to say that…My students need to know where my gaps are. Because I informed them. However, I informed them in a way where we need to explore [and] figure this out. Or how about help your teacher out." It is interesting to note that one of Maya's strategies to teach is using students themselves to help her figure things out when she encounters something she doesn't understand.

In order to help students learn, Maya used peer support rather than being the only source of support for the students. She stated, "I'm really kind of gauging them. You don't really want to provide all of the answers, but kind of walking them through with where the struggle of the challenge began. Okay, I'm one on one support. Pairing with, you know, someone else. Having that scaffolding, having those resources available for students to review and go over, you know, in isolation if they need to."

Culturally responsive-sustaining computing. When asked about the role of CRSC within her computer science class, Maya discussed how using culturally situated design tools (see Eglash et al., 2017) that are based on SNAP! to program cornrow braids would be appealing to students, stating, "when females hear that we're designing designs for cornrows and braids and look what I have in my hair…This is our culture. We love this". However, she also noted that within her CS class, using CRSC activity was a break from regular activities. She said, "I think what it did was it kind of took some of it [and] gave them a little bit of downtime. And then it gave them Okay, so we can go back and work on their projects."

Georgia

Journey into computer science. Georgia has been teaching for the past 26 years, predominantly as an English teacher, before she started to engage her students in game design using Scratch through the Games for Change competition. As a part of that effort, she learned Scratch, which she believed was special, stating "I can't remember the name of the group [Games for Change] that

got me kind of started in learning how to use Scratch and everything. I realized that it was you know, I liked the fact that everybody couldn't do it." Her formal foray into teaching stand-alone computer science courses started when a fellow teacher with a CS background pushed her into teaching computer science as she would go to him to ask questions related to game design. She said, "I always tell everybody that my coworker bullied me into becoming a computer science teacher."

Both Georgia and her fellow teacher participated in a professional development program through Code.org that she stated that she enjoyed as it was different from "regular math," which she is afraid of. This is reflected in the following comment. "And then we did the Code.org training together. I've done it twice. So, the first time we did that. And so, I learned a lot of different things and I really enjoyed it. And it was fun. It's numbers, but it's different from just regular math. Math just freaks me out." Two years ago, Georgia used some AP Computer Science Principles materials into her CS course.

Co-teaching experience and impact. When discussing the role of co-teachers and her experience with Dr. Luke, Georgia mentioned it as a modeling experience stating, "it's great with me because I pick up a little bit from what he's doing with the kids." At the same time, she stated that she was putting more effort into her own learning and paying attention to the lessons because "I don't want him to think I'm stupid. Yeah, but you know, I'm just putting a little bit more effort into knowing what I'm doing when I'm presenting it." And when she is uncertain about teaching something, Dr. Luke would step in and teach it. She said, "if it's something that I wasn't sure of, and so he would go on and take from that point, and then kind of bounce it back to me".

However, she also found the co-teaching experience intimidating since she was a novice CS teacher. She said,

> I was a little intimidated, to be honest, because he's so well versed in it, whereas I'm not. And then when I would take over the class, like, when he would have me teach certain things. I was okay with that, you know, I'd have a problem with it. But it's like, always at the end, I'm like, Did I do okay….I get an opportunity to also, you know, as I'm going through it, but it can be a little intimidating sometimes.

Georgia discussed how co-teaching was a better experience for students than virtual volunteers who previously joined her class to support students. In particular, she noted how students did not have a relationship with virtual volunteers, stating, "With them [volunteers] being virtually there's no personal relationship…they [students] didn't want to ask the volunteers for help. They didn't want to talk to them. So, it made me look bad as a teacher, because I'm trying to push them to ask for help." Georgia further elaborated on Dr. Luke as a co-teacher was a positive experience for the students, stating "I love the fact that Dr. [Luke] is here and he's going through it. He's just like a ball of energy and I don't have that type of energy. I'm a little older, but he's great with the kids." She also stated that her own relationship with Dr. Luke was positive and that led to them working well together. She said, "I think because we have a pretty good relationship, that it was kind of easy for us to work together."

