Jl. of Technology and Teacher Education (2022) **30**(4), 1–31

Supporting Early Elementary Teachers' Coding Knowledge and Self-Efficacy Through Virtual Professional Development

MADHU KAPOOR
WestEd, USA
madhu.govind@gmail.com

ZHANXIA YANG
Boston College, USA
zhanxia.yang@bc.edu

MARINA BERS
Boston College, USA
marina.bers@bc.edu

Prior work has shown a lack of quality professional development (PD) programs specifically targeted for early elementary teachers to improve their knowledge and self-efficacy around teaching coding in their classrooms. Whereas traditional PD programs in this area have relied upon in-person workshops, the COVID-19 pandemic necessitated the need to explore virtual PD models. This article reports a study in which $N = 44$ early elementary teachers participated in a virtual PD training. The virtual PD focused on guided play and exploration of the ScratchJr programming app and introduced teachers to the Coding as Another Language (CAL)-ScratchJr pedagogical approach and curriculum. Before and after the virtual PD, teachers completed a survey and were assessed on their level of ScratchJr knowledge. Results showed significant improvements in teachers' coding knowledge and self-efficacy. Teachers without prior coding experience exhibited greater growth in their reported self-efficacy. Implications and future work for designing effective virtual PD opportunities for early elementary teachers are discussed.

Learning to code has become an increasingly popular and highly demanded skill in the twenty-first century. Although most U.S. states have adopted policies supporting computer science (CS) education, one of the biggest impediments to progress is the lack of quality professional development (PD) for classroom teachers (Code.org, 2022). In 2016, the U.S. saw the graduation of 10,000 math teachers and 12,000 science teachers but only 75 CS teachers from pre-service preparation programs. Unlike these other well-established subject areas, there are limited in-service PD opportunities for CS, particularly in the lower grades where CS is one of many subjects that classroom teachers are responsible for covering in their curriculum (Rich & Hodges, 2017, Stanton, 2017; Yadav, Gretter et al., 2016).

In previous years, a major obstacle to promoting coding education in the lower grades was the lack of developmentally appropriate technologies and pedagogical approaches. That is no longer the case. Today, there are over 30 computational kits with different design affordances and interfaces targeted for children eight and under (Yu & Roque, 2018). ScratchJr, the graphical interface used in this study, is the most popular free programming language for young children worldwide with users in almost every country (Bers, 2018; Bers & Resnick, 2015; Bers & Sullivan, 2019; Leidl et al., 2017; Unahalekhaka & Bers, 2021). With respect to pedagogical approaches, various companies and nonprofit organizations have developed coding curricula and family enrichment activities to introduce coding to young children in a variety of educational settings (Bers et al., 2022; Century et al., 2020; Govind et al., 2020; Relkin et al., 2020; Rich et al., 2021). The challenge stakeholders now face is how to equip early elementary teachers with the knowledge and self-efficacy needed to integrate coding effectively in their classrooms. This involves providing technical skills, pedagogical frameworks, and developmentally appropriate practice.

New PD models have aimed to address this challenge, such as single-day and weeklong workshops and university-based courses spanning one or multiple semesters. However, the onset of the COVID-19 pandemic drastically reduced opportunities for in-person workshops, forcing PD providers to shift and adapt to digital platforms. The assumption was that teachers would benefit from attending these virtual workshops and gain content and pedagogical knowledge around teaching coding. However, little empirical research has been published on whether these kinds of virtual opportunities have fulfilled their intended goal of promoting teachers' coding knowledge and self-efficacy. The study addressed in this article aims to fill this gap and contribute to the existing literature on professional development around early computer science education.

This article focuses on the impact of a virtual PD on a cohort of K-2 teachers in a public school district in the northeastern part of the U.S. As part of a broader research-practice collaboration between the district and a university-based research lab, the teachers participated in a 4-hour PD conducted virtually over Zoom, split into two 2-hour sessions. The first part of the PD engaged participants in guided play and hands-on exploration of the ScratchJr programming app. The second part of the PD introduced participants to the Coding as Another Language (CAL) pedagogical approach, which understands the learning of computer science as a new literacy that supports young children in developing new ways of thinking about themselves and the world (Bers, 2019). Teachers then explored the CAL-ScratchJr curriculum that they would later implement in their classrooms as part of the broader study. Before and after the PD, teachers completed a survey and were assessed on their level of ScratchJr knowledge using the validated Coding Stages Assessment (CSA) (de Ruiter & Bers, 2021). Results from pre-post surveys and CSAs are presented in this article.

LITERATURE REVIEW

This section is organized around two broad areas of literature: 1) teaching coding for early elementary students, and 2) supporting teachers' knowledge and self-efficacy through effective PD. Understanding this existing landscape was essential not only for developing and implementing the CAL-ScratchJr virtual PD, but also for designing appropriate measures to explore the effectiveness of this virtual PD model for early elementary teachers.

Coding for Early Elementary Students

Coding, programming, computer science (CS), and computational thinking (CT) are related terms that apply to this growing area of educational interest. There are subtle but important differences among the terms. Whereas coding and programming (used interchangeably in this article) refer to the activity of reading and writing instructions that can be interpreted by a computer, CT refers to the set of "thought processes involved in expressing solutions as computational steps or algorithms that can be carried out by a computer" (K-12 CS Framework, 2016). Although CT was once thought to be limited to the CS discipline, it is now widely considered a universally applicable skill that can be learned with or without the use of computing devices (Barr & Stephenson, 2012; CSTA & ISTE, 2011; Guzdial, 2008; Wing, 2006, 2011; Yadav, Hong et al., 2016). Research in the last several decades has shown that young children can learn foundational coding and CT skills (e.g., algorithmic thinking, pattern recognition, and debugging) when exposed to developmentally appropriate technologies and pedagogical approaches (Bers, 2020; K-12 CS Framework, 2016; Relkin et al., 2021).

The ScratchJr Application. In this study, teachers were introduced to the free ScratchJr application, an introductory programming language for children ages 5-8 (Bers, 2018; Bers & Resnick, 2015; Bers & Sullivan, 2019; Govind et al., 2020; Leidl et al., 2017; Unahalekhaka & Bers, 2021). ScratchJr has a graphical user interface with a main project editor, and tools for selecting and drawing characters and backgrounds (Bers & Resnick, 2015). Children assemble colorful puzzle-shaped blocks in a left-to-right sequence to make characters move, jump, dance, and sing. An important ScratchJr feature is the prevalence of symbols and colors to support pre-literate children's user experiences. The blocks are organized into six categories, which are represented by different colors: yellow Trigger blocks, blue Motion blocks, purple Looks blocks, green Sound blocks, orange Control flow blocks, and red End blocks. The design of ScratchJr supports both the development of coding and CT skills and the ability for children to express themselves in creative ways. To elicit the most optimal outcomes for children, the ScratchJr app must be supported by an effective pedagogical approach, which is described next.

Coding as Another Language (CAL). Many existing pedagogical approaches view coding as belonging to or extending from STEM (Science, Technology, Engineering, and Mathematics) disciplines. Thus, coding is seen primarily as a problem-solving activity that engages children with abstraction and logic while developing CT (Wing, 2006, 2011). The Coding as Another Language (CAL) pedagogical approach, in contrast, understands coding not only as a problem-solving activity, but as an expressive activity that enables individuals to manipulate a symbolic and grammar system to communicate ideas and create shareable artifacts (Bers, 2019; Govind et al., 2021; Hassenfeld et al., 2020). From this perspective, programming languages are viewed akin to written natural languages (Fedorenko et al., 2019; Ivanova et al., 2020; Norman, 2017; Vee, 2013). For instance, assembling a sequence of ScratchJr blocks is analogous to stringing words together to make a meaningful sentence. Whereas the ScratchJr sequence begins with a yellow Trigger block and ends with a red End block, a sentence begins with a capital letter and concludes with an ending punctuation mark.

