International Journal of
STEM Education

REVIEW

Open Access

Bringing computational thinking into classrooms: a systematic review on supporting teachers in integrating computational thinking into K-12 classrooms

Zhichun Liu[1*], Zarina Gearty[2], Eleanor Richard[2], Chandra Hawley Orrill[3], Shakhnoza Kayumova[2] and Ramprasad Balasubramanian[2]

Abstract
Although computational thinking (CT) is becoming increasingly prevalent in K-12 education, many teachers find it challenging to integrate it with their classroom learning. In this systematic review, we have reviewed empirical evidence on teachers' computational-thinking-focused professional development (PD). The findings depict the landscape of what has been done in terms of how PDs have been designed, how CT has been conceptualized, how learning outcomes have been assessed, and how teachers have been supported in integrating CT into their teaching practices. We have further summarized the lessons learned from the PDs and discussed the gaps as the field moves forward. These findings shed light on supporting teachers as the first step to creating an effective model for CT learning and development in K-12 education.

Keywords Computational thinking, Teacher professional learning, K-12, Systematic review

Introduction

Computational thinking (CT) is increasingly important in K-12 education as a set of skills and processes required for all students. Despite the prevalence of instructional materials designed to teach programming and the readily accessible environments designed to introduce CT and programming to K-12 students (e.g., Scratch), broad access and adoption of CT teaching and learning still faces many challenges. Because CT is a relatively new construct in K-12, there are limited support systems in place to help teachers and students connect their existing knowledge and skills with those needed for CT (Ketelhut et al., 2020; Weintrop et al., 2016). As a result, challenges persist in reaching all students, including those with diverse backgrounds. While there have been various CT initiatives such as CT for All by the International Society for Technology in Education and the Computer Science Teachers Association, National Curriculum of Computing Programmes of Study in the United Kingdom, there is still much unknown about how best to support teachers with the integration of CT into their classroom instruction (Ketelhut et al., 2020; Kong et al., 2023). In this paper, we aim at investigating what is known about professional development designed to support teachers in learning and implementing CT, particularly in the STEM context in K-12 classrooms.

*Correspondence:
Zhichun Liu
liulukas91@gmail.com
[1] Faculty of Education, The University of Hong Kong, Hong Kong, SAR, China
[2] Kaput Center for Research and Innovation in STEM Education, University of Massachusetts Dartmouth, Dartmouth, MA 02747, USA
[3] Rethink Learning Labs, Fall River, MA, USA

Computational thinking in K-12 STEM classrooms

Although the operationalizations of CT varies in the research community (e.g., Grover & Pea, 2013; Israel et al., 2015; Kafai & Proctor, 2022; Lodi & Martini, 2021; Shute et al., 2017), CT has generally come to refer to the ability to analyze and solve various problems "in a form that can be effectively carried out by an information-processing agent" (Wing, 2010, p. 34). As a "universally applicable attitude and skill set" (Wing, 2006), CT is not specifically about using computational tools or tech. Instead, it is focused on cognitive competencies (e.g., abstraction) and dispositions (e.g., persistence) that are essential for meaningful problem solving. For example, the International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) describe CT as a problem-solving process that includes characteristics such as (a) problem decomposition, (b) algorithmic thinking, (c) abstraction and modeling, (d) data analysis, and (e) generalization and transfer. Through the development of CT skills, students also learn to (a) deal with complex and open-ended problems; (b) persist through difficult problems; (c) tolerate ambiguity; and (d) communicate and collaborate with others (Barr & Stephenson, 2011; ISTE & CSTA, 2011).

These cognitive and social skills are also important for STEM learning (Fofang et al., 2020; Scherer et al., 2019; Tran, 2019). As a result, researchers and practitioners have been pursuing the intersection of CT and STEM learning as a way to integrate CT into the classroom (Grover & Pea, 2013; Lee et al., 2019; Li et al., 2020; Sengupta et al., 2018). For example, Weintrop and colleagues (2016) mapped high school math and science teaching practices and CT practices. They identified 22 CT practices that closely relate to STEM learning (e.g., using a computational model to understand a concept, preparing problems for computational solutions, and investigating a complex system as a whole).

Too often, though, access to CT learning opportunities has been limited to informal learning settings such as after-school clubs (Lye & Koh, 2014) or reserved for only particular groups of students such as those with access to AP Computer Science courses (Hestness et al., 2018). These opportunities fail to reach a diverse student population, perpetuating the problem of underrepresentation of females and minority groups (Coenraad et al., 2022; Joshi & Jain, 2018). Studies have shown that providing underrepresented populations with CT learning opportunities can positively impact their confidence and interest in STEM careers (Gomoll et al., 2016; Leonard et al., 2016; McGonagle et al., 2014). Thus, there is an opportunity to support STEM learning and provide more students with access to CT by incorporating it into K-12 STEM learning (Kafai & Proctor, 2022).

Computational thinking professional learning for K-12 teachers

As noted above, one issue preventing integration of CT in classrooms is teachers' limited experience with CT. One viable way to address this issue is through the creation of effective professional development (PD) experiences focused on integrating CT into STEM learning. In the past decade, efforts to introduce CT have typically been focused on creating prepackaged curricular materials for students and teachers to learn to code (e.g., CSforAll, CS Unplugged, Creative Computing Curriculum). PDs for these materials are typically focused on preparing teachers to implement the curriculum in their own classroom by introducing them to new CT-related tools (Curzon, 2013; Sabin et al., 2018), which often leads to challenges in long-term professional development and sustainable changes (Brinkerhoff, 2006).

Numerous research efforts have been invested in identifying the challenges and solutions to teachers' need for PD to effectively integrate CT into regular classroom settings (e.g., Dong et al., 2019; Hamner et al., 2016; Ketelhut et al., 2020; Love et al., 2022; Pokorny & White, 2012). However, PD designed to support teachers' CT integration faces several critical challenges: first, CT has been defined and operationalized differently across contexts due to varying theoretical orientations and practical constraints (Kafai & Proctor, 2022). In one recent paper, Lodi and Martini (2021) discussed two views of CT and how CT benefits other subjects from a historical perspective (i.e., the Computer Science centrality in Wing's proposal versus the constructionism in Papert's proposal). They argued that CT has been abused as a buzzword without attending to the educational context and the epistemological foundations. The ongoing discussion and confusion surrounding the definition of CT have led to varied interpretations and operationalization in implementing CT in K-12 teachers' professional learning.

Second, the literature suggests that many PDs for CT development have been short and limited to workshop-style, one-shot experiences (Liu et al., 2012) without clear ties to classroom teaching and learning. While several practical arguments exist for using workshop formats, research has shown that extended PDs are more impactful (Darling-Hammond et al., 2017). Similarly, research has shown (van Veen et al., 2012) that PDs are most effective when they are relevant to teachers' classrooms.

Third, research indicates that PDs are most effective when they are relevant to teachers' classrooms (Desimone & Garet, 2015). Although CT does not just mean computer science and the ability to code, many CT PDs fail to make explicit connections between CT concepts and teachers' specific subject areas or grade levels. Many CT PDs fail to adequately address how CT concepts can be meaningfully integrated into STEM areas (Ketelhut et al., 2020). Teachers often struggle to see how CT principles apply to their particular discipline, making it challenging to incorporate these ideas into their existing classrooms. Lane et al. (2023) worked with experienced teachers to integrate Python in physics classrooms and reported their re-novicing professional learning experiences as they learned about programming and CT integration.

Finally, effective PD may have multiple endpoints from immediate reactions to long-term student performance (Desimone, 2009). Despite the extensive efforts in CT PD, there is a lack of consensus on how to assess teachers' CT knowledge and skills, as well as their ability to integrate CT into their teaching practices. This makes it difficult to evaluate the effectiveness of PD programs and support teachers' ongoing teachers' professional learning (Rich et al., 2021).

Research gap and scope of the review

Although the body of literature on CT PD is growing rapidly, there is a notable lack of comprehensive synthesis that specifically examines PD models and programs designed to support teachers in this integration process. While previous reviews have focused on introducing CT to teachers (Hsu et al., 2018), teacher preconceptions (Cabrera, 2019), or student learning (Wang et al., 2022, Zhang et al., 2023), little attention has been paid to synthesizing the various approaches to teacher PD for CT integration. This gap in the literature limits our understanding of effective strategies for preparing teachers to incorporate CT across STEM subjects.