Pedagogical approaches to CS teaching. One of the pedagogical approaches Georgia used to teaching CS was like Maya's where she used students to help teach CS content given her own lack of CS background. For example, she discussed that she is trying to learn herself and then using questioning to understand what students need, stating,

> I try to kind of figure it out first for myself, and then try to find a way that would help them to understand exactly what we're doing. And I know that sometimes if they don't, that I have to go back, I have to step back and ask them, okay, so what is it that you don't understand? How can I help you to understand it better?

She further elaborated on this idea on how students helped her when she struggled to teach. She said,

> If it was something I didn't know, we would go through it together. And then sometimes when they would have problems, we would go through it line by line as a class to say, you know, where the problem might have been to debug, I guess you would say call it to try and figure it out…if I didn't get it, and they helped me, and I think they can see like when they are able to help their teacher.

Georgia also had students sit next to her to help teach. She stated, "Why would you not ask them when they're right here, now, and they know what they're doing more so than I do."

As mentioned previously, George used questions as an approach to help students work through problems. She stated, "Go through it with them, you know, ask questions. Ask them to ask questions, if it's something that they don't understand. And then kind of just go through it that ways, they'll step by step, and making sure that they understand exactly what it is that they're supposed to do."

Georgia used peer tutoring to support students who did not have a strong CS foundation. It is important to note that this could be a strategy given her own lack of CS content knowledge:

> Well, some of them pick up really fast and others don't. I like when peers help each other because sometimes it is easier for them to catch on from their peers than sometimes if I'm teaching it, because they speak the same language, even though we all speak English, but I speak grownup English and they speak kid English. So, they're more apt to pick up on someone that's closer to their age and go from there.

Georgia also discussed how challenging it is for her to teach given her lack of CS knowledge and the role CSP curriculum plays in teaching CS. She said:

> The challenge for me, because I don't have a real background is me, really. Because I'll do what I have to do to get started. And a lot of times for me, because I do have the answers, sometimes I'm that person that kind of works backwards. So see how they see what the answer is, and then try and figure out how I got to it. So that kind of works for me, but because I don't have a background in it. I like it. I'm learning with them. But this also can become challenging at times….I liked that they have the lesson plans there. And I would go through it by the lesson plan. And I would prepare through that lesson plan. And so it helped me to when that's something that I didn't understand, to either ask Andy for help or I kind of figured it out myself.

This suggests that a canned curriculum can play a role in helping teachers deliver CS lessons despite her lack of background CS knowledge, but Georgia used several other strategies to teach the content.

Culturally responsive-sustaining computing. When asked to reflect on the CRSC lessons that were implemented in the classroom, Georgia discussed how it provided an opportunity for students to explore relevant issues in their own community. She said:

> I think it opens their eyes to see the issues that are out there that they may know and may not know. And so it kind of educates them in those areas, and allows them to kind of try and figure out what they could do to make a difference.

Georgia had not been introduced to culturally responsive approaches before she joined this project. It is important to note that Georgia couldn't fully participate in the PD as she got COVID-19 and had to miss days when we introduced CRSC to the teachers. Hence, her views about CRSC were limited when compared to the other two teachers.

Matthew

Journey into computer science

Matthew reflected on the roots of his CS teaching beginnings from his days as a Special Education teacher when he "dabbled with computers to modify the learning that I needed to be individualized with my students." As Matthew worked to integrate technologies into his classrooms, "people started to notice and so they drafted me into other roles that utilized technology and

technology training." This led him to teach "different things related to computers," running "a camp for kids to write their own programs," and teaching vocational education for a few years. The summer camp offered all students, including students with disabilities, the opportunity to use HyperCard and hype-script to design a game.

Matthew transitioned back to teaching special education when he joined his current school about 15 years ago, which also gave him an opportunity to start an introduction for a computer applications class. The class was to help the diverse immigrant student population at the school understand "fundamentals of computer applications" as they did not have opportunities to use computers in their home countries. After a couple of years of teaching the class, Matthew asked the school administration about teaching a CS course and was given permission.

Matthew acknowledged that he did not have the necessary background knowledge to teach CS but enjoys programming and believes that he is a good teacher. This is reflected in the following comment, "So I like computers. I like programming. I don't I wouldn't say I'm very good at it. But I like teaching, people have told me that I do pretty well with teaching. So having that combination I've enjoyed in my years of teaching, and particularly here at Highland High School, the computer science classes."