To make these kinds of coding and literacy connections concrete for educators and students, a supporting curriculum exemplifying the CAL pedagogical approach was developed. The curriculum--divided into Kindergarten, First Grade, and Second Grade units-- aims to place powerful CT concepts in conversation with complementary concepts in early literacy and language. For example, students learn about debugging and discuss the importance of editing in the writing process. Similarly, the CT concepts of decomposition and modularity (i.e., breaking down complex tasks into small, manageable parts) are introduced to teachers using the example of phonological awareness in literacy instruction. These broad coding and literacy connections are made more explicit through scaffolded lesson activities that are grade-appropriate and aligned to national academic standards (e.g., Common Core, K-12 CS Framework, ISTE Student Standards). Each CAL-ScratchJr lesson consists of a variety of unplugged and plugged activities, including songs, technology circles, structured and unstructured ScratchJr activities, and books. Pilot studies of the CAL-ScratchJr curriculum have demonstrated students' ability to learn coding and CT skills and to create more complex ScratchJr projects over the course of the lessons (de Ruiter & Bers, 2021; Unahalekhaka & Bers, 2021).

To support teachers in developing expertise in the ScratchJr programming language and familiarity with the CAL-ScratchJr curriculum, a PD workshop was devised. In the traditional PD model, expert facilitators would organize full-day in-person workshops, during which participants would engage in hands-on ScratchJr play and exploration of the CAL-ScratchJr curriculum and associated pedagogical strategies. Although plans were initially set to conduct in-person workshops with teachers, the COVID-19 pandemic necessitated the development of a virtual PD that would enable teachers to reap similar benefits as during an in-person workshop. To do so, it was essential to have a deep understanding of adult learning, teacher education, and the factors that contribute to effective PD. This next section explores these topics and illustrates how this literature helped to inform the development of the CAL-ScratchJr virtual PD activities.

Adult Learning and Teacher Professional Development

This research is grounded in adult learning theory, specifically andragogy, which is defined by Malcom Knowles as "the art and science of helping adults learn in contrast to pedagogy as the art and science of teaching children" (Knowles, 1980, p. 43). Knowles' foundational premise that adults learn differently from children is based on several assumptions. He argued that adults are independent, intrinsically driven learners who carry a multitude of prior experiences and must see the direct application of their learning in their immediate context. As such, training experiences for adults (i.e., professional development) that appropriately consider these assumptions are likely to be perceived more favorably by adults and produce more optimal learning outcomes (Knowles, 1980; Terehoff, 2002).

This study utilizes the theory of andragogy to conceptualize how adult educators gain knowledge and self-efficacy around the novel content area of coding, and how adults are able to facilitate their own learning in a virtual workshop setting. The virtual PD examined in this study was developed with these key assumptions in mind. For example, the PD included time for teachers to work on individual ScratchJr projects based on a topic of their choosing. Whereas teachers would often collaborate on projects during in-person workshops, the opportunity to work independently in a virtual setting forced teachers to take ownership of their learning. Just as most in-person workshops prioritize hands-on learning, the virtual PD was also designed to maximize teachers' hands-on experiences with ScratchJr and the CAL-ScratchJr lesson activities. Furthermore, activities were scaffolded to support teachers with varying levels of prior ScratchJr experience, and examples were provided from the CAL-ScratchJr curriculum to demonstrate how teachers could implement lessons with their students. Rather than learning every ScratchJr feature, teachers were exposed to common ScratchJr blocks and concepts and encouraged to explore features specific to their respective grade-level curriculum.

Teacher Knowledge. In addition to considering the unique needs of adult learners, successful PD must encompass both content knowledge and pedagogical knowledge (Shulman, 1986, 1987), and in the realm of CS education, also the knowledge of coding technologies (Mishra & Koehler, 2006). The Technological Pedagogical Content Knowledge (TPACK) framework understands the effective use of technology in the classroom as dependent on these three interacting factors. Accordingly, the virtual PD developed for this study addressed foundational CS and CT skills appropriate to introduce in early childhood (content knowledge), teaching strategies to promote positive behaviors and learning outcomes for children (pedagogical knowledge), and ScratchJr concepts and skills (technological knowledge). The TPACK framework additionally considers the specific classroom culture and context in which a particular technology is integrated. As such, the virtual PD included activities and discussions around how teachers would implement the CAL-ScratchJr curriculum in their respective settings. For example, one discussion prompt invited teachers to reflect and share how they might adapt a lesson to meet the needs of specific students in their classroom.

Teacher Self-Efficacy. Another important goal of the virtual PD was to enhance teachers' self-efficacy around teaching ScratchJr and confidence in implementing the CAL-ScratchJr curriculum. Self-efficacy is defined as "confidence or belief in one's own abilities to perform an action or activity necessary to achieve a goal or task" (Watson, 2006, p. 152). High-quality PD can make a positive impact on teachers' self-efficacy, especially when teachers receive direct instruction from knowledgeable facilitators and work through lessons as students themselves (Rich et al., 2017). Prior studies have indicated that teachers' confidence around teaching coding and CT can improve in as few as eight lessons or even in standalone workshops (Bers et al., 2013; Bower et al., 2017; Kaya et al., 2019). For instance, Kaya and colleagues (2019) taught preservice elementary teachers to use robotics, code.org, and the Zoombinis video game to promote CT and administered multiple self-efficacy measures before and after teachers' interactions with the various technologies. The researchers found a significant effect on teachers' personal self-efficacy for CT, as well as their self-efficacy to teach CT. In the virtual PD explored in this study, teachers' self-efficacy was fostered by providing structured and open-ended opportunities to engage with PD content, reflect on their learning, and ask questions to better facilitate teachers' understanding and confidence around implementing the CAL-ScratchJr curriculum.

Teacher Professional Development. The structure and format of a PD play a key role in how teachers' knowledge and self-efficacy are fostered. Researchers have identified five key characteristics of effective PD: content focus, active learning, collective participation, duration, and coherence (Avalos, 2011; Darling-Hammond, Hyler, & Gardner, 2017; Desimone, 2009; Guskey & Yoon, 2009; Odden & Picus, 2014). These characteristics were considered in the development of this virtual PD (see Table 1 for PD agenda). For instance, teachers were exclusively learning about ScratchJr and the CAL-ScratchJr curriculum (content focus) and engaged in multiple hands-on activities with the ScratchJr app (active learning). Because teachers had the option to choose which PD sessions they attended, many teachers picked the same days as their colleagues, enabling them to collaborate and share ideas with one another (collective participation). Although the PD sessions lasted only four hours, ongoing support was provided for teachers by the research team and district project coordinator (duration). Finally, at multiple points of the PD, teachers were asked to reflect on their current instructional practices and connect what they were learning in the PD to what they were already doing in their classrooms (coherence).

Table 1
Coding as Another Language (CAL)-ScratchJr
Virtual Professional Development Agenda

Part 1: Programming with ScratchJr		
Activity	**Duration**	**Description**
Introductions	20 min	Participants and PD facilitators greet one another.
Intro to ScratchJr	10 min	Participants learn about the history of ScratchJr and see a variety of projects that can be created using the block-based programming language.
Guided Explorations	30 min	Participants engage in a facilitator-guided, hands-on ScratchJr exploration using their own devices. The exploration activities are interspersed with formative “check for understanding” questions.
Brief Break	5 min	-
Advanced ScratchJr	15 min	Participants learn about advanced ScratchJr features, such as sending and receiving messages, inserting pictures, and parallel programming.
Create a ScratchJr Project	20 min	Participants listen to a children’s book read-aloud and recreate the story using ScratchJr.
Share Projects	15 min	Participants share their ScratchJr projects with others and practice sending their projects by email.
Closing	5 min	Q&A
Part 2: The CAL-ScratchJr Curriculum		
Activity	**Duration**	**Description**
Four Powerful Metaphors	30 min	Participants learn and reflect on four metaphors used as guiding frameworks for teaching coding in early childhood: coding as a playground, coding as another language, coding as a bridge, and coding as a palette of virtues (Bers, 2018, 2019, 2022).