Therefore, this systematic review aims to analyze existing CT PD models and programs for K-12 teachers, focusing on their learning goals, practical implementation, strategies for CT integration, and assessment methods. By examining these aspects, we seek to identify past practices in supporting teachers to learn CT and integrate CT into their classrooms. To that end, the following research questions will be answered:

RQ1: What were the learning goals and topics of the existing CT PD models and programs?

RQ2: What did CT PD implementations in practice look like (i.e., sample size, duration of the PD, and computing tools involved)?

RQ3: How were the PD learning experiences designed to support teachers in integrating CT into STEM classrooms?

RQ4: How were teachers assessed in the PD models and programs?

Method

Literature search and search strategies

This study is a systematic literature review based on the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) protocol, which guides the pre-established, systematic, and transparent methods of review (Moher et al., 2015). The search strategy, eligibility criteria, and data evaluation plan were pre-established. We conducted our search across four bibliography databases: (a) ERIC, (b) Web of Science, (c) IEEE Xplore, and (d) ACM digital library. We employed a sensitive search strategy (Relevo, 2012) to capture any potential candidate article. The search keywords used were the topic word "computational thinking", a set of synonyms for professional development (i.e., "professional development", "teacher development", "training", "intervention", and "workshop") and a set of synonyms for the potential outcomes of professional development (i.e., "teacher thinking", "teacher perception", and "teacher knowledge"). The search terms combined CT + professional-development-related terms or CT + outcome-related terms. To capture a wide range of empirical evidence, our search was not limited to peer-reviewed academic publications (e.g., books, book chapters, dissertations, and theses). Table 1 shows the list of the search results from each combination of the keywords. As a result, the search returns 3203 unique articles (see Fig. 1 for the PRISMA flowchart and details of search results). The search was done in June 2023.

Eligibility criteria and screening

Eligibility criteria

We pre-established the seven eligibility criteria (i.e., content, time, participant, intervention, evaluation, language, and publication relevance) for inclusion and exclusion (Table 2). It is worth noting that we did not exclude candidate articles based on any specific CT definitions

Table 1 Number of results from each search term

	Search term	Number of results
1	"Computational thinking"+"professional Development"	822
2	"Computational thinking"+"teacher development"	133
3	"Computational thinking"+"training"	1658
4	"Computational thinking"+"intervention"	730
5	"Computational thinking"+"workshop"	1177
6	"Computational thinking"+"teacher thinking"	3
7	"Computational thinking"+"teacher perception"	26
8	"Computational thinking"+"teacher knowledge"	71
	Total	4600

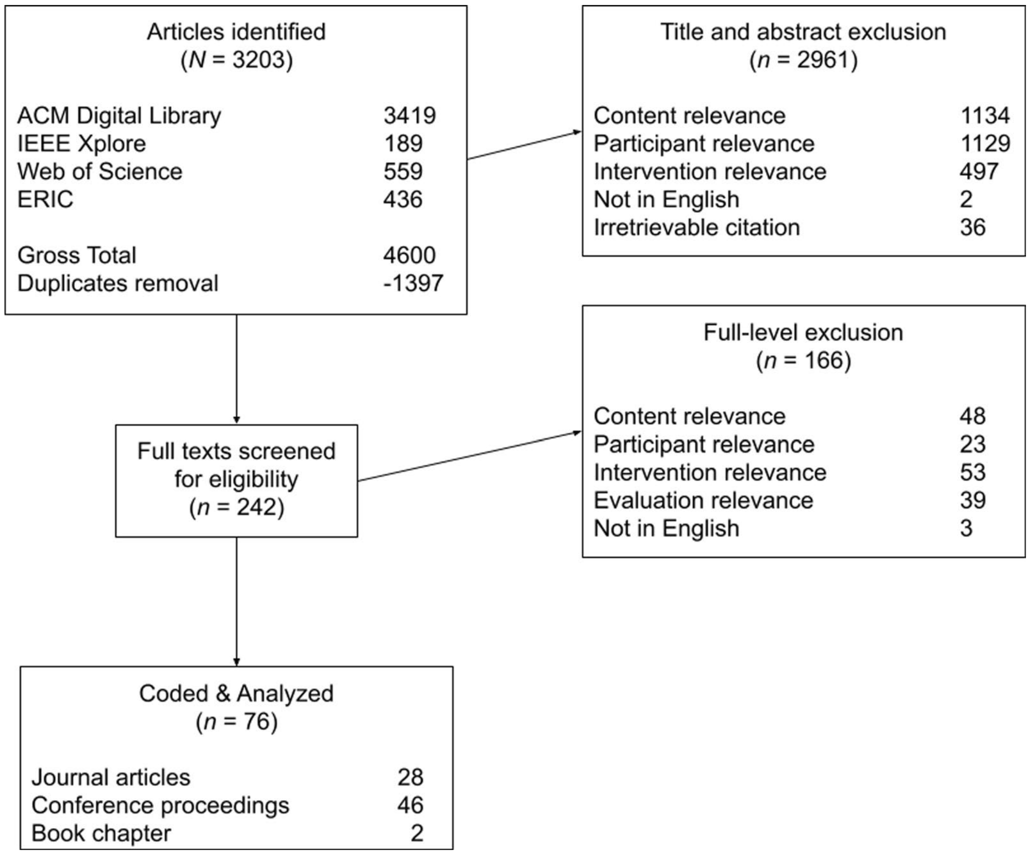

Fig. 1 PRISMA flowchart

Table 2 Tentative eligibility criteria

Eligibility criteria	Description
Content relevance	The candidate study should self-identify as promoting computational thinking (e.g., computational thinking knowledge, skills, or practice in teaching) instead of the knowledge *only* regarding computer science (e.g., programming syntax or computation)
Time relevance	The candidate study should be published after 2006 when the concept of computational thinking was coined by Wing (2006)
Participant relevance	The participants of the candidate study should be K-12 in-service teachers
Intervention relevance	The candidate study should implement one or more professional development modules (e.g., training or workshop) as the intervention and should include sufficient design details (e.g., objectives and activities)
Evaluation relevance	The candidate study should include quantitative or qualitative evaluation to assess the outcome of the professional development intervention
Language relevance	The study is reported in English
Publication relevance	Both peer-reviewed and non-peer reviewed are included

because we aimed to include a wide range of CT PD. We relied on whether the authors identified their PD as CT in their titles, abstracts, or keywords. In addition, the challenges and needs of in-service teachers and pre-service teachers may vary enough that the PD designs and learning experiences are heterogeneous. Therefore, the data collection only focused on in-service teachers and studies with only pre-service teachers were excluded.

Screening process

The authors of this paper participated as coders in the screening and data evaluation process. We initially assigned each unique article to one coder, and the coder screened the articles at the title and abstract level based on the first seven eligibility criteria. To ensure standardization in the title and abstract-level screening, a small set of articles ($n = 12$) were randomly selected to be screened by all three coders. The inter-rater reliability for these 12 articles was 100%. After the screening, full texts of the remaining articles were retrieved. After evaluation, 2961 articles were removed, and 242 articles were included for full-text screening.

We repeated this process when we moved to full-text screening. Each coder read a small set of articles ($n=9$) randomly selected from the candidate articles. The inter-rater reliability at that training was 100%. Each remaining article was assigned to at least two coders who coded independently. The inter-rater reliability reached 88.76%. Discrepancies between coders were resolved through discussions at group meetings. In all, we included 76 articles for full-text screening. Details of inclusion and exclusion are illustrated in Fig. 1.

Data evaluation

We began with an initial coding scheme that evolved through rounds of iterations (shown in Table 3). At the initial training session, the lead coder explained the coding form as well as the descriptions of each coded variable to the rest of the coders. Nine articles were used as training examples and were coded by all coders to establish agreement within the group. The coders resolved all questions and discrepancies at the training session. Subsequently, each remaining article was assigned to one individual coder. The coding results were discussed and shared at the weekly group meetings to ensure consensus among all coders and to avoid major discrepancies.

Table 3 shows the information extracted from each selected study. We first extracted the author-stated goal(s) of each PD from the article. Based on those goals, we identified various topics of PDs. Ideally, the goals and topics of each PD should focus on or contain CT. However, we recognized that all CT activities are not branded as CT in practice (e.g., robotics or CS unplugged). Therefore, we decided to include a wide range of CT-related topical words. Because CT lacks a universal definition (Shute et al., 2017) and different conceptualizations may lead to different program goals, we coded the main topic and goal of each PD as well as how CT was conceptualized and operationalized using Lodi and Martini (2021).