Co-teaching experience and impact

When asked to discuss his experiences with co-teaching alongside Dr. Luke, Matthew discussed how it allowed him to develop his knowledge to teach CS as well as observe good instructional practices. He stated:

> Co-teaching with [Dr Luke] is primarily modeling and going through the content and doing various activities to teach. So, I think I'm getting equally prepared through the mentoring. I'm more excited about this and definitely excited about the fact that this grant went through with [University] and [Dr Luke] because while I gained a familiarity with the content, and some of the techniques of delivering and promoting the teamwork and group activities, and the pair sharing and whatnot. I'm seeing it being modeled, and I'm getting a chance to practice it more.

When comparing his experience co-teaching with Dr. Luke, as compared to volunteers that had previously come to support his CS teaching, Matthew reported how having a co-teacher with teaching experiences who comes every day is far more valuable than volunteers who are not certified to teach come in for 1-2 days a week. This is highlighted by his following comment:

> And he's [Dr. Luke] a certified teacher, and my volunteers are not certified to teach. And so the way that they deliver the information is, has been decidedly different. So I've learned things a little deeper, a little bit better with [Dr. Luke], and I'm hopeful in carrying on the baton, so to speak, that I'll be better yet on my own with more practice.

Matthew also elaborated on how he has incorporated pedagogical strategies implemented by Dr. Luke into his own practice. For example, Matthew described how he now provides opportunities for students to solve problems on their own rather than giving them the answer, stating:

> My approach is to ask them, but usually, usually it'll be let's see, show me show me what's going on. Because oftentimes, they'll say, how do I do this? And it'd be so easy for me to just say, well, he just, you know, assign the variable, set this up and pull these in and you got it, versus what are you trying to do here? So, I try to ask the question. And this, of course, is something decidedly different than what I would have done with my other classes in the past.

In addition to having Dr. Luke as a model and resource in the classroom for his own learning, Matthew felt that the co-teacher helped improve student learning in his CS classroom, stating:

> I enjoyed seeing the kids' projects, and to see the ideas of all of them, even the kids that couldn't complete the program that they were attempting, just their ideas, creativity, were clearly brought out by a lot of the activities and efforts of [Dr. Luke].

Matthew also believed that the learning in the classroom also translated to the AP exam:

> Many of the kids were quite pleased after taking the AP exam in computer science versus the other AP exams that they had. So part of it, I think, they thought, Oh, this is just it was really easy. Yeah. But the reality was, I think that they had some really good preparation, especially with [Dr. Luke].

However, it is important to note that Matthew initially found having a co-teacher to be intimidating, which may be due to lack of sufficient CS content knowledge. He said, "there's a nervousness that I had early on. Not just with [Dr. Luke] but with the concepts with the AP, because I wanted to be sure of so much [content] and I had so little time."

In terms of Matthew's development as a teacher due to the co-teaching experience, he discussed opportunities to teach on his own and getting feedback from Dr. Luke. Describing this, Matthew states, "he [Dr. Luke] will let me know when he'd like me to teach something with enough lead time that I might prepare for. And, and he's been great to give me feedback, right after almost in all cases. And I tried to take it in." Matthew also wished that there would be more opportunities for him to be active in leading lessons even though much of the content was new for him. He stated:

> So, content regarding, you know, the vocabulary, the algorithms, the loops, and that sort of thing. You know, the fundamentals of everything that we had to accomplish for the task, the end tasks [performance tasks and end of unit creative projects] that we had. I had some experience with some of it. Yet, I feel like I didn't really practice as much in preparing some of them because he did the lesson instead of me. So then the next lesson, he did it, so then I didn't have to prepare for that one. And then the next lesson, I did it, and it's like, did he cover that today? So I think if this were a situation that was planned, with the Parallel Teaching, so to speak, where I'm set up with a co teacher again. I don't want to say I expect Matthew to do more. But know that somehow he needs to maintain some active and practicing of the content before each lesson, I ideally, right.

Pedagogical approaches to CS teaching

Mathew used pedagogical approaches that overlapped with other two teachers to support his teaching, such as questioning students. As discussed in previous section questions students was a strategy that Matthew picked up because of co-teaching experience with Dr. Luke. Matthew further elaborated on it and how he has incorporated it into his teaching. He said:

> And this [questioning], of course, is something decidedly different than what I would have done with my other classes in the past. I've, I've learned to try not to volunteer as quick a result or even to say, yeah, that's perfect. If they're asking me for help, they're sensing that something's not right. What problem are they trying to figure out with this? It's not working. And then half the time I'll say so.