Part 2: The CAL-ScratchJr Curriculum		
Activity	**Duration**	**Description**
Intro to CAL-ScratchJr	30 min	Participants are introduced to the CAL-ScratchJr curriculum, its overall scope and sequence, and sample lesson activities.
Brief Break	5 min	-
Lesson Deep-Dive	15 min	Participants explore one lesson from their grade-level unit and reflect on how they would implement the lesson in their respective classrooms.
Reflection	15 min	Participants share their reflections with others and discuss their ideas about curriculum implementation.
Research Study	15 min	Participants learn about prior research conducted on the CAL-ScratchJr curriculum and how they can contribute to future research.
Closing	10 min	Q&A

Studies have identified various factors that are likely to influence early elementary teachers' CS PD experiences. One important factor is the PD modality: physical/in-person versus virtual/online. Since the start of the COVID-19 pandemic, many researchers have adapted their in-person PD for K-12 teachers to virtual formats (Amiel & Blitz, 2021; Jocius et al., 2021; Mouza et al., 2022; Skuratowicz et al., 2021; Tsan et al., 2022). Their findings illustrate that virtual PD can be effective when teachers are provided flexible times for live sessions, a mix of collaborative and individual activities, and adequate support with new technology. Given that the CAL-ScratchJr PD was restructured to a virtual format with these considerations in mind, the question emerged as to whether teachers experiencing the virtual CAL-ScratchJr PD would achieve intended PD outcomes (i.e., exhibit growth in their coding knowledge and self-efficacy). Another important factor discussed in these previous studies is teachers' prior coding experience, which not only may impact their level of coding knowledge, but also how teachers feel about and approach new CS PD experiences. In this study, it was hypothesized that teachers with previous exposure to coding may have higher baselines of coding knowledge and self-efficacy and thus exhibit less change during the virtual PD.

Research Questions

This article focuses on the impact of the CAL-ScratchJr virtual PD on a group of 44 K-2 teachers in a public school district in the northeastern part of the U.S. Specifically, the following research questions are addressed:

1. What is the impact of the virtual PD on teachers' self-efficacy toward teaching coding and ScratchJr?
2. Does prior coding experience make a difference in a) teachers' self-efficacy (pre- and post-PD) around teaching coding, b) post-PD confidence around implementing the CAL-ScratchJr curriculum, c) post-PD concerns around implementing the CAL-ScratchJr curriculum, and d) PD training satisfaction?
3. What is the impact of the virtual PD on teachers' ScratchJr knowledge?
4. What is the relationship between teachers' ScratchJr knowledge and their prior coding experience, self-efficacy, curriculum confidence, concerns, and PD training satisfaction?

METHOD

Study Design

This is a quantitative research study, in which both before and after PD training data were collected. Before the PD began, the following baseline measures were collected from teachers in a pre-PD survey: their initial coding performance in ScratchJr (assessed one-on-one by research assistants), self-efficacy around teaching coding to students, demographics, years of teaching, and previous experience with coding and STEM education. After completing the baseline survey, teachers participated in two 2-hour synchronous PD trainings with an experienced trainer via Zoom. After attending the PD, teachers' coding performance in ScratchJr was again measured, along with their self-efficacy, curriculum confidence, concerns, and satisfaction of the PD training. This pre-post study design is a common design to examine the impact of short-term interventions (i.e., PD training in this study). Such design is cost efficient, requires less time from teachers and researchers, and provides useful information for the researchers to follow up (Thiese, 2014). Figure 1 displays the study design.

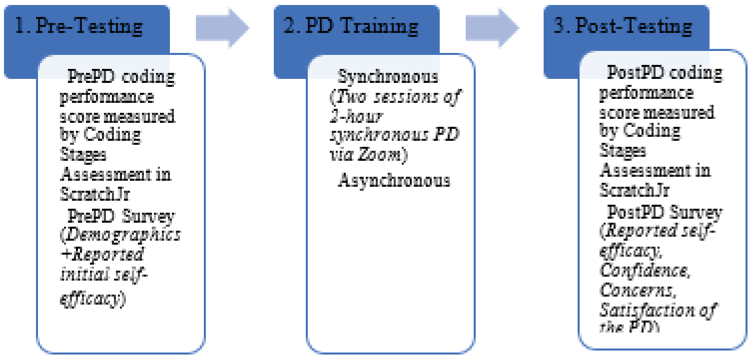

Figure 1. Study Design.

Synchronous PDs were hosted in July and August 2021, and all data were collected by December 2021. Depending on teachers' availability, three synchronous PD time slots were offered. 11 teachers participated in the first offered session (July 27, 2021), 14 teachers participated in the second session (August 4, 2021), and 15 teachers participated in the third session (August 12, 2021). Four teachers who could not attend the synchronous PD participated in an asynchronous PD, during which they watched a recorded video of the live synchronous session and completed written reflection activities. Due to the small size of participants in each PD, comparative analyses were not conducted based on the time or format of the PD. Nevertheless, a qualitative examination was done, and no substantial differences were found between these four teachers and the remaining sample.

Sample

The participants in this study consisted of 44 teachers from 12 elementary schools. Among those 44 teachers, 42 indicated their gender as female and two as male. Regarding race and ethnicity, the majority (n = 41) of the teachers were White. One teacher identified as Hispanic and two as biracial: American Indian/White and African American/White. Regarding the roles of the teachers, 31 indicated that they were classroom teachers (9 in Kindergarten, 10 in First Grade, and 12 in Second Grade), four indicated that they were Instructional Technology Resource Teachers (ITRT), and nine indicated other education roles, such as enrichment specialist, library media specialist, math specialist coach, multi-language learner (MLL) teacher, tech leader, technology director, and special educator. There is one teacher who could not do post-PD activities due to medical leave and another teacher moved to a different grade level after the PD training and only completed the post-PD survey but not the post-PD coding assessment. Therefore, the sample sizes in the post-PD survey and assessment are smaller than 44.

Regarding previous ScratchJr experience, 11 teachers reported that they had experience with ScratchJr while 33 reported that they did not have prior experience with ScratchJr. Those 11 teachers who had some ScratchJr experience indicated a range of descriptions including, "I have had some Professional Development and have used it in the library classroom," and "Took 12-hour Scratch Jr. Professional Development through Copernicus Learning Lab. Implanted in my classroom weekly. Buddy taught ScratchJr. with other first grade class." When asked about having any prior coding experience, 23 teachers responded "Yes" and 21 responded "No". With respect to post-secondary STEM education, 9 teachers responded "Yes" and 35 responded "No." Figure 2 below presents the number of years of teaching experience from all 44 teachers, which ranged from 0 to 45 years. The line indicates the mean and median of years of teaching, which was 18 years.

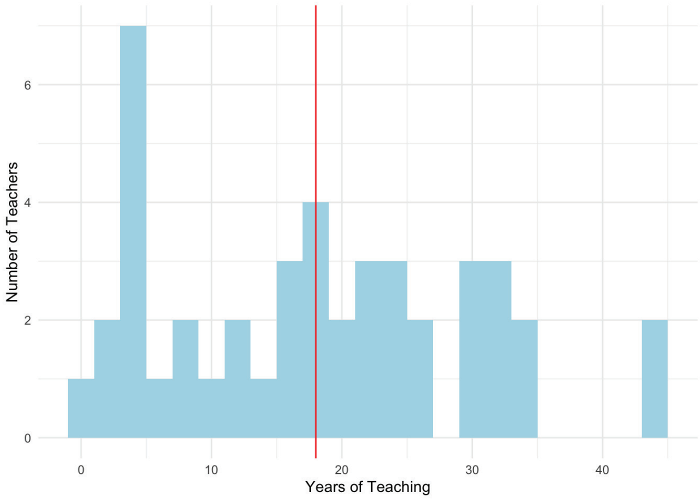

Figure 2. Distribution of Participants' Teaching Experience in Years.

Measures

As discussed in the study design, several measures were included in this study. Specifically, teachers' coding performance in ScratchJr was measured by trained researchers one-on-one using an established validated instrument, Coding Stages Assessment (CSA). Additionally, teachers' self-efficacy around teaching coding, teachers' confidence in teaching the CAL-ScratchJr curriculum, teachers' concerns about implementing the CAL-ScratchJr curriculum, and teachers' satisfaction of the virtual PD were measured by multiple items in a survey.