We recorded the basic information from each study (i.e., author, year, publication status, sample size, participant information, PD duration, and programs/curriculums used). In terms of the implementation, we

Table 3 The coding scheme

Coded information	Description
Study information	
Author	Names of the authors
Year	Date of publication
Published/ unpublished work	Whether the study is published or not
Participants information	
Sample size	Number of participants
Participant information	Description of the participants (subject and grade)
PD goal and CT conceptualization	
Goal of the PD	The goal of the PD
Goal disposition	Whether the goal is CT, pedagogy, or tool focused
Main topic and conceptualization of CT	The main topic of the PD and the words used to define CT
PD implementation	
Duration of the PD	The amount of time spent with participants
Program(s) used in PD	Any programs and/or tools (if any)
Curriculum used in PD	The curriculum(s) (if any) used in the PD
Design to support CT integration	
CT integration	Whether the PD provided opportunities for CT integration
PD design features	Design features of PD projects to support CT integration (e.g., time and mentorship)
PD learning experiences	Professional learning experiences that are elicited through the design features (e.g., reflection and teaching practices)
Implemented and/or recommended	Whether the design feature or learning experiences are implemented and/or recommended
Outcome variable	
Evaluation type	Specific methods of evaluation(s) used
Level of evaluation	Reaction, attitude, perception, content knowledge, skill and application, and student impact
Additional information	Any other information regarding the PD

extracted duration, programs/tools, curriculum, and whether the PD devoted time and support to engage participants with designing or discussing instructional activities that employed elements of non-CT or non-CS general subjects such as math or science (i.e., CT integration). Specifically for the PDs that involve CT integration, we coded the design features and learning experiences to understand how researchers have designed the PD to support non-CS teachers in integrating CT into their classrooms.

We analyzed the evaluation approach(es) each PD had adopted based on the four levels of training evaluation in the Kirkpatrick Model (Kirkpatrick, 1996): (1) participants' reactions to the training; (2) acquisition of knowledge, skills, attitude, etc.; (3) application of skills at job; and (4) target outcomes occurring because of the training. We further elaborated on these four levels based on a conceptual framework for studying the effects of PD on teachers and students (Desimone, 2009). Using this framework, teachers' knowledge, skills, and attitudes, and perceptions change as a function of the PD (corresponding to Level 2 in the Kirkpatrick model). These changes are then reflected as changes in instruction (corresponding to Level 3 in the Kirkpatrick model) and then impact students learning (corresponding to Level 4 in the Kirkpatrick model). The outcomes of evaluation are categorized into six levels: (1) reaction, teachers' reaction to PD; (2) attitude, change in attitudes towards CT (e.g., the importance of CT, willingness of adoption); (3) perception, change in perceptions or willingness (e.g., self-efficacy, teachers' willingness of adopting CT); (4) content knowledge, understandings of CT; (5) skill and application, the ability to apply CT in real-world problem solving or design CT-integrated curriculums; and (6) student impact, change in student learning outcomes.

Results

What has been done? An overview

Overall, 76 studies were included in our analysis after the screening process (see Appendix). All of the studies were published in peer-reviewed venues, with most appearing in published conference proceedings ($n=46$). The most frequent source was *IEEE Frontiers in Education Conference* and *ACM Technical Symposium on Computer Science Education*. The publication dates ranged from 2007 to 2023 with the number of publications generally increasing every year (Fig. 2). As a note, the search was conducted in June 2023. Therefore, potential articles could have been missed in our search due to (a) the global pandemic; (b) the unfinished publication process; and (c) the lag in accessing articles from the bibliography databases. The PDs reported experiences in the US, Europe, Asia, and South America and covered a wide range of teachers (e.g., CS/IT teachers, math and science teachers, and teachers who work with vulnerable populations).

CT conceptualization and PD goals

CT conceptualization

Because we decided to use the author-developed definitions for CT, data collection captured a diverse collection of operationalizations of CT across settings and contexts. Based on the coding of the author-identified topics and

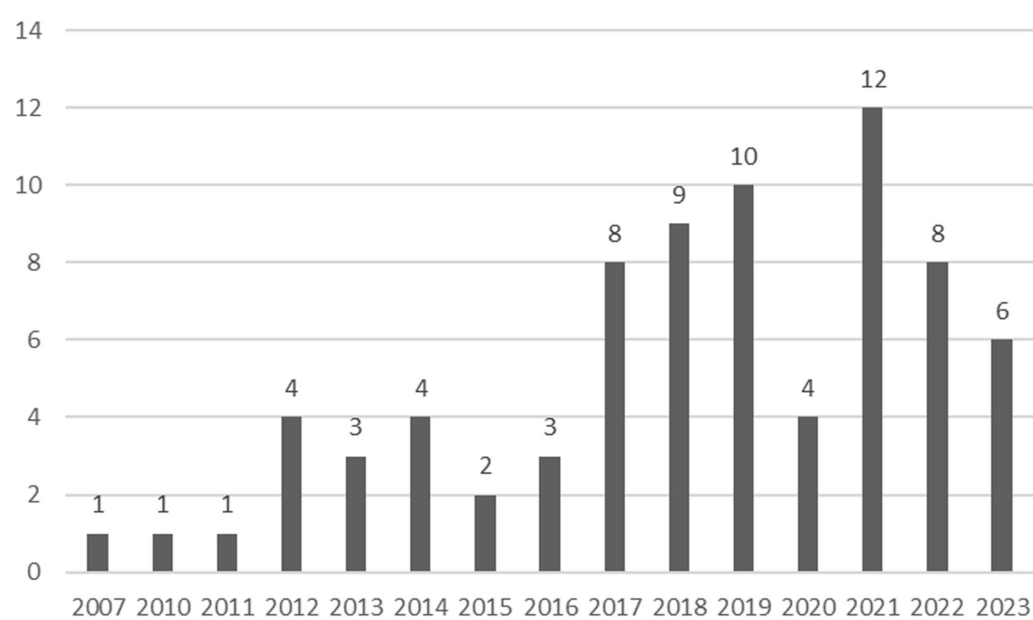

Fig. 2 Publications by year

conceptualizations, we found a wide spectrum of definitions for CT and related concepts in the literature. Table 4 illustrates several different ways these studies described and defined CT. These conceptualizations were identified as CT by each study's authors, but there was considerable variation in the interpretation of this umbrella term due to the differences in contexts and theoretical foundations (Lodi & Martini, 2021). We agree with Lodi and Martini that Papert's view of CT differs from Wing's in that Papert sees the goal of CT as being able to engage with interdisciplinary tools that allow access to big ideas—tools to think with. In contrast, for Wing, the goals of CT are focused on developing the skills and dispositions to solve a variety of problems by programming.

To this end, we arranged the elements on a spectrum that ranges from operationalizations that we determined to be more focused on the goal of learning CT as engaging with an "interdisciplinary tool for learning" (i.e., Papert's constructionism view) on one end to developing "computational competencies" with which to solve CS problems (i.e., Wing's CS-centric view) on the other end. We placed the operationalizations with a stronger emphasis on computing tool usage on the right side of the spectrum while the operationalizations with a heavier emphasis on thinking abilities on the left side of the spectrum (Fig. 3). As consumers of this research, we do not place more value on one end than the other of the continuum or attempt to claim that this is the definitive order of these conceptualizations. Instead, we offer it as a visual tool for organizing different conceptualizations of CT to navigate the complex landscape of existing CT PDs.