Matthew also used peer support and group work to help students learn. Specifically, he discussed:

> I put them in random groups or deliberate groups oftentimes is a deliberate pairing and deliberate group. So that if one has a stronger language than the other, then that other might benefit from translation, might benefit from having not just instruction, but a one to one partner, to carry them through situations of switching roles and taking responsibility. It's a growing task for them in claiming more ownership to this opportunity and recognizing that they can, they can get as much out of this as they would like.

Matthew also discussed inviting role models into his class to allow students to see themselves in computing roles and motivate them to learn CS. He stated:

> I strive for at least once a month to have somebody come in. And oftentimes, it'll be a local entrepreneur or a programmer themselves, or somebody who has had experience in programming. And, and usefully connect them as representing the population of African American person that fits all those other ingredients or a female. So at different times, whomever he or she might be, they're connected to having something to do with programming. And there they are. So the kids not only get to see, like the poster, but they get to ask questions and they get to so I'll dedicate a whole hour or period with that.

Culturally responsive-sustaining computing

Matthew implemented a culturally responsive computing lesson where students created apps to collect data about their communities and use this data to then suggest a decision for the community to make. It required students to use conditionals and variables and have output which depended upon input. When asked to reflect on culturally responsive computing and teaching it in the AP-CSP classroom, Matthew discussed how it provided students with an opportunity to think critically about the data that students wanted to collect while also learning about their community. This is reflected in the following comment:

> …certainly giving them some facts to think about, and giving them the opportunity to dialogue, and to see what information they wanted to collect. And then it was a matter of how they wanted to collect it. And then what they might do with it……. really noting that, oh, not everybody has a cell phone. And not everybody has a computer that has high speed, let alone you know, so the concepts of who hasn't was a great opportunity to talk. We talk about yes, different countries, huge disparities, but even in the United States, we still have some, some disparities as well.

While Matthew believed that CRSC engaged students to use computing to examine larger issues, he saw that there was little time to bring CRSC into the AP classroom as there was not enough time for students to complete the project. He stated, "So we had half the amount of time and then it was pretty much the last review day before the AP… half [of the students] didn't get a chance to have the working app poll their peers. They didn't get the satisfaction, they didn't get the closure of not just oh, my app works."

Discussion

The case studies of Maya, Georgia, and Matthew provide insightful narratives into their journeys into CS teaching, their experiences with co-teaching, pedagogical approaches, and the integration of CRSC within their classrooms. Each teacher, coming from diverse backgrounds and teaching experiences, shares their unique challenges and growth in teaching CS, highlighting the importance of mentorship, collaboration, and culturally responsive-sustaining computing in enhancing their teaching practices and student learning. Across the three case studies, some key findings emerged that are relevant to answering our two research questions above.

Finding #1: Co-teaching is a valuable professional learning experience for teachers

One of the main findings from our study is that all three teachers found co-teaching to be a valuable professional development strategy. All three teachers discuss how they benefited from the mentorship and collaboration offered by co-teaching with Dr. Luke. In particular, the co-teaching allowed them to develop pedagogical strategies they could adapt in their own teaching; however, it is unclear the impact it had on their CS content knowledge. All three teachers discussed how Dr. Luke served as a model that is preparing them to teach on their own. It is important to note that both Georgia and Matthew had volunteers in their classroom previously but found co-teaching on a daily basis much more valuable for their own development as well as being a positive experience for students. Georgia and Matthew also noted that having Dr. Luke was initially intimidating given their lack of CS knowledge while Maya brought up the need for her to devote more time to develop her own knowledge during co-teaching experience. There is limited research on how to support teachers with limited content knowledge through a co-teaching model and this research has primarily only focused on co-teaching between general education and special education teachers for supporting students with disabilities. This prior research has found that co-teaching models require both teachers to be collaborative as one teacher can observe other teachers' teaching approaches and then adapt those strategies to engage all students (Shaffer & Thomas-Brown, 2015). Our findings show that all three teachers saw Dr.