Coding Stages Assessment (CSA). The CSA aims to identify a learner's level of coding proficiency by asking the learner to create programs or explain how to achieve specific project goals by coding in the ScratchJr software application. The five coding stages are Emergent, Coding and Decoding, Fluency, New Knowledge, and Purposefulness, with each stage progressing in complexity (Bers, 2019). A weighted overall score is computed as an overall indicator CSA. Specifically, points in each stage are weighted such that points received in higher stages receive more weight (see details from de Ruiter & Bers, 2021). The assessment was administered to all participating teachers individually before and after the PD.

Self-Efficacy Around Teaching Coding. Teachers' self-efficacy around teaching coding and ScratchJr was surveyed in both the pre- and post-PD survey. The construct was measured by seven 5-point Likert-scale items (1= Strongly Disagree, …, 5= Strongly Agree; see Table 2). Six items were adopted from teachers' computing self-efficacy items developed by Rich and colleagues (2017). The seventh item was added by the research team to measure teachers' self-efficacy specifically around teaching ScratchJr: "I can teach ScratchJr to children." An aggregate self-efficacy score was computed using the sum of the seven items. The larger the value, the higher the teacher's self-efficacy around teaching coding and ScratchJr. The teacher self-efficacy measure showed a satisfactory reliability (α = 0.91). The difference between pre- and post-PD self-efficacy scores was also computed. Shapiro-Wilk normality test indicated that the self-efficacy difference was normally distributed ($W = 0.97$, $p = 0.26$). There were no significant outliers in the self-efficacy difference.

Table 2
Teacher Self-Efficacy Items

Construct	Item
Self-Efficacy	1. I can explain basic programming concepts to children (e.g., algorithms, loops, conditionals).
	2. I can plan out the logic for a computer program even if I don't know the specific programming language.
	3. I know where to find the resources to help students learn to code.
	4. I can find applications for coding that are relevant for students.
	5. I can integrate coding into my current curriculum.
	6. I can help students debug their code.
	7. I can teach ScratchJr to children.

Confidence in Teaching CAL-ScratchJr Curriculum. Teachers' confidence around teaching the CAL-ScratchJr curriculum was measured by eight 5-point Likert-scale items (1=Strongly Disagree, …, 5= Strongly Agree; see Table 3). These items were internally developed and tested by the research team. An aggregate score was obtained by taking the sum of all eight items. The higher the value, the more confident teachers were in teaching the CAL-ScratchJr curriculum. The teacher confidence measure showed a good reliability ($\alpha = 0.85$).

Table 3
Teacher Confidence Items

Construct	Item
Confidence	1. I am excited to teach the Coding as Another Language curriculum.
	2. I am confident in my ability to teach the Coding as Another Language curriculum.
	3. I am confident in my ability to teach ScratchJr.
	4. I know how to make the Coding as Another Language curriculum engaging for all students.
	5. I have the knowledge and skills I need to teach the Coding as Language curriculum effectively.
	6. I have the curriculum, tools, and resources I need to teach the Coding as Another Language effectively.
	7. I know how to formally assess students' Coding as Another Language curriculum learning and performance.
	8. I have a group of trusted colleagues that help me teach the Coding as Another Language curriculum effectively.

Concerns About CAL-ScratchJr Curriculum Implementation. Teachers' concerns about implementing the CAL-ScratchJr curriculum were measured using ten 5-point Likert-scale items (1=Strongly Disagree, …, 5= Strongly Agree; see Table 4). These items were also internally developed and tested by the research team, and an aggregate score was obtained by taking the sum of all ten items. The higher the value (i.e., the higher the agreement of having concerns), the more concerned teachers were about implementing the CAL-ScratchJr curriculum. The teacher concern measure showed an excellent reliability ($\alpha = 0.93$).

Table 4
Teacher Concern Items

Construct	Item
Concerns	1. Deciding whether I want to teach Coding as Another Language curriculum. 2. My ability to teach ScratchJr 3. Understanding what the Coding as Another Language curriculum is 4. Understanding what teaching the Coding as Another Language curriculum requires 5. Finding out what students need to know to be successful at coding 6. Preparing to teach the Coding as Another Language curriculum 7. Assessing my ability to teach the Coding as Another Language curriculum 8. Achieving intended student learning outcomes 9. Assessing student learning outcomes 10. Working with others to teach the Coding as Another Language curriculum

Satisfaction of the PD Training. Regarding the measure of teachers' satisfaction of the PD training, teachers rated on the following five 5-point Likert scale items (1=Needs a lot of improvement, 2= Needs some improvement, 3= Met my expectation, 4=Better than my expectation, 5= Couldn't be better): overall PD quality, instructor quality, session content, session format, session's learning climate, and time of training session. These items were internally developed and tested by the research team, and an aggregate score was obtained by taking the sum of all five items. The higher the value, the more satisfied teachers were with the PD training. The teacher PD satisfaction measure showed an excellent reliability ($\alpha = 0.94$).

Table 5 displays the analysis plan for each of the four research questions. Screening analyses were performed to investigate normality, linearity, and outliers. All statistical analyses were performed using R (Version 4.1.1, R Core Team, 2019).

Table 5
Research Questions and Data Analysis Plan

Research Question	Data Source	Analysis Method
RQ1: What is the impact of the virtual PD on teachers' self-efficacy toward teaching coding and ScratchJr?	Teacher Survey (Pre-PD, Post-PD)	Paired sample *t*-test
RQ2: Does prior teaching experience make a difference in a) teachers' self-efficacy around teaching coding, b) post-PD confidence around implementing the CAL-ScratchJr curriculum, c) concerns around implementing the CAL-ScratchJr curriculum, and d) PD training satisfaction?	Teacher Survey (Pre-PD, Post-PD)	Independent samples *t*-test
RQ3: What is the impact of the virtual PD on teachers' ScratchJr knowledge?	Coding Stages Assessment (Pre-PD, Post-PD)	Paired sample *t*-test
RQ4: What is the relationship between teachers' ScratchJr knowledge and their prior coding experiences, self-efficacy, confidence, concerns, and PD training satisfaction?	Coding Stages Assessment (Post-PD), Teacher Survey (Post-PD)	Mann-Whitney *U* test, Independent samples *t*-test, Pearson correlations

RESULTS

Teachers' Self-Efficacy

Table 6 displays teachers' pre- and post-PD self-efficacy scores. To answer the first research question, the paired sample *t*-test showed that teachers' self-efficacy increased significantly after the PD, with an average increase of 8.49 ($t(42) = 7.48$, $p < 0.001$). This finding suggests that the virtual PD training successfully improved teachers' self-efficacy in teaching coding and ScratchJr.

Table 6
Teachers' Self-Efficacy Before and After the Virtual PD

	N	*Min*	*Max*	*M*	*SD*
Pre-PD Self-Efficacy	44	7	34	19.6	8.0
Post-PD Self-Efficacy	43	9	35	27.9	5.1

Prior Coding Experience and PD Outcomes

The second research question was to examine the impact of prior coding experience on teachers' self-efficacy (pre- and post-PD) and post-PD curriculum confidence, concerns, and training satisfaction. The following paragraphs present the difference of each variable of interest by teachers' prior coding experience.

Self-Efficacy. The teachers who had previous coding experience had significantly higher reported initial self-efficacy before the PD compared to teachers who had no previous coding experience ($t(42) = 5.21$, $p < 0.001$; see Figure 3a). After the PD training, there was no significant difference in self-efficacy between teachers who did and did not have previous coding experience ($t(41) = 1.60$, $p = 0.11$). Thus, the teachers without previous coding experience showed a significantly higher increase in their change of self-efficacy compared to the teachers with previous coding experience ($t(41) = 4.26$, $p < 0.001$; see Figure 3b). This finding suggests that the PD helped all teachers improve their self-efficacy about teaching coding, but especially the teachers who did not have prior coding experience.