PD goals

Through our analysis, three foci emerged among the PDs: CT knowledge focus, pedagogy focus, and tool focus. We defined (a) a CT-focused PD as having explicit components that address the development of CT knowledge or skills; (b) a pedagogy-focused PD as introducing

Table 4 Examples of CT-related terms and conceptualizations

CT-related terms	Conceptualization
Computational thinking (CT)	Logically organizing information and data, analyzing data, formulating problems, representing data and problems through abstractions, creating algorithms, analyzing possible solutions, and generalizing or transferring problem-solving process (ISTE & CSTA, 2011)
Computational thinking (CT)	Problem decomposition, pattern recognition, abstraction, and algorithmic thinking (Dong et al., 2019)
Computational thinking (CT)	CT concepts, CT practices, and CT perspectives (Brennan & Resnick, 2012)
Computer science principles (CSP)	Big ideas of computer science such as abstraction, algorithms, programming, Internet, and how computing and technology can impact the world (College Board, n.d.)
CS in STEM	CS concepts, modeling and simulation, and the study of complex adaptive systems (Lee et al., 2017)
Beauty and joy in computing (BJC)	The programming language/environment, growth mindset, pair programming, computing in the news (Garcia et al., 2015)
Computing concepts	Program structures, objects, methods/functions, conditional branches, variables, loops (e.g., Liu et al., 2015)
CS unplugged	Solving problems that use cards, string, crayons, and lots of running arounds (CS unplugged, n.d.)
Robotics	Designing, programming, and building robotic systems (e.g., Hamner et al., 2016)

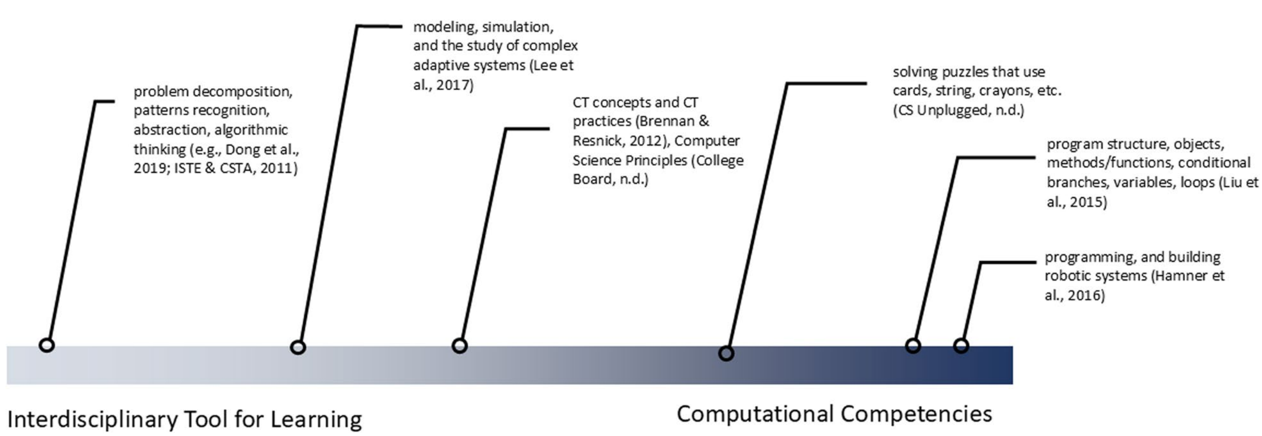

Fig. 3 Spectrum of CT operationalization

CT-related pedagogies (e.g., teaching strategies, CT-integration cases, CT curriculums) to teachers; and (c) a tool-focused PD as preparing participants for using one or more specific tool/skills (e.g., programming language, robots). These foci are distinctly different from each other and have significant implications on the design of the PD, but they are not mutually exclusive. Thus, one PD could be categorized into multiple foci.

Overall, the goals of the PDs focused on improving teachers' competence with CT-related concepts, tools, and pedagogies. Figure 4 shows the distribution of various PD foci. Forty-five of the PDs (59.21% out of 76 PDs) had goals explicitly focused on CT. Forty of those 45 PDs had both CT and pedagogical foci (88.88%). Although we found all 76 studies to be related to CT based on our screening process, 31 studies of those 76 PDs (40.79%) did not explicitly mention CT in their PD goals. Instead, CT knowledge and practices were implied through the PD activities. For example, one PD devoted time to training teachers to identify students' skills in CT and engineering design (Hamner et al., 2016). Another PD had an activity in which teachers designed a flowchart for a subtraction algorithm (Borowczak & Burrows, 2019). The PDs that did not explicitly list CT as part of their goals were either too specific to tools or too generic in their goal statement (e.g., "help teachers improve coding ability through Scratch" in Lazarinis et al., 2019, or "provide teachers with knowledge about digital basic education" in Tengler et al., 2021). Comparing CT conceptualizations, PD studies that did not mention CT explicitly tend to focus more on the computational competencies perspective for CT while such perspectives were not present in the studies that mentioned CT explicitly. Seven PDs explicitly operationalized "preparing teachers to teach CT" as teaching programming concepts to teachers. None of the seven mentioned CT explicitly in their goal statement. We did not exclude them from the dataset because (a) these studies explicitly claimed that they were teaching CT and (b) we aimed to demonstrate how CT PDs may be conceptualized and implemented differently from each other.

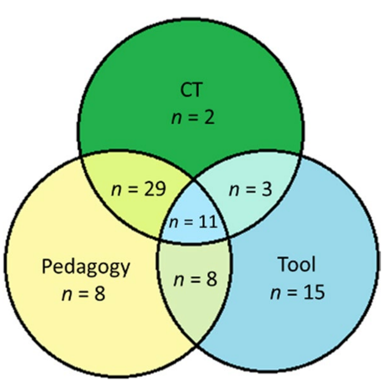

Fig. 4 PD goal foci

Fifty-six of the 76 PD studies reported a pedagogical focus (73.68%). These PDs focused on teachers' competence in using CT-related tools or pedagogies for designing instruction. However, most of the pedagogy-focused PDs included an additional focus such as CT or tool usage ($n=48$). For example, Ketelhut et al. (2020) stated that the goal of their PD was to "understand the nature of CT, the importance of integrating it into science education, and the recommendations for CT integration in elementary science classrooms". This goal reflects the foci of both CT and CT-related pedagogy. In contrast, there were a small number of pedagogy-only PDs ($n=8$). For example, Ozturk et al. (2018) stated that their PD aimed to support elementary teachers who lacked knowledge of computer science (CS), and project-based learning (PBL) to develop the skills and understandings to integrate CS through PBL within their curriculum. This goal does not emphasize CT per se and does not focus on any specific computing tools. Regarding the CT conceptualizations, all these pedagogically focused PDs viewed CT as an interdisciplinary tool for learning more than computational competencies. The specific operationalization of CT within each project varied. For example, some drew on the CT definition proposed by ISTE and CSTA (2011), while others used the CT concepts, practices, and perspectives by Brennan and Resnick, (2012) or the framework for CT for mathematics and science integration by Weintrop et al. (2016).

A substantial number of the PD studies reported a tool-only focus ($n=15$, 19.73%). In those PDs, a specific tool, such as a software program (e.g., ALICE in Liu et al., 2012) or a hardware tool, like a robot, was the main focus of the training. For example, Kay and colleagues (2014) described a workshop focused on building and using LEGO Mindstorm Robots. Their stated goals included increasing teacher confidence and using the robot in classrooms or after-school clubs. Perhaps not surprisingly, 10 out of these 15 PD studies operationalized CT as specific programming skills. While CT typically includes competencies beyond the knowledge of specific computing tools (Brennan & Resnick, 2012), in practice, computing tools—such as programming languages, coding challenges, and robots—are still a primary way to engage with CT and help novices contextualize the content knowledge.

We found 11 studies that included all three foci (i.e., CT, pedagogy, and tool). For example, one workshop led by von Wangenheim and colleagues (2017) had three sessions: one about computing knowledge where teachers assembled a robot, a second one focused on preparing teachers to run lessons by providing instructional material, and a third session where teachers learned and practiced installing the software and hardware necessary. The other study we found with all three foci was a 3-year-long professional development (Buss & Gamboa, 2017). Teachers first spent several weeks in an online course learning how to program and how CT skills fit into programming. Then, sessions were held to discuss how teachers could teach robotics and CT skills in their classes and in after-school clubs. There are more such examples in recent years (e.g., Biddy et al., 2021; Jocius et al., 2021b; Kelter et al., 2021; Kong et al., 2023; Rich et al., 2021; Simmonds et al., 2021), which indicates that the PD goals have been evolving.

Implementation of CT PD

We analyzed the implementation of CT PDs from three aspects: (1) sample size, (2) duration of PDs, (3) curricula and programs used. First, the sample size varied widely. Some studies reported massive implementations of CT PDs involving from 100 to more than 1000 participants (e.g., Kay et al., 2014; Lazarinis et al., 2019; Simmonds et al., 2021). These were typically either regional implementations or in the form of MOOCs (Massive Open Online Courses). There were also studies that had fewer than 10 participants (e.g., Biddy et al., 2021; Liu et al., 2011; Tsouccas & Meletiou-Mavrotheris, 2017). These smaller PDs often reported lower than anticipated enrollment of the PD, attrition, or funding limitations. More than half of the PD studies reported in this literature review involved fewer than 30 people. Across the studies reported here, participants in the PDs were typically teachers interested in technology (Lamprou & Repenning, 2018), thus, they may be early adopters of CT.