Luke as a model and then adapted aspects of his pedagogical strategies into their own instruction. Furthermore, our co-teaching approach served as a job-embedded professional development as teachers learned CS content-specific teaching skills. Job-embedded professional development has been found to improve teacher outcomes, such as their content knowledge and teaching efficacy beliefs as well as student achievement (Althauser, 2015; Wei et al., 2009). It should be noted that while teachers adapted the pedagogical strategies, their content knowledge did not significantly improve.

Finding #2: Teachers use pedagogical approaches to fill gaps in CS content knowledge

Another key finding from our study is that despite varying levels of initial CS expertise, each teacher navigated teaching AP-CSP during the year by employing strategies such as peer support, student-supported teaching, questioning to facilitate learning in their classrooms, and relying on the curriculum. It is important to note that both Maya and Georgia noted how they used students to help teach when they themselves did not have the necessary CS knowledge. Maya seemed comfortable letting her students know when she didn't know something and seeking their help. On the other hand, Georgia used her students' help when she was teaching, but it was evident that she didn't want to come across as a bad teacher. One of the strategies Georgia used was questioning to help students work through problems, which was like how Matthew shifted to using questions as a way for students to problem solve rather than just giving them answers. Both Maya and Matthew also discussed how they utilized peer support and group work where students with stronger CS knowledge could help students who were still developing their CS skills. It is important to point out that this study did not examine whether and how teachers' limited CS background played a role in use of peer-to-peer teaching. Future research should investigate the role of teacher knowledge in adapting specific instructional practices. It is, however, important to note that these CS teachers had taught other subjects before moving to CS and were the only CS teacher in their school. As such they could have adopted practices from their previous teaching experiences in other content areas. Furthermore, as mentioned by Matthew, teachers could have also adopted instructional strategies, such as questioning students from observing Dr. Luke. While there is no prior studies on how teachers' content knowledge influences their use of instructional strategies, previous research has suggested that teachers' prior CS experiences as well as previous subjects taught do not impact their efficacy beliefs about managing a CS classroom, using effective instructional strategies, and engaging students in CS classrooms (Yadav et al., 2021). This along with our findings potentially suggest that novice CS teachers leverage certain pedagogical approaches from their own teaching repertoire to deliver CS content to their students. While this approach could be useful in the short term, our results point out a need to develop teachers' capacity in CS content as well as CS pedagogical content knowledge to allow them to feel confident and provide high quality CS instruction to their students.

Finding # 3: Teachers' limited adoption of culturally responsive-sustaining approaches

Finally, we found that teachers' attempts to incorporate CRSC into the curriculum were limited even as they saw the importance of the CRSC to make CS education relevant and accessible to a diverse student population. This is not surprising given teachers' limited knowledge of CS content and limited CS pedagogical skill, which may complicate their abilities to integrate CRSC into their classroom instruction (see Goode & Ryoo, 2019 for a discussion of the relationship between content knowledge and knowledge of students' culture and community). There is limited evidence in how teachers' own content knowledge and pedagogical skills influences their implementation of CRSC approaches in their instruction, but Chu and Garcia (2014) found that in-service teachers' perceived quality of professional preparation is a significant predictor of their culturally responsive teaching efficacy beliefs. Furthermore, teachers' implementation of CRSC may be tied to their self-efficacy

beliefs in their ability to bring CRSC approaches and build a sense of trust with their students (e.g. Siwatu, 2007). And if teachers feel they can't fully support their students and gain their trust, it could lead them to be less confident in their teaching as well as integrate CRSC into their CS instruction. Relatedly, given that teachers were using AP CS Principles curriculum, they may not feel they have the agency to change and adopt CRSC approaches into the given curriculum. One aspect of teacher agency is "when teachers attempt to influence curriculum change in their school, department and/or classroom in an effort to achieve a desired outcome." (Jenkins, 2020, p. 168). A systematic review of 104 empirical studies on teacher agency found that teachers' knowledge, skills, and prior experiences play a significant role in their capacity to exercise their agency (Cong-Lem, 2021). In the context of our study, the lack of CS knowledge and prior experience could have led teachers to feel they did not have the agency to implement changes to the curriculum and bring CRSC approaches into the classroom.