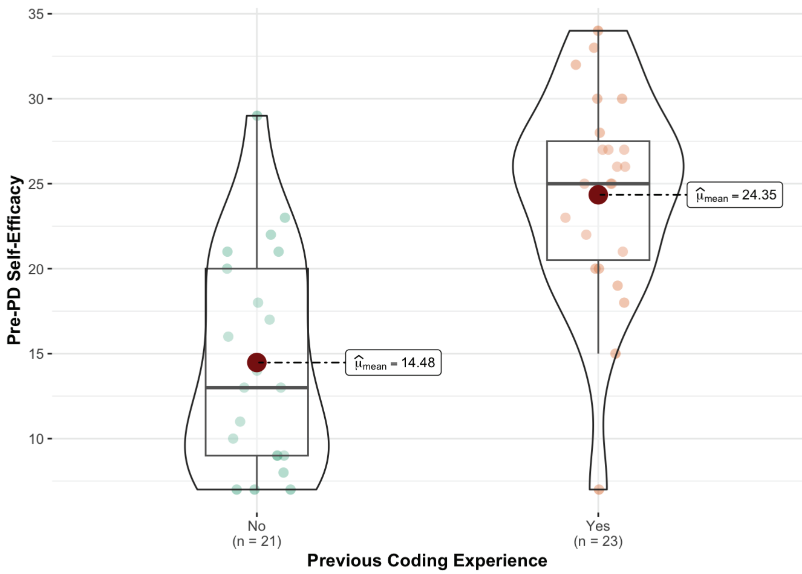

Figure 3a. Teachers' Pre-PD Self-Efficacy by Prior Coding Experience.

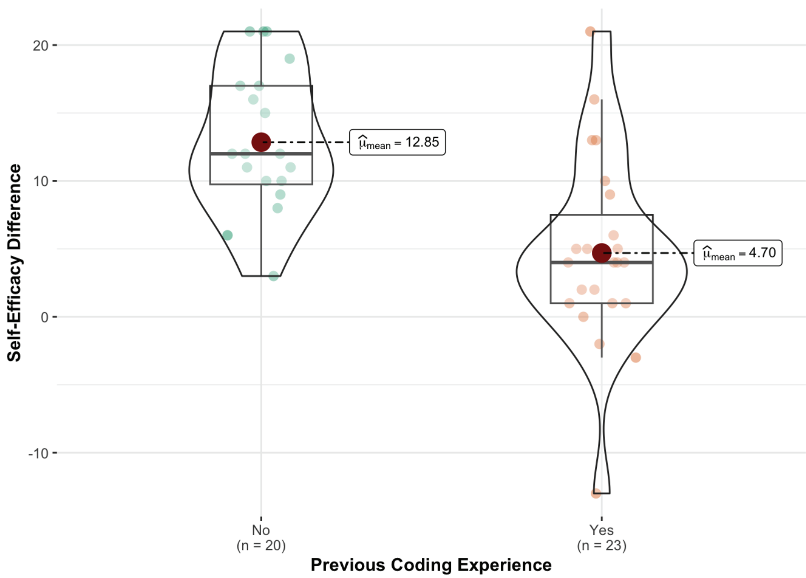

Figure 3b. Teachers' Change of Self-Efficacy Before and After the PD by Prior Coding Experience

Confidence. After the PD training, these 43 teachers reported a high level of confidence in teaching the CAL-ScratchJr curriculum, with a mean of 31.8 (*SD* = 4.5) out of a total score of 40 after aggregating the eight items. Whether or not teachers had previous experience, their confidence in implementing the CAL-ScratchJr curriculum after the PD training did not vary significantly ($t(41) = 1.52$, $p = 0.14$). This finding suggests that after the PD training, all teachers showed a high confidence in teaching the CAL-ScratchJr curriculum regardless of their prior coding experience.

Concerns. Whereas teachers were equally confident about teaching coding after the PD training, concerns about implementing the CAL-ScratchJr curriculum were significantly higher among those teachers who did not have previous coding experience than those who had previous coding experience ($t(41) = 2.36$, $p = 0.02$; see Figure 4). These 43 teachers had an aggregated rating of concerns about implementing the CAL-ScratchJr curriculum with a mean of 22.4 (*SD* = 7.6) out of a total score of 50 from the ten items. For example, teachers raised more concerns in the following aspects: "preparing to teach the Coding as Another Language curriculum," "achieving intended student learning outcomes," and "assessing student learning outcomes." Several possible reasons for these concerns are provided in the discussion.

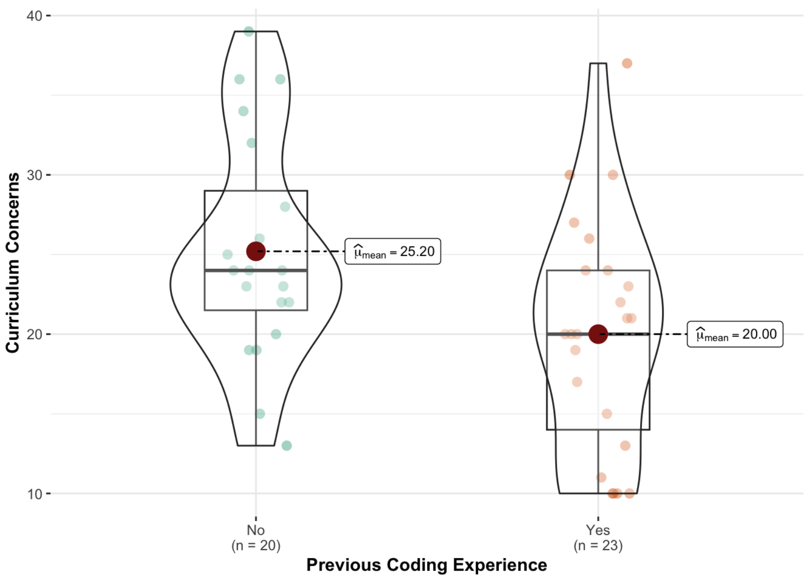

Figure 4. Teachers' Concerns by Prior Coding Experience.

Satisfaction. After the PD training, these 43 teachers had an aggregated rating of satisfaction on the virtual PD with a mean of 19.0 ($SD = 4.4$) out of 25 from the five items. No significant differences in teachers' satisfaction were found based on teachers' prior coding experience ($t(41) = 0.42$, $p = 0.68$).

In summary, these findings suggest that teachers' previous coding experience had some impact on their initial self-efficacy around teaching coding before the virtual PD but not after. While the virtual PD training helped improve all teachers' self-efficacy, it helped the teachers without previous coding experience even more. For example, on average teachers without previous coding experience showed a 12.85 point increase out of 35 points (36.7% increase) in their efficacy compared to teachers with the experience who showed a 4.7 point increase out of 35 points (13.4% increase). However, teachers without previous coding experience showed higher concerns on implementing the CAL-ScratchJr curriculum.

Teachers' ScratchJr Knowledge

There were 44 teachers who completed the pre-PD CSA and 42 teachers who completed the post-PD CSA (see Table 7). To address the third research question regarding the impact of the virtual PD on teachers' ScratchJr knowledge, pre-PD and post-PD CSA from 42 teachers were included in the paired sample *t*-test analysis. Results of the analysis indicated that these teachers' post-PD CSA was significantly higher than their pre-PD CSA ($t(41) = 11.40$, $p < 0.001$) with an average increase of 13.34 out of 39 points. Figure 6 presents the number of teachers in each CSA coding stage: Pre-Coding, Emergent, Coding and Decoding, Fluency, New Knowledge, and Purposefulness. Before the PD, most teachers' scores were categorized as "Pre-Coding" or "Emergent", whereas after the PD, over half of the teachers attained at least the "Fluency" stage and ten teachers reached the most advanced stage of "Purposefulness". These findings illustrate the virtual PD's positive impact on teachers' learning of ScratchJr skills and concepts.