The PD efforts in this review lasted from 1 day (e.g., Bower et al., 2017) to multi-year programs (e.g., Buss & Gamboa, 2017; Hug et al., 2018). A majority of the PDs were called “workshops” by the authors. The term “workshop” was never defined, but based on the descriptions of the PDs, we took it to mean an intensive seminar-like training for teachers. Most of these workshops were held on consecutive days during school vacations (e.g., Ahamed et al., 2010); a few were held after school during the week (e.g., Simmonds et al., 2019). For example, Dong et al., (2019) implemented a 5-day workshop on CT and CT-related pedagogies. Although CT-integrated STEM learning examples were shared at the workshop, teachers seemed to be more enthusiastic about demonstrating their understanding of CT in the context of coding rather than in their own disciplines when they were creating CT lesson plans. Only seven PDs (9.21%) supported teachers through the entire school year or even multiple years, and many PDs lasted 5 days or less ($n=34$, 44.74%). There was one semester-long course (Joshi et al., 2019), and two MOOC courses that were available to teachers to complete at their own pace (Kay & McKlin, 2014; Lazarinis et al., 2019). Kong et al., (2023) engaged teachers in a 9-month-long continuous PD. A few PDs provided ongoing support by using designs that incorporated both online courses and face-to-face sessions (e.g., Hamlen et al., 2018). In another PD (Viera & Magana, 2013), teachers participated in a 3-day workshop but then were given 2 weeks to write and implement a lesson based on what they learned during the workshop.

Our analysis of CT PDs highlighted that a variety of CT tools were used and that PDs addressed many different curricula. There was notable variation in the programming, robotics, or CT tools used in the PDs, varying from free online programming tools such as Scratch (Haden et al., 2016), non-computer-based tools such as “CS unplugged” (Curzon et al., 2014), to robotics kits such as Lego NXT (Kay & McKlin, 2014). While 28 PDs (36.84%) focused on teaching only one tool, there were PDs which taught more than one tool during a CT PD (e.g., Kong et al., 2023; von Wangenheim et al., 2017). There were a few studies that included more than three programs or tools taught during the PD (e.g., Biddy et al., 2021; Bower et al., 2017; Byrne et al., 2015).

CT integration

Through our analysis of the PDs designed to support teachers’ CT integration ($n=38$), common design features and professional learning experiences emerged. “Features” were aspects of the PD designed to support teachers’ professional learning “experiences” (Table 5). We listed the design features (i.e., time, mentorship, community support, and pedagogical supports) and the experiences (i.e., feedback, reflection, and teaching practices) within the second column. Additionally, we provided the studies from our literature review that implemented these aspects as well as studies that recommended them.

First, some specific design features were implemented and reported to be effective while some were recommended by the authors based on their implementation (see examples in Table 5). Consistent with the “consensus model” for professional development (Brinkerhoff, 2006; Darling-Hammond et al., 2017; Desimone, 2009; Hill, 2004; Roth et al., 2017), studies that involved prolonged engagement with teachers reported positive outcomes in the form of positive attitudes and changes to teaching practices (e.g., Hestness et al., 2018; Ketelhut et al., 2020). However, as previously mentioned, many of the reported CT PDs were designed as short, one-shot programs, which does not fit within the “consensus model” for effective PD. Personalized mentoring for scaffolding teachers’ experiences was another design feature found

Table 5 Identified elements of supporting CT integration

		Description	Implemented	Recommended
Features	Time	Time is needed for substantial change	Ketelhut et al. (2020), Lehmkuhl-Dakhwe (2018), Feng et al. (2021), Peters-Burton et al. (2022), Peters-Burton et al. (2023), Pollock et al. (2017), Reding and Dorn (2017)	Bort and Brylow (2013), Feng et al. (2021), Jocius et al. (2021b), Ozturk et al. (2018), Simmonds et al. (2019)
	Mentorship	Participants need various forms of personalized mentorship (e.g., tutor, coach, graduate assistant, consultant)	Choate et al. (2018), Feng et al. (2021), Ketelhut et al. (2020), Liu et al. (2011), Ouyang et al. (2018), Ozturk et al. (2018), Peters-Burton et al. (2022), Sherwood et al. (2021), Simmonds et al. (2019)	Liu et al. (2015), Simmonds et al. (2019), Vieira and Magana (2013)
	Community support	Establishing a community of support can help create a sustainable PD beyond the professional learning experience	Coenraad et al. (2022), Jocius et al. (2021a), Sherwood et al. (2021), Simmonds et al. (2021), Skuratowicz et al. (2021)	Ghani et al. (2022), Rachmatullah and Wiebe (2023)
	Pedagogical support	Participants need CT-related pedagogical supports for integration purposes	Bort and Brylow (2013), Espinal et al. (2021), Lehmkuhl-Dakhwe, (2018), Ozturk et al. (2018), Peters-Burton et al. (2023), Rachmatullah and Wiebe (2023)	Coenraad et al. (2022), Ghani et al. 2022), Jocius et al. 2021b), Mumcu et al. 2023), Ozturk et al. 2018), Pokorny and White (2012), Simmonds et al. (2019)
Experiences	Feedback experiences	Feedback on lessons, level of understanding, etc	Ahamed et al. (2010), Choate et al. (2018), Dong et al. (2019), Jocius et al. (2021a), Ouyang et al. (2018), Vieira and Magana (2013)	Bort and Brylow (2013)
	Reflection experiences	Opportunities for teachers to reflect on their experience, knowledge, and practice either verbally or in written forms	Biddy et al. (2021), Bort and Brylow (2013), Choate et al. (2018), Dong et al. (2019), Ketelhut et al. (2020), Lehmkuhl-Dakhwe (2018)	
	Hands-on practices	Teacher participants need various forms of hands-on practice (e.g., tool, planning curriculum, co-design, pilot run of lessons, classroom implementation)	Ahamed et al. (2010), Biddy et al. (2021), Bort and Brylow (2013), Choate et al. (2018), Coenraad et al. (2022), Dong et al. (2019), Feng et al. (2021), Jocius et al. (2021a, b), Kelter et al. (2021), Ketelhut et al. (2020), Kite and Park (2022), Lehmkuhl-Dakhwe (2018), Mumcu et al. (2023), Ozturk et al. (2018), Peters-Burton et al. (2022), Peters-Burton et al. (2023), Pokorny and White (2012), Reding and Dorn, 2017, Sherwood et al. (2021), Simmonds et al. (2019), Vieira and Magana (2013)	Biddy et al. (2021), Jocius et al. (2021b), Liu et al. (2011), Ozturk et al. (2018), Pokorny and White, (2012), Pollock et al. (2017), Reding and Dorn, (2017)

to be effective. Such personalization took various forms (e.g., tutoring in Liu et al., 2015, coaching in Hamner et al., 2016, consultation in von Wangenheim et al., 2017). Researchers also reported that teachers needed both technical supports and pedagogical supports to actively integrate CT into their classrooms (Dong et al., 2019; Pollock et al., 2017). In addition, it was helpful to develop a community of practice that included various stakeholders including teachers of different proficiency levels, students, mentors, and administrators (Choate et al., 2018; Hestness et al., 2018).

Second, based on our review of the literature, we identified specific learning experiences that were implemented and recommended for teachers. For example, many studies addressed the role of feedback and reflection in supporting teachers' CT integration (e.g., Lehmkul-Dakhwe, 2018; Ouyang et al., 2018). According to these studies, CT integration involved not only supporting teachers' conceptual understanding, but also providing them with practice. Thus, actively planning and teaching CT-integrated classes was an important component of effective PD efforts. The practices ranged from experience working with the tool (e.g., Pokorny & White, 2012), to curriculum planning (e.g., Liu et al., 2015), and to classroom implementation (e.g., Tsouccas & Meletiou-Mavrotheris, 2017). Hickmott and Prieto-Rodrigues (2018) reported that teachers enjoyed sessions that had step-by-step exercises because of their limited technology proficiency, suggesting that practice may need to be at a basic level for some teachers.