Conclusion: Implications for practice and future research

Our findings have some implications for practice to support novice teachers. Specifically, it is important to develop co-teaching models that facilitate pedagogical as well as an explicit focus on enhancing teachers' CS content knowledge. Pairing novice teachers with experienced CS educators can provide real-time feedback and modeling of pedagogical strategies. However, teachers may need additional support to develop their CS content knowledge. One approach may be for teachers to take courses that specifically address their need to learn CS to teach rather than taking courses designed for CS majors.

In addition, teacher PD should include components that prepare teachers to effectively incorporate CRSC practices in their CS classrooms. This could involve strategies for making CS more relevant and accessible to diverse student populations, as well as developing teachers' self-efficacy in implementing these approaches. However, to achieve this, curriculum and accompanying PD should empower teachers to exercise their agency in adapting and enriching the curriculum with CRSC approaches. This should be an iterative process where teachers are provided support for developing CRSC approaches before implementing them and revising them with feedback from peers and/or a co-teacher.

Based on these findings, future studies should explore how co-teaching influences novice CS teachers' content knowledge, not just their pedagogical strategies. Indeed, we should also note that while these teachers were novices when it came to CS, they were all veteran teachers with decades of pedagogic experience. It is, thus, interesting that our co-teaching intervention seemed to shape the teachers' ideas about their pedagogic strategies most, and not so much their CS content knowledge. It is unclear if this would have been different for new teachers without CS backgrounds or new teachers coming with CS backgrounds. Thus, these findings and our model might be most relevant for future teams who are working with teachers who have a long history of pedagogy but are new to CS. Whether this model would be relevant to new teachers with or without CS backgrounds is unclear but given that the shifts in perceptions were around pedagogy it could, potentially, be useful to any new teacher. Understanding the dynamics of content knowledge acquisition in co-teaching settings could inform the development of more effective professional development models. Future research should also focus on the interplay between teachers' CS content knowledge and their instructional practices by examining how teachers' backgrounds in CS influences their adoption and adaptation of instructional strategies. Furthermore, research is also needed to understand the barriers to implementing CRSC approaches in CS education, particularly in relation to teachers' content knowledge and pedagogical skills.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is supported by the National Science Foundation under grant numbers 2122349 and 2122314. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Directorate for Computer and Information Science and Engineering.

Notes on contributors

Aman Yadav is the Lappan-Phillips Professor of Computing Education in the College of Education and College of Natural Science at Michigan State University with extensive experience in research, evaluation, and teacher professional development. His research and teaching focus on supporting educators to understand, apply, and critically evaluate the use of computing in K-12 classrooms. His co-edited book, Computational Thinking in Education: A Pedagogical Perspective tackles how to integrate computational thinking, coding, and subject matter in relevant and meaningful ways.

Michael Lachney is an Associate Professor in the Educational Psychology and Educational Technology program at Michigan State University. He studies the roles that community and cultural knowledge can play in designing racially equitable computer science education. His research has appeared in the journals, Learning, Media and Technology, Science as Culture, Computer Science Education, among others.

Richard Hill, PhD, is the Associate Dean for Research and External Initiatives in the College of Engineering & Science, and a Professor of Mechanical Engineering, at the University of Detroit Mercy. His research interests include formal synthesis of control algorithms, as well as engineering education and techniques for broadening participation in engineering and computer science. He is the founder of the iDRAW high school STEM pathways program.

Andrew Lapetina is an instructor for the iDRAW program in the College of Engineering and Science at the University of Detroit-Mercy. Prior to this he was a traditional high school teacher in Detroit, and earned his Ph.D. in civil engineering at the University of Florida.

Anne Drew Hu is a PhD candidate in Educational Psychology & Educational Technology at Michigan State University. Their research focuses on integrating critical perspectives into computing education. For more information, visit https://annedrewhu.com/

Hyein Jee is a doctoral student in the Educational Psychology and Educational Technology program at Michigan State University. Her research focuses on supporting high school computer science teachers' integration of cultural contents into their instructional practices.

Madison C. Allen Kuyenga is a PhD candidate in the Educational Psychology and Educational Technology Program at Michigan State University. Madison's research employs design-based, mixed, and qualitative methods to advance culturally responsive and sustaining computer science education, informing approaches to equity within the field. Madison collaborates with educators and community experts to co-create learning experiences that extend beyond the classroom, shaping inclusive and culturally informed educational practices.