Table 7

Teachers' Coding Stages Assessment (CSA) Scores Before and After the PD

	N	*Min*	*Max*	*M*	*SD*
Pre-PD CSA	44	2.2	37.6	10.7	7.5
Post-PD CSA	42	7.8	39	24.2	9.8

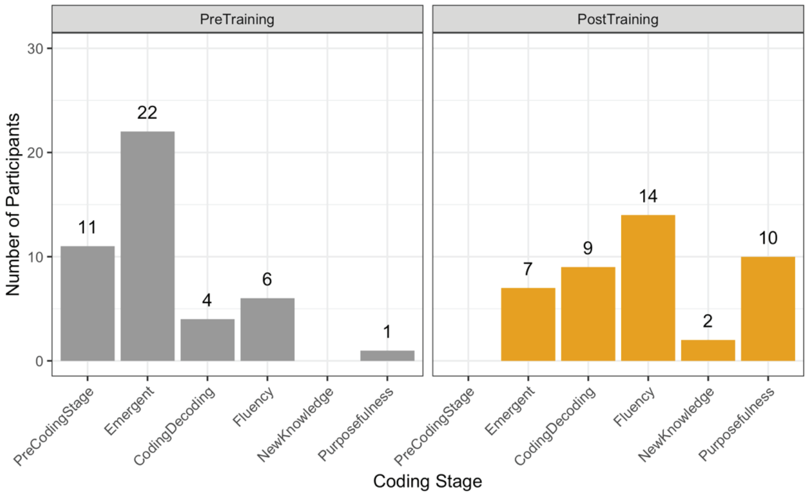

Figure 6. Teachers' Coding Stages Assessment (CSA) Levels Before and After the PD.

ScratchJr Knowledge and PD Outcomes

The last research question in this study was to examine the relationships between teachers' ScratchJr knowledge and their prior coding experiences, self-efficacy, curriculum confidence, concerns, and PD training satisfaction. Because normality assumptions were not met among teachers' pre-PD CSA in both groups with and without previous coding experience, Mann-Whitney *U* test was used to examine teachers' pre-PD CSA difference by their previous coding experience. The Mann-Whitney *U* test showed that there was no statistically significant difference due to prior coding experience on teachers' pre-PD CSA scores ($U = 142.5$, $p = 0.05$) although there was a trend that the teachers with prior coding experience showed higher pre-CSA (see Figure 7). After the PD, because normality assumptions were met, independent samples *t*-test was used to examine whether teachers' ScratchJr knowledge differed by their prior coding experience. No significant difference was found due to prior coding experience in teachers' post-PD CSA scores ($t(40) = 1.21$, $p = 0.23$). Again, the finding suggests that the PD training helped all teachers attain ScratchJr knowledge.

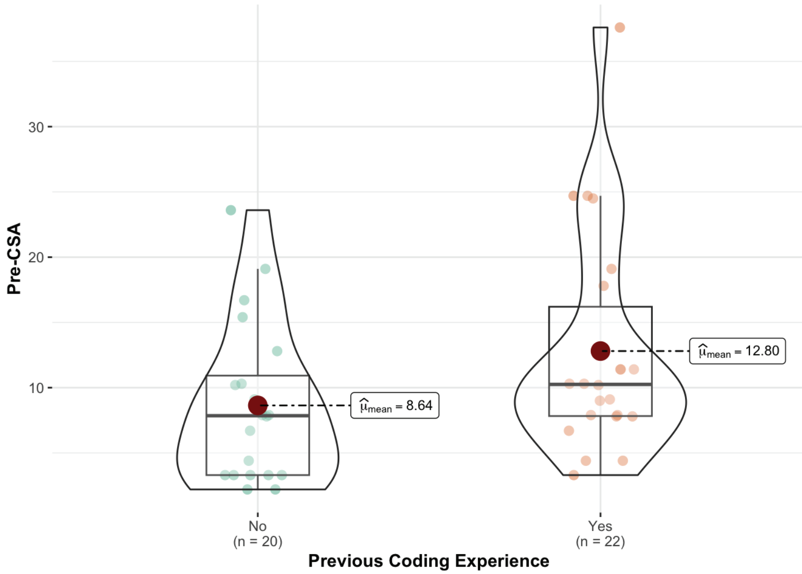

Figure 7. Teachers' Pre-CSA Distributions by Previous Coding Experience.

Pearson's correlation was computed to explore the relationships between teachers' ScratchJr knowledge (as measured by post-PD CSA) and post-PD self-efficacy, curriculum confidence, concerns, and PD satisfaction. Pearson's correlations showed that none of these relationships were statistically significant, $p > .05$. However, one may note that the sample size is 42, which does not have enough power to detect weak to medium correlations. Based on this finding, it cannot be concluded whether these factors significantly correlate to teachers' ScratchJr knowledge after PD. However, it is evident that none of these factors were *strongly* correlated to their ScratchJr knowledge after PD.

DISCUSSION

With the increasing popularity of coding education in the early grades comes the need to adequately support teachers with the knowledge and confidence to teach CS. While there are growing efforts to create and implement quality professional development for early elementary teachers, the additional challenges brought forth by the COVID-19 pandemic forced the opportunity to explore the design and implementation of a condensed, virtual PD. The goal of this study was to investigate the impact of a 4-hour virtual PD on a cohort of K-2 public school educators in the U.S. Specifically, the study aimed to address how the PD supported teachers' coding knowledge and self-efficacy around teaching coding, and to understand how teachers' prior experience with coding may have influenced their virtual PD experience.

Results from this study indicate that the virtual PD was successful in promoting teachers' self-efficacy around teaching coding and implementing the CAL-ScratchJr curriculum. Teachers who did not have prior coding experience had lower baseline self-efficacy scores but still had comparable post-PD self-efficacy scores as their colleagues with prior coding experience. This finding parallels what Bers, Seddighin, and Sullivan (2013) found in their 3-day, in-person workshop with early education professionals, and supports the initial hypothesis about the relationship between teachers' prior coding experience and change in self-efficacy. The finding that even a few hours of exposure can support teachers' self-efficacy around teaching coding serves promising for the future of early computer science professional development. For educators who find in-person opportunities lasting a week or longer to be too time-consuming or expensive—which Century and colleagues (2020) and others have identified as common barriers for educators—this type of condensed, virtual PD might be equally beneficial for their professional learning. On a larger scale, researchers and policymakers might look to expanding virtual PD opportunities to increase coding participation among early elementary teachers, a largely underrepresented group in K-12 computer science education.

Findings from this study also indicated that although teachers without prior coding experience experienced significant growth in their self-efficacy, they also reported greater concerns around implementing the CAL-ScratchJr curriculum. A possible explanation for this finding may be that these teachers felt increasingly more comfortable about their *own* ScratchJr proficiency but did not yet feel prepared to take what they learned in the virtual PD back into their classrooms. Furthermore, the teachers primarily reported concerns around preparing to teach the CAL-ScratchJr curriculum, as well as achieving and assessing student learning outcomes. Although qualitative examinations are considered beyond the scope of this article, there are several possible reasons for these concerns. The first concern related to preparedness may imply teachers' concerns about time needed to review lesson activities and prepare lesson materials. The latter concerns related to student learning outcomes make sense because the virtual PD primarily focused on teacher knowledge and self-efficacy, rather than assessment of student learning.

Moreover, the majority of teachers (75%) had never seen ScratchJr prior to this virtual PD, and all of them were seeing the CAL-ScratchJr curriculum for the very first time. Thus, we would expect teachers to spend more time with the lessons and implement them with students before being able to report confidently about student learning outcomes.

Although this study focused on findings from the virtual PD and not the curriculum implementation, the findings brought to light how teachers were connecting their learning from the virtual PD to their classroom practice. This is consistent with literature that shows that teachers benefit from PD models that are integrated into their teaching (Darling-Hammond et al., 2017; Lo, 2021; Roth et al., 2011; Williams et al., 2019). For example, in Roth and colleagues' (2011) study, teacher participants used what they had learned to plan and deliver their own lessons and discussed their teaching practice in a small group facilitated by a program trainer in the following school year. Future work will explore how the teachers who attended this virtual PD supported one another during curriculum implementation and used ongoing coaching supports to improve their lesson delivery. Additionally, Kuijpers, Houtveen, and Wubbels (2010) explained that PD integrated into teachers' teaching considers the school context, incorporates educational goals at school, and focuses on school procedures, roles, structures, and facilities that support teaching and learning. To better address educators' concerns about curriculum implementation, future work might consider PD participants' specific classroom and school contexts in the development and delivery of virtual PD trainings.