PD assessments and evaluations

We coded how researchers in each study evaluated the outcomes of their efforts using the Kirkpatrick model of training evaluation (Kirkpatrick, 1996) along with the conceptual framework for the effects of professional development on teachers and students (Desimone, 2009). We coded the types of evaluation methods (e.g., survey, interview, observation, review of artifacts, scale, and performance-based assessment).

First, 28 PD studies (36.84%) measured participants' reactions. These reactions were primarily measured through satisfaction surveys and interviews evaluating the extent to which the participants liked the experience. These same studies were occasionally measured by facilitators' observation of the participants as they engaged in PD. Typical questions included perceived enjoyment (e.g., Byrne et al., 2015), perceived usefulness (e.g., Morais & Bachrach, 2019), perceived challenges (e.g., Hickmott & Prieto-Rodriguez, 2018), etc.

Second, 15 PD studies (19.74%) measured teachers' attitudinal change. Like participants' reactions, attitudinal outcomes were also measured by surveys and interviews. The attitudes included teachers' willingness to adopt CT (e.g., Haden et al., 2016), motivation to implement CT in their classrooms (e.g., von Wangenheim et al., 2017), and view on how students are going to benefit from CT as well as the future career opportunities (e.g., Mouza et al., 2016; Simmonds et al., 2019).

Third, 39 PD studies (51.32%) measured teachers' perceptions of CT learning. These efforts also relied on surveys and interviews. For example, many studies measured whether teachers were comfortable with the content knowledge (e.g., Hoic-Bozic, 2019; Liu et al., 2015). In contrast, some other studies measured teachers' perception of whether they are comfortable or familiar with teaching CT or teaching computing technologies (e.g., Hamlen et al., 2018; Kong et al., 2020). Some studies that measure teachers' self-efficacy on either technology or teaching, asked similar questions (e.g., Borowczak et al., 2019).

Fourth, in addition to perceptions of competence in CT and teaching, some studies ($n=27$, 35.53%) also assessed the professional development outcomes by measuring changes in teachers' CT understanding (e.g., Blum et al., 2007; Buss & Gamboa, 2017; Ketelhut et al., 2020). CT understandings were often measured through scales, open-ended questions, or interviews. For example, Buss and Gamboa (2017) asked teachers to endorse a few pre-established CT-related statements based on the attitude survey developed by Yadav et al. (2014). Alternatively, Ketelhut et al. (2020) asked teachers to complete a written reflection at the end of each day's PD activity or discuss how the PD had influenced their understanding of what CT is. In contrast to conceptual changes, some studies measured teachers' CT knowledge through test items (e.g., Liu et al., 2015). These test items are typical CT/programming problems that evaluate teachers' understanding of concepts.

Fifth, some studies moved beyond teacher surveys of perceptions and knowledge to focus on changes in skills and applications. Thirty-eight studies (50%) measured teachers' abilities to apply learning outcomes to solve real-world problems. For example, Kong et al. (2020) developed a performance-based measure which used question stems based on real-world scenarios and asked the teachers to apply what they had learned about CT practices to their problem solutions. In addition, many studies required teachers to implement CT teaching practices. For example, Simmonds et al. (2019) asked teachers to design a CT-integrated curriculum at the end of the PD. Because this end-of-PD assessment may not be implemented by the teacher, some studies required teachers to implement the design so their practice could be evaluated either by the research team or through the teachers' own reflection. For example, some studies asked teachers to self-report their teaching practices or behavioral changes after the PD through follow-up surveys (e.g., Hoic-Bozirc et al., 2019; Pokorny et al., 2012; Pollock et al., 2017). Some other studies chose to interview and observe teachers in practice to identify if the change has indeed happened (e.g., Hestness et al., 2018; Ketulhut et al., 2020).

Finally, a small number of studies ($n=11$, 14.47%) measured the impact of teachers' participation in the PD on students' learning. This type of measurement was done through surveys and scales and could measure a wide range of impacts on students. For instance, some studies measured student impact by considering class enrollment (e.g., Joshi et al., 2018; Kay et al., 2014). Other studies measured student impact by measuring social and emotional learning levels (e.g., motivation in Neutens et al., 2018; learner autonomy in Ozturk et al., 2018). Some other studies measured student impact at knowledge level through test items measuring students' understanding of the content (e.g., Ouyang et al., 2018).

Discussion

CT objectives of PDs

Addressing our first research question, concerning the learning goals and topics of the existing CT PD models and programs, our findings suggest that studies mentioning CT in their goals often do so in a relatively broad manner. Examples of broad CT PD goal statements based off an amalgamation of our dataset would be to "help prepare teachers to teach computational thinking" or to "teach basic computational thinking skills to elementary teachers." Additionally, many PDs that include CT do so without explicitly mentioning the term "computational thinking" in their goal statement. However, the activities of PDs suggest they included CT as an important component of the PD. Examples of such goal statements would be to "help middle school teachers improve their ability to code using Scratch" or "provide teachers new pedagogical techniques in computer science education." While using vague language may make the PD more appealing to a wider audience or may convey the inclusion of an assortment of ideas, such goals are hard to develop activities to meet and are even harder to measure. This can cause activities or instructional material to appear unrelated. Two examples of precise and measurable goal statements in our analysis were from Dong et al. (2019) and Ketelhut et al. (2020). Dong et al. (2019) broke their learning objectives for CT down into four aspects and identified the connections between each facet with the subject content knowledge (e.g., abstraction in science can be taught through simplifying "models of Newtonian mechanics or solar systems" and algorithms in math can be taught through listing "steps for doing long division or integral calculus"). Ketelhut et al. (2020) designed the PD objectives based on teachers' professional growth in four domains (i.e., external domain, personal domain, domain of practice, and domain of consequence) per Clarke and Hollingsworth (2002).

One of the challenges in articulating the CT-related objectives is that CT can be operationalized in different ways. We identified a wide spectrum of conceptualizations of the construct of CT from the RQ1 findings (Fig. 4). While CT was formalized over a decade ago (e.g., Wing, 2006), researchers and practitioners do not seem to operate from one unified definition of CT. We posit that this lack of consensus is, at least in part, due to the multi-faceted nature of CT. It can be considered as a set of competencies (e.g., problem decomposition, systemic thinking), a set of dispositions (e.g., perseverance), and a set of skills (e.g., knowing how to use programming tools to express a solution derived from engaging in CT) that can be seen in diverse contexts. For instance, CT is involved when students learn the basic computing concepts and programming languages in a CS class as well as when they are engaged in embodied activities to solve math problems; and when students program robots to simulate science phenomena and make predictions (Wang et al., 2022).

Although there exist various realizations, many PDs continue to regard knowledge and skill acquisition as the endpoint of learning CT (i.e., the cognitive framing mentioned by Kafai & Proctor, 2022, and Wing's definition according to Lodi & Martini, 2021). There are very few PDs reviewed in this study that incorporated multiple perspectives in CT such as understanding of learning as identity formation and how the societal values and histories are embedded in a world with computing technologies (Kafai & Proctor, 2022). Through mapping the conceptual spectrum of CT operationalization (Fig. 3), we intend to demonstrate that CT is not just about one pre-defined concept. It is equally important to support teachers' understanding of the multi-faceted nature of CT and explore the contexts where students can develop CT in their local settings instead of only relying on premade curricula or packages.

CT PD implementation and CT integration

Addressing what CT PDs look like in practice (RQ2) and how PDs support CT integration (RQ3), we found many PDs were conducted in the workshop-style and generally focused on introducing the concept of CT to teachers. Twenty-five of 76 in this study had a brief, intensive informational format (e.g., lecture) and took place over a few consecutive days. While many teachers know little about CT, short workshops are generally not the best PD format (Ketelhut et al., 2020; Lehmkul-Dakhwe et al., 2018; Pollock et al., 2017; Reding & Dorn, 2017) and have been discouraged by previous literature (Bort & Brylow, 2013; Ouyang et al., 2018; Ozturk et al., 2018; Simmonds et al., 2019). If the goal of the PD is to support teachers' CT integration into classrooms, designers of PDs need to design learning opportunities that allow sufficient time for this change to happen. Research has generally shown that engagement over time is more effective for supporting changes in teachers' practice (e.g., Darling-Hammond et al., 2017; Desimone, 2009).