References

Althauser, K. (2015). Job-embedded professional development: Its impact on teacher self-efficacy and student performance. *Teacher Development*, *19*(2), 210–225. https://doi.org/10.1080/13664530.2015.1011346

Astrachan, O., Cuny, J., Stephenson, C., Wilson, C. (2011). *The CS10K project: Mobilizing the community to transform high school computing* [Paper presentation]. Proceedings of the 42nd ACM technical symposium on computer science education (pp. 85–86).

Balta, N., Amendum, S. J., & Fukkink, R. G. (2023). The effect of job-embedded professional development on teacher and student outcomes: A multi-level meta-analysis. *International Educational Review*, *1*(1), 1–23. https://doi.org/10.58693/ier.111

Chu, S. Y., & Garcia, S. (2014). Culturally responsive teaching efficacy beliefs of in-service special education teachers. *Remedial and Special Education*, *35*(4), 218–232. https://doi.org/10.1177/0741932513520511

Code.org, CSTA, ECEP Alliance (2023). *2023 state of computer science education*. https://advocacy.code.org/stateofcs

Cong-Lem, N. (2021). Teacher agency: A systematic review of international literature. *Issues in Educational Research*, *31*(3), 718–738.

Cornish, F., Gillespie, A., & Zittoun, T. (2013). Collaborative analysis of qualitative data. *The SAGE Handbook of Qualitative Data Analysis*, *30*(79), 93.

Croft, A., Coggshall, J. G., Dolan, M., Powers, E., & Killion, J. (2010). *Job-embedded professional development: What it is, who is responsible, and how to get it done well*. National Comprehensive Center for Teacher Quality.

Dai, C. P., Ke, F., Pan, Y., & Liu, Y. (2023). Exploring students' learning support use in digital game-based math learning: A mixed-methods approach using machine learning and multi-cases study. *Computers & Education*, *194*, 104698. https://doi.org/10.1016/j.compedu.2022.104698

Davis, J., Lachney, M., Zatz, Z., Babbitt, W., & Eglash, R. (2019). *A cultural computing curriculum* [Paper presentation]. Proceedings of the 50th ACM technical symposium on computer science education (pp. 1171–1175). https://doi.org/10.1145/3287324.3287439

Derrington, M. L., & Kirk, J. (2016). Job-embedded professional development policy in michigan: can it be successful? *Professional Development in Education*, *42*(2), 201–217.

Dubek, M., & Doyle-Jones, C. (2021). Faculty co-teaching with their teacher candidates in the field: Co-planning, co-instructing, and co-reflecting for STEM education teacher preparation. *The Teacher Educator*, *56*(4), 445–465. https://doi.org/10.1080/08878730.2021.1930310

Eglash, R., Babbitt, W., Bennett, A., Bennett, K., Callahan, B., Davis, J., … Tully, K. (2017). Culturally situated design tools: Generative justice as a foundation for STEM diversity. In *Moving students of color from consumers to producers of technology* (pp. 132–151) IGI Global.

Eglash, R., Gilbert, J. E., & Foster, E. (2013). Toward culturally responsive computing education. *Communications of the ACM*, *56*(7), 33–36. https://doi.org/10.1145/2483852.2483864

Ericson, B. (2022). *AP CS Data for 2022*. https://cs4all.home.blog/2023/04/10/ap-cs-data-for-2022/

Ernst, J. V., Clark, A. C., & Bowers, S. W. (2016). Flexible and job-embedded professional development for in-service technology, design, and engineering educators. *Journal of Technology Studies*, *42*(2), 65–74. https://doi.org/10.21061/jots.v42i2.a.1

Glaser, B., & Strauss, A. (2017). *Discovery of grounded theory: Strategies for qualitative research*. Routledge.

Goins, R., Koshy, S., Scott, A., Payton, F., Cobb, Lundgren, K., Toldson, I. (2022). *The state of tech diversity: The black tech ecosystem*. https://www.kaporcenter.org/wp-content/uploads/2023/03/KC22001_black-tech-report.final_.pdf

Goode, J., Johnson, S. R., & Sundstrom, K. (2020). Disrupting colorblind teacher education in computer science. *Professional Development in Education*, *46*(2), 354–367. https://doi.org/10.1080/19415257.2018.1550102

Goode, J., & Ryoo, J. J. (2019). Teacher knowledge for inclusive computing learning. In *The Cambridge handbook of computing education research* (pp. 709–726). Cambridge University Press.