This study is the first of its kind to use the validated Coding Stages Assessment (CSA) to assess early elementary teachers' coding knowledge before and after a PD training. Although pre-PD CSA scores did not significantly differ between teachers with and without prior coding experience, the scores did indicate a slight difference between the two subgroups, demonstrating the assessment's ability to distinguish novice and experienced adult ScratchJr users. Furthermore, to support early elementary teachers' learning of coding, we must first be able to assess adults' coding knowledge beyond self-reported measures. This study provides initial evidence that teachers, regardless of prior coding experience, improved their ScratchJr coding knowledge. Due to a small sample size, post-CSA scores were not found to be significantly correlated to teachers' post-PD self-efficacy, concerns, nor PD satisfaction. This finding reveals that none of these factors were highly associated with teachers' ScratchJr knowledge after the PD. That is, teachers with various levels of self-efficacy, concerns, or PD satisfaction may all attain a high ScratchJr knowledge after the PD training.

Implications, Limitations, and Future Work

Findings from this work have implications for research and practice by informing the design of effective virtual PD opportunities aimed at broadening participation in early computer science education. Many lessons were learned in the development and analysis of this virtual PD. First and foremost, in line with Knowles' (1980) theory of andragogy, this virtual PD was developed with adult learners in mind. Only half the participants in this study had any prior coding experience, and only a quarter of participants had specifically identified having ScratchJr experience. The lack of coding exposure—in this sample and across the broader early education landscape (Code.org, 2022)—is justification enough to rethink and expand ways of bringing coding to early elementary educators and students. Seeing that the PD participants reported a high level of satisfaction and significantly improved in their coding knowledge, other PD providers can look to this virtual model as a promising way to engage early elementary educators in coding instruction. The success of this virtual, synchronous PD also brought into question the possible effectiveness of a virtual, *asynchronous* PD. How might educators experience a PD in which there is no live facilitator, and teachers engage in ScratchJr play and curriculum exploration on their own? Would PD participants be able to experience similar gains in their coding knowledge and self-efficacy in this modality? Future work will explore these questions.

Secondly, in line with Mishra and Koehler's (2006) TPACK framework, findings from this work reveal that early elementary teachers need support not only with the content and technological aspects of coding using the ScratchJr app, but also with the pedagogical aspect of implementing the CAL-ScratchJr curriculum in their respective classrooms. Future PD providers, regardless of the training modality, can look to this virtual PD and this study's takeaways to understand how to effectively engage PD participants and support their technological, pedagogical, and content knowledge of coding and their confidence around teaching coding. Thirdly, support and training may go beyond the PD training. As the post-PD concerns revealed, teachers showed more concerns on preparing to teach the CAL-ScratchJr curriculum. As a design of the larger project, we have designated Tech Leaders who received additional training to help teachers at their site and virtually coach to answer any questions teachers may experience during the curriculum implementation. This coaching model is based on prior work indicating that teachers with varied attitudes and beliefs about coding need varying levels of support, and that coaches can be well-positioned to provide differentiated support depending on teachers' needs. For instance, whereas teachers experiencing little self-efficacy and substantial concerns about teaching coding might benefit from co-teaching lessons with coaches, other teachers who feel more confident might benefit from checking in periodically with coaches to address specific lesson challenges (Govind, 2022). Future work will examine how teachers' use of coaching and other ongoing supports may have impacted their overall curriculum implementation experience.

Aside from the larger study design, from informal conversations, we noticed that teachers prefer to have peer support (e.g., creating lesson slides and sharing them with their colleagues). Future work may explore how peer support can enhance the effect of PD training on teachers' curriculum implementation experiences. Last but not least, for early elementary educators, PD training is just the beginning. With the challenges of having limited classroom time and the substantial cognitive load required to learn new things, teachers' willingness to try something new like ScratchJr and the CAL-ScratchJr curriculum in this study is a huge leap forward. We must acknowledge that the amount of time and effort needed to teach students soon after learning something new themselves could be overwhelming.

There are several limitations to this study. The number of teachers who participated in the virtual PD was relatively small ($N = 44$). Data were collected and analyzed from teachers who self-selected to participate in research activities, which limits the generalizability of findings. Currently, this study is being replicated with another U.S. public school district, which presents the future opportunity to answer these research questions with a larger sample and explore possible contextual differences between districts. Furthermore, because teachers could choose which of the three synchronous PD sessions to attend, each session was unique and the virtual learning community was shaped by the teachers in attendance. Thus, teachers' survey responses may be influenced by their experience in the specific virtual PD that they attended. A few teachers were unable to participate in the live sessions and instead participated asynchronously. Although their individual scores were not outliers in these findings, it may be interesting to explore in the future how teachers experience PD in different virtual modalities.

Collectively, findings from this study indicate that virtual PD can be effective in supporting early elementary teachers' coding knowledge and self-efficacy. Current work is also underway to develop a fully asynchronous virtual PD, accessible anywhere around the world as an online course hosted through an e-learning platform. This work opens the door to many future research questions about the effectiveness of different virtual PD experiences. In doing so, this work helps address the growing need to equip teachers with the knowledge and confidence to integrate CS successfully into their classrooms.

References

Amiel, D., & Blitz, C. (2021). An immersive virtual experience to drive change in computer science education. *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, 1–5. https://doi.org/10.1109/RESPECT51740.2021.9620681

Avalos, B. (2011). Teacher professional development in teaching and teacher education over ten years. *Teaching and Teacher Education*, *27*(1), 10–20.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *Inroads, 2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Bers, M. (2018). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education*, *3*(3), 08.

Bers, M. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom* (2nd ed). Routledge Press.

Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education, 6*(4), 499–528.

Bers, M., Govind, M., & Relkin, E. (2022). Coding as another language: Computational thinking, robotics, and literacy in first and second grade. In A. Ottenbreit-Leftwich, & A. Yadav, A. (Eds.), *Computational thinking in preK-5: Empirical evidence for integration and future directions* (pp. 30–38). ACM and the Robin Hood Learning + Technology Fund.

Bers, M., & Resnick, M. (2015). *The official ScratchJr book: Help your kids learn to code*. No Starch Press.

Bers, M., Seddighin, S., & Sullivan, A. (2013). Ready for robotics: Bringing together the T and E of STEM in early childhood teacher education. *Journal of Technology and Teacher Education*, *21*(3), 355–377.

Bers, M. U. & Sullivan, A. (2019). Computer science education in early childhood: The case of ScratchJr. *Journal of Information Technology Education: Innovations in Practice*, *18*, 113–138.

Bower, M., Wood, L., Lai, J. W. M., Howe, C., Lister, R., Mason, R ..., & Veal, J. (2017). Improving the computational thinking pedagogical capabilities of school teachers. *Australian Journal of Teacher Education, 42*(3), 53–72. https://doi.org/10.14221/ajte.2017v42n3.4

Century, J., Ferris, K. A., & Zuo, H. (2020). Finding time for computer science in the elementary school day: A quasi-experimental study of a transdisciplinary problem-based learning approach. *International Journal of STEM Education*, *7*(1), 1–16.

Code.org. (2022). https://code.org/

CSTA & ISTE (2011). *Operational definition of computational thinking for K–12 education.* https://id.iste.org/docs/ctdocuments/computational-thinking-operational-definitionflyer.pdf

Darling-Hammond, L., Hyler, M. E., & Gardner, M. (2017). *Effective teacher professional development*. Learning Policy Institute.

de Ruiter, L. E., & Bers, M. U. (2021). The Coding Stages Assessment: Development and validation of an instrument for assessing young children's proficiency in the ScratchJr programming language. *Computer Science Education*, *32*(4), 388–417.