Multiple studies identified that a lack of technology experiences (e.g., coding skills) may hinder teachers' self-efficacy for learning and teaching CT. This can be particularly intimidating for novice teachers because CT is a set of competencies that are often introduced through unfamiliar technologies (Hickmott & Prieto-Rodriguez, 2018), and, past research has suggested that teachers' beliefs can limit their adoption of CT. Further, teachers may not see relevance in integrating CT into STEM disciplines. Based on our synthesis, adequate support from both mentors and a community of practice is crucial to a successful PD (Choate et al., 2018; Ketelhut et al., 2020; Liu et al., 2011; Ouyang et al., 2018; Ozturk et al., 2018; Simmonds et al., 2019; Vieira & Magana, 2013). Teachers are often new to CT (Yadav et al., 2014), therefore, appropriate scaffolds are needed to support them in developing the confidence and skills to successfully design CT-integrated STEM lessons on their own. The scaffolds can include mentorship from the PD team (Hoic-Bozic et al., 2019; Lehmkuhl-Dakhwe, 2018; Ouyang et al., 2018; Ozturk et al., 2018) or the development of a community of practice (Hestness et al., 2018; Hickmott & Prieto-Rodriguez, 2018; Simmonds et al., 2019) that can persist beyond the life of the PD. The community should include teachers with various levels of competencies, subjects, and backgrounds. Further, Simmonds et al. (2019) explicitly highlighted how getting buy-in from district or building administrators is key for building a long-term CT program. Through mentorship and community-building, teachers can receive formative feedback, which provides scaffolds to teachers' CT knowledge as well as CT-related pedagogical content knowledge (PCK) along the way.

Finally, PDs can provide enough pedagogical support to support teachers in integrating CT into their classrooms. Teachers often lack both CT content knowledge and PCK to integrate CT into regular classrooms. Therefore, support including reflection and classroom practice experiences is vital for teachers to develop PCK (Liu, 2023). Past studies have shown that iterative implementation experience that involves coaching, reflection, and practice helps teachers grow longitudinally (Bort & Brylow, 2013; Choate et al., 2018; Dong et al., 2019; Ketelhut et al., 2020; Lehmkuhl-Dakhwe, 2018).

Methods of assessment and evaluation

In response to research question 4, we identified a variety of evaluation methods that have been implemented in PDs. They range from teachers' reactions to PDs to impacts on students. Each study reviewed included one or more evaluations of outcomes, however, none provided a comprehensive assessment of the longitudinal impacts of the PD on teachers or their students. Most studies relied on only one or two measures. For example, the researchers may have only measured teachers' perceptions of their content knowledge. The impact of a PD contains a series of causal links (e.g., from increased teacher knowledge to change in instruction, from change in instruction to improved student learning, see Desimone, 2009 and Manizade et al., 2023) and must go through various "barriers" such as structural/administrative issues, acceptance issues, and implementation issues (McChesney & Aldridge, 2021). Thus, it can be a challenge to measure the impact of PD on teaching and learning. However, we argue that the broad goal statements provided by the researchers also obfuscated the ability to measure impact. When researchers are not clear about the learning objectives of PDs, it is impossible to design valid and aligned assessments to evaluate the impact (Mohr & Shelton, 2017). Further complicating the issue of measurement, even measuring only the impact on teacher knowledge would ideally need to measure both content knowledge and pedagogical content knowledge for teachers (e.g., Shulman, 1987), a difficult benchmark to meet in instrument design.

At this point in the evolution of CT, measuring learning from CT is also complicated by the field's lack of understanding of what a teacher needs to know for teaching CT. And, assessment development for CT is extremely difficult given the complex landscape of CT conceptualizations for K-12 and the diverse contexts where CT learning can happen. Therefore, the findings of this literature review aim to help future PD designers and researchers see how CT professional learning has been defined and measured so that they can configure specific assessments tailored to their goals, contexts, and endpoints.

Limitations

We acknowledge several limitations in our review. Many of these limitations come from the criteria we set in the screening stage. First, we limited our population to K-12 in-service teachers to provide a focused scope. We acknowledge that considerable work is being undertaken in the pre-service teacher space, but that was outside the scope of our review. Second, our pre-established review protocol may not have captured all interchangeable synonyms for our keywords (e.g., professional learning). Third, we relied on four bibliography databases to collect data. We acknowledge that the industry (e.g., Google) and professional organizations (e.g., ISTE) have initiated and sponsored many PDs in recent years. Some science curricula used in schools are now starting to include CT elements, and their associated PD programs may also address CT integration. However, some of the initiatives are still ongoing, and scholarly publications are not necessarily a goal for those efforts. Thus, their valuable experiences are not included. Lastly, we reviewed only articles in which the authors self-identified their work as focusing on the topic of CT, and our coding of the studies was based solely on the manuscripts and their self-identified themes. More than two-thirds of coded studies in this review are conference proceedings, which do not include rich theoretical discussion or in-depth implementation details. Therefore, our analysis necessarily relied on each study's author-reported themes to code the studies based on our best judgment from the texts.

Future directions

This systematic review highlights several promising avenues for future research and practice in CT PD for K-12 teachers. First, future reviews could expand their scope to include pre-service teachers' experiences with CT integration, research–practitioner partnerships initiated by the industry, or science-curricula-related CT PD. These perspectives could offer valuable insights into teacher professional learning and the role of various stakeholders in supporting CT integration. Second, future professional development (PD) programs should explore a clearer, expanded conceptualization of CT. This expanded view should emphasize how CT can be integrated into diverse STEM classrooms to promote equity and inclusion. Third, there is a pressing need for research in teacher PD that focuses on building sustainable, prolonged, and engaging learning experiences that foster communities of practice among teachers. This requires PD to extend beyond one-time workshops to create ongoing support structures that allow for continuous learning, collaboration, and reflection on CT-integration practices. Fourth, future research should also investigate what constitutes "good CT instruction" and how it can be effectively measured. This includes developing robust assessment tools that can capture teachers' professional growth longitudinally and its impact on student performance. Such assessments should consider not only content knowledge, but also pedagogical skills and the ability to adapt CT concepts to various contexts.

Conclusion

In this synthesis, we identified empirical studies on CT PDs for K-12 teachers. We systematically reviewed 76 articles that reported on PDs to present what has been done in the field to support K-12 teachers' CT development and CT integration. The results highlighted the experiences and challenges of current PDs. Based on the findings, we suggested various issues to be addressed to improve CT PD projects for K-12 teachers with a focus on integrating CT into STEM education. The promotion of CT started more than a decade ago, but the research on sustainable and equitable classroom implementation has just started. Only with the help from classroom teachers, we can eventually weave CT into students' learning in the near future.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s40594-024-00510-6.

Additional file 1.

Acknowledgements

We would like to thank the editors and reviewers at the *International Journal of STEM Education* for their thoughtful feedback and recommendations.

Author contributions

ZL planned the study. Data collection, analysis, synthesis, and results write-up were done by ZL, ZG, ER, and CHO. The funding and organizational support was provided by CHO, SK, and RB. All authors contributed to the writing of the paper. The names are in order of the amount of contribution given. All authors read and approved the final manuscript.

Funding

This work is supported by the National Science Foundation under Grant No. 1934111. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Availability of data and materials

A brief version of the data used and analyzed in the current study is listed in the Appendix. The detailed version is available from the corresponding author on request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 11 March 2024 Accepted: 24 September 2024
Published online: 07 October 2024

References

Ahamed, S. I., Brylow, D., Ge, R., Madiraju, P., Merrill, S. J., Struble, C. A., & Early, J. P. (2010). Computational thinking for the sciences: A three day workshop for high school science teachers. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 42–46). https://doi.org/10.1145/1734263.1734277

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12. *ACM Inroads, 2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Biddy, Q., Chakarov, A. G., Bush, J., Elliott, C. H., Jacobs, J., Recker, M., Sumner, T., & Penuel, W. (2021). A professional development model to integrate computational thinking into middle school science through codesigned storylines. *Contemporary Issues in Technology and Teacher Education, 21*(1), 53–96.