Hill, R., & Garraway, J. B. (2022, August). Development of a high school engineering pathways program (work in progress, diversity). In *2022 ASEE Annual Conference & Exposition*.

Jenkins, G. (2020). Teacher agency: The effects of active and passive responses to curriculum change. *The Australian Educational Researcher*, *47*(1), 167–181. https://doi.org/10.1007/s13384-019-00334-2

Kapor Center (2021). *Culturally responsive-sustaining computer science education: A framework*. https://www.kaporcenter.org/wp-content/uploads/2021/07/KC21004_ECS-Framework-Report_final.pdf

Lachney, M., Bennett, A. G., Eglash, R., Yadav, A., & Moudgalya, S. (2021). Teaching in an open village: A case study on culturally responsive computing in compulsory education. *Computer Science Education*, *31*(4), 462–488. https://doi.org/10.1080/08993408.2021.1874228

Lang, K., Galanos, R., Goode, J., Seehorn, D., & Trees, F. (2013). *Bugs in the system: Computer science teacher certification in the U.S. computer science teachers association*. ACM.

Margolis, J., Estrella, R., Goode, J., Holme, J. J., & Nao, K. (2017). Stuck in the shallow end. In *Updated edition: Education, race, and computing*. MIT press.

Rankin, Y. A., Thomas, J. O., & Erete, S. (2021). Black women speak: Examining power, privilege, and identity in CS education. *ACM Transactions on Computing Education*, *21*(4), 1–31. https://doi.org/10.1145/3451344

Semon, S., Lane, D., Jones, P., & Smith, S. M. (2020). Job-embedded professional development: Implementing co-teaching practices in general education classrooms. *International Journal of Inclusive Education*, 1–16. https://doi.org/10.1080/13603116.2020.1821448

Shaffer, L., & Thomas-Brown, K. (2015). Enhancing Teacher competency through co-teaching and embedded professional development. *Journal of Education and Training Studies*, *3*(3), 117–125. https://doi.org/10.11114/jets.v3i3.685

Siwatu, K. O. (2007). Preservice teachers' culturally responsive teaching self-efficacy and outcome expectancy beliefs. *Teaching and Teacher Education*, *23*(7), 1086–1101. https://doi.org/10.1016/j.tate.2006.07.011

Skoretz, Y. M. (2011). *A study of the impact of a school-based, job-embedded professional development program on elementary and middle school teacher efficacy for technology integration* [Theses]. Dissertations and Capstones. 150. https://mds.marshall.edu/etd/150

Tannebaum, R. P. (2016). Cooperating teachers' impact on preservice social studies teachers' autonomous practices: A multi-case study. *The Journal of Social Studies Research*, *40*(2), 97–107. https://doi.org/10.1016/j.jssr.2015.07.005

Wei, R. C., Darling-Hammond, L., Andree, A., Richardson, N., & Orphanos, S. (2009). *Professional learning in the learning profession: A status report on teacher development in the United States and abroad*. National Staff Development Council.

Yadav, A., Connolly, C., Berges, M., Chytas, C., Franklin, C., Hijón-Neira, R., … Warner, J. R. (2022). *A review of international models of computer science teacher education* [Paper presentation]. Proceedings of the 2022 working

group reports on innovation and technology in computer science education (pp. 65–93). https://doi.org/10.1145/3571785.3574123

Yadav, A., Lishinski, A., & Sands, P. (2021). *Self-efficacy profiles for computer science teachers* [Paper presentation]. Proceedings of the 52nd ACM technical symposium on computer science education (pp. 302–308). https://doi.org/10.1145/3408877.3432441

Yadav, A., Gretter, S., Hambrusch, S., & Sands, P. (2016). Expanding computer science education in schools: Understanding teacher experiences and challenges. *Computer Science Education*, *26*(4), 235–254. https://doi.org/10.1080/08993408.2016.1257418

Yin, R. K. (2014). *Case study research: Design and methods* (5th ed.). Sage Publications.