Desimone, L. (2009). Improving impact studies of teachers' professional development: Toward better conceptualizations and measures. *Educational Researcher, 38*(3), 181–199. https://doi.org/10.3102/0013189X08331140

Fedorenko, E., Ivanova, A., Dhamala, R., & Bers, M. U. (2019). The language of programming: A cognitive perspective. *Trends in Cognitive Sciences, 23*(7), 525–528. https://doi.org/10.1016/j.tics.2019.04.010

Govind, M. (2022). *A qualitative examination of second grade teachers' experiences and attitudes around coding and robotics education* (Publication No. 29169893) [Doctoral dissertation, Tufts University]. ProQuest Dissertations & Theses Global.

Govind, M., Hassenfeld, Z., & de Ruiter, L. (2021). Rhyme and reason: The connections among coding, computational thinking, and literacy. In M. U. Bers (Ed.), *Teaching computational thinking and coding to young children.* IGI Global.

Govind, M., Relkin, E., & Bers, M. U. (2020). Engaging children and parents to code together using the ScratchJr app. *Visitor Studies, 23*(1), 46–65. https://doi.org/10.1080/10645578.2020.1732184

Guskey, T., & Yoon, K. S. (2009). What *Works in Professional Development*? *Phi Delta Kappan*, *90*(7), 495–500. https://doi.org/10.1177/003172170909000709

Guzdial, M. (2008). Education: Paving the way for computational thinking. *Communications of the ACM*, *51*(8), 25–27. https://doi.org/10.1145/1378704.1378713

Hassenfeld, Z. R., Govind, M., de Ruiter, L. E., & Bers, M. U. (2020). If you can program, you can write: Learning introductory programming across literacy levels. *Journal of Information Technology Education: Research, 19*, 65–85. https://doi.org/10.28945/4509

Ivanova, A. A., Srikant, S., Sueoka, Y., Kean, H. H., Dhamala, R., O'Reilly, U.M., …, & Fedorenko, E. (2020). Comprehension of computer code relies primarily on domain-general executive resources. *Elife*, *9*, e58906.

Jocius, R., Joshi, D., Albert, J., Barnes, T., Robinson, R., Cateté, V., …, & Andrews, A. (2021). The virtual pivot: Transitioning computational thinking PD for middle and high school content area teachers. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE'21).* Association for Computing Machinery, 1198–1204. https://doi.org/10.1145/3408877.3432558

K-12 Computer Science Framework Steering Committee (2016). *K–12 computer science framework*. https://k12cs.org.

Kaya, E., Yesilyurt, E., Newley, A., & Deniz, H. (2019). Examining the impact of a computational thinking intervention on pre-service elementary science teachers' computational thinking teaching efficacy beliefs, interest, and confidence. *Journal of Computers in Mathematics and Science Teaching, 38*(4), 385–392.

Knowles, M. S. (1980). *The modern practice of adult education: From andragogy to pedagogy.* Cambridge Adult Education.

Kuijpers, J. M., Houtveen, A. A., & Wubbels, T. (2010). An integrated professional development model for effective teaching. *Teaching and Teacher Education*, *26*(8), 1687–1694.

Leidl, K., Bers, M. U., & Mihm, C. (2017). Programming with ScratchJr: A review of the first year of user analytics: *Proceedings of the International Conference on Computational Thinking Education* (pp. 116–121).

Lo. (2021). Design principles for effective teacher professional development in integrated STEM Education: A Systematic Review. *Educational Technology & Society*, *24*(4), 136–152.

Mishra, P., & Koehler, M. J. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record*, *108*(6), 1017–1054.

Mouza, C., Mead, H., Alkhateeb, B., & Pollock, L. (2022). A virtual professional development program for computer science education during COVID-19. *TechTrends, 66,* 436–449.

Norman, K. L. (2017). *Cyberpsychology: An introduction to human-computer interaction*. Cambridge University Press.

Odden, A., & Picus, L. O. (2014). School finance: A policy perspective. McGraw Hill.

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education, 169*, 104222. https://doi.org/10.1016/j.compedu.2021.104222

Relkin, E., Govind, M., Tsiang, J., & Bers, M. (2020). How parents support children's informal learning experiences with robots. *Journal of Research in STEM Education, 6*(1), 39–51.

Rich, P. J., & Hodges, C. B. (2017). *Emerging research, practice, and policy on computational thinking*. Springer.

Rich, P. J., Jones, B. L., Belikov, O., Yoshikawa, E., & Perkins, M. (2017). Computing and engineering in elementary school: The effect of year-long training on elementary teacher self-efficacy and beliefs about teaching computing and engineering. *International Journal of Computer Science Education in Schools*, *1*(1), 1–20. https://doi.org/10.21585/ijcses.v1i1.6

Rich, P. J., Mason, S. L., & O'Leary, J. (2021). Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. *Computers & Education*, *168*, 104196. https://doi.org/10.1016/j.compedu.2021.104196

Roth, K. J., Garnier, H. E., Chen, C., Lemmens, M., Schwille, K., & Wickler, N. I. Z. (2011). Videobased lesson analysis: Effective science PD for teacher and student learning. *Journal of Research in Science Teaching, 48*(2),117–148.

Shulman, L. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard Educational Review*, *57*(1), 1–23.

Shulman, L. S. (1986). Those who understand: Knowledge growth in teaching. *Educational researcher*, *15*(2), 4–14.

Skuratowicz, E., Vanderberg, M., Hung, E. E., Krause, G., Bradley, D., & Wilson. J. P. (2021). I felt like we were actually going somewhere: Adapting summer professional development for elementary teachers to a virtual experience during COVID-19. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 739–745). Association for Computing Machinery. https://doi.org/10.1145/3408877.3432482

Stanton, J., Goldsmith, L., Adrion, W. R., Dunton, S., Hendrickson, K. A., Peterfreund, A., ..., & Zinth, J. D. (2017). *State of the states landscape report: State- level policies supporting equitable K–12 computer science education*. https://www.edc.org/state-states-landscapereport-state-level-policies-supporting-equitable-k-12-computer- science

Terehoff, I. I. (2002). Elements of adult learning in teacher professional development. *NASSP Bulletin, 86*, 65–77. https://doi.org/10.1177/019263650208663207

Thiese, M. S. (2014). Observational and interventional study design types: An overview. *Biochemia Medica*, *24*(2), 199–210. https://doi.org/10.11613/BM.2014.022

Tsan, J., Coenraad, M., Crenshaw, Z., Palmer, J., Eatinger, D., Beck, K., ..., & Franklin, D. (2022). Reimagining professional development for K-8 CS teachers: Evaluating a virtual, diffuse model. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V* (pp. 530–536). Association for Computing Machinery. https://doi.org/10.1145/3478431.3499361

Unahalekhaka, A., & Bers, M. U. (2021). Taking coding home: Analysis of ScratchJr usage in home and school settings. *Educational Technology Research and Development, 69*(3), 1579–1598. https://doi.org/10.1007/s11423-021-10011-w

Vee, A. (2013). Understanding computer programming as a literacy. *Literacy in Composition Studies, 1*(2), 42–64. https://doi.org/10.21623/1.1.2.4

Watson, G. (2006). Technology professional development: long-term effects on teacher self-efficacy. *Journal of Technology and Teacher Education, 14*(1), 151–165.

Williams, T., Singer, J., Krikorian, J., Rakes, C., & Ross, J. (2019). Measuring pedagogy and the integration of engineering design in STEM classrooms. *Journal of Science Education and Technology, 28*(3), 179–194.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://www.cs.cmu.edu/*15110-s13/Wing06-ct.pdf

Wing, J. (2011). Research notebook: Computational thinking—What and why? *The Link Magazine*, *6*, 20–23. https://www.cs.cmu.edu/link/research-notebookcomputational-thinking-what-and-why

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

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding a 21st century problem solving in K-12 classrooms. *TechTrends, 60*(6), 565–568. https://doi.org/10.1007/s11528-016-0087-7

Yu, J., & Roque, R. (2018). A survey of computational kits for young children. *Proceedings of the 17th ACM Conference on Interaction Design and Children - IDC '18* (pp. 289–299). Association for Computing Machinery. https://doi.org/10.1145/3202185.3202738