Blum, L., & Cortina, T. J. (2007). CS4HS: An outreach program for high school CS teachers. *ACM SIGCSE Bulletin, 39*(1), 19–23. https://doi.org/10.1145/1227504.1227320

Borowczak, M., & Burrows, A. C. (2019). Ants go marching—Integrating computer science into teacher professional development with Netlogo. *Education Sciences, 9*(1), 66. https://doi.org/10.3390/educsci9010066

Bort, H., & Brylow, D. (2013). CS4Impact: Measuring computational thinking concepts present in CS4HS participant lesson plans. In: *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (pp. 427–432). https://doi.org/10.1145/2445196.2445323

Bower, M., Wood, L. N., Lai, J. W. M., Highfield, K., Veal, J., Howe, C., Lister, R., & Mason, R. (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

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In A. F. Ball & C. A. Tyson (Eds.), *Proceedings of the 2012 annual meeting of the American Educational Research Association* (pp. 1-25). American Educational Research Association.

Brinkerhoff, J. (2006). Effects of a long-duration, professional development academy on technology skills, computer self-efficacy, and technology integration beliefs and practices. *Journal of Research on Technology in Education, 39*(1), 22–43. https://doi.org/10.1080/15391523.2006.10782471

Buss, A., & Gamboa, R. (2017). Teacher transformations in developing computational thinking: Gaming and robotics use in after-school settings. In P. J. Rich & C. B. Hodges (Eds.), *Emerging research, practice, and policy on computational thinking* (pp. 189–203). Springer International Publishing. https://doi.org/10.1007/978-3-319-52691-1_12

Byrne, J. R., Fisher, L., & Tangney, B. (2015). Empowering teachers to teach CS—Exploring a social constructivist approach for CS CPD, using the Bridge21 model. *IEEE Frontiers in Education Conference (FIE), 2015*, 1–9. https://doi.org/10.1109/FIE.2015.7344030

Cabrera, L. (2019). Teacher preconceptions of computational thinking: A systematic literature review. *Journal of Technology and Teacher Education, 27*(3), 305–333.

Choate, B., Dubosarsky, M., & Chen, K. (2018). An innovative professional development model for teaching robotics to novice educators. *IEEE Frontiers in Education Conference (FIE), 2018*, 1–5. https://doi.org/10.1109/FIE.2018.8658771

Clarke, D., & Hollingsworth, H. (2002). Elaborating a model of teacher professional growth. *Teaching and Teacher Education, 18*(8), 947–967. https://doi.org/10.1016/S0742-051X(02)00053-7

Coenraad, M., Cabrera, L., Killen, H., Plane, J., & Ketelhut, D. J. (2022). Computational thinking integration in elementary teachers' science lesson plans. In A. Ottenbreit-Leftwich & A. Yadav (Eds.), *Computational thinking in PreK-5: empirical evidence for integration and future directions* (pp. 11–18). Association for Computing Machinery. https://doi.org/10.1145/3507951.3519283

College Board. (n.d.). *AP Computer Science Principles*. Retrieved September 30, 2024, from https://apstudents.collegeboard.org/courses/ap-computer-science-principles

CS Unplugged. (n.d.). *Computer Science without a Computer*. Retrieved September 30, 2024, from https://www.csunplugged.org/en/

Curzon, P. (2013). Cs4fn and computational thinking unplugged. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education* (pp. 47–50). https://doi.org/10.1145/2532748.2611263

Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014). Introducing teachers to computational thinking using unplugged storytelling. In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education* (pp. 89–92). https://doi.org/10.1145/2670757.2670767

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

Desimone, L. M. (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

Desimone, L. M., & Garet, M. S. (2015). Best practices in teachers' professional development in the United States. *Psychology, Society, & Education, 7*(3), 252. https://doi.org/10.25115/psye.v7i3.515

Dong, Y., Catete, V., Jocius, R., Lytle, N., Barnes, T., Albert, J., Joshi, D., Robinson, R., & Andrews, A. (2019). PRADA: A practical model for integrating computational thinking in K-12 education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 906–912). https://doi.org/10.1145/3287324.3287431

Espinal, A., Vieira, C., & Magana, A. J. (2021). Professional development in computational thinking for teachers in Colombia. *IEEE Frontiers in Education Conference (FIE), 2021*, 1–4. https://doi.org/10.1109/FIE49875.2021.9637310

Feng, S., & Yang, D. (2021). Teachers' perspective on implementing computational thinking in elementary classrooms. *IEEE Frontiers in Education Conference (FIE), 2021*, 1–5. https://doi.org/10.1109/FIE49875.2021.9637221

Fofang, J., Weintrop, D., Walton, M., Elby, A., & Walkoe, J. (2020). Mutually supportive mathematics and computational thinking in a fourth-grade classroom. In M. Gresalfi & I. S. Horn (Eds.), *14th International Conference of the Learning Sciences* (pp. 1389–1396). International Society of the Learning Sciences. https://doi.org/10.22318/icls2020.1389

Garcia, D., Harvey, B., & Barnes, T. (2015). The beauty and joy of computing. *ACM Inroads, 6*(4), 71–79. https://doi.org/10.1145/2835184

Ghani, A., Griffiths, D., Salha, S., Affouneh, S., Khalili, F., Khlaif, Z. N., & Burgos, D. (2022). Developing teaching practice in computational thinking in Palestine. *Frontiers in Psychology*. https://doi.org/10.3389/fpsyg.2022.870090

Gomoll, A., Hmelo-Silver, C. E., Šabanović, S., & Francisco, M. (2016). Dragons, ladybugs, and softballs: Girls' STEM engagement with human-centered robotics. *Journal of Science Education and Technology, 25*(6), 899–914. https://doi.org/10.1007/S10956-016-9647-Z

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189X12463051

Haden, P., Gasson, J., Wood, K., & Parsons, D. (2016). Can you learn to teach programming in two days? In *Proceedings of the Australasian Computer Science Week Multiconference* (pp. 1–7). https://doi.org/10.1145/2843043.2843063

Hamlen, K., Sridhar, N., Bievenue, L., Jackson, D. K., & Lalwani, A. (2018). Effects of teacher training in a computer science principles curriculum on teacher and student skills, Confidence, and Beliefs. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 741–746). https://doi.org/10.1145/3159450.3159496

Hamner, E., Cross, J., Zito, L., Bernstein, D., & Mutch-Jones, K. (2016). Training teachers to integrate engineering into non-technical middle school curriculum. *IEEE Frontiers in Education Conference (FIE), 2016*, 1–9. https://doi.org/10.1109/FIE.2016.7757528

Hestness, E., Jass Ketelhut, D., McGinnis, J. R., & Plane, J. (2018). Professional knowledge building within an elementary teacher professional development experience on computational thinking in science education. *Journal of Technology and Teacher Education, 26*(3), 411–435.

Hickmott, D., & Prieto-Rodriguez, E. (2018). To assess or not to assess: tensions negotiated in six years of teaching teachers about computational thinking. *Informatics in Education, 17*(2), 229–244. https://doi.org/10.15388/infedu.2018.12

Hill, H. C. (2004). Professional development standards and practices in elementary school mathematics. *The Elementary School Journal, 104*(3), 215–231. https://doi.org/10.1086/499750

Hoic-Bozic, N., Loncaric, D., & Dlab, M. H. (2019). Preparing primary junior grade teachers to teach computational teaching: Experiences from the GLAT project. *Mathematics and Informatics, 62*(5), 487–499.

Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education, 126*, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004

Hug, S., Eyerman, S., Cota, R., & Pontelli, E. (2018). Embedding K12 professional development through co-teaching experiences-sustaining computational thinking in interdisciplinary courses. In *2018 Research on Equity*

and Sustained Participation in Engineering, Computing, and Technology (RESPECT) (pp. 1–6). https://doi.org/10.1109/RESPECT.2018.8491708
Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education, 82*, 263–279. https://doi.org/10.1016/J.COMPEDU.2014.11.022
ISTE, & CSTA. (2011). *Operational definition of computational thinking for K-12 Education*. Retrieved September 30, 2024, https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf
Jocius, R., Joshi, D., Albert, J., Barnes, T., Robinson, R., Cateté, V., Dong, Y., Blanton, M., O'Byrne, I., & Andrews, A. (2021a). 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* (pp. 1198–1204). https://doi.org/10.1145/3408877.3432558
Jocius, R., O'Byrne, W. I., Albert, J., Joshi, D., Robinson, R., & Andrews, A. (2021). From professional development to classroom practice. *Educational Technology & Society, 24*(4), 166–179. https://doi.org/10.30191/ETS.202110_24(4).0013
Joshi, A., & Jain, A. (2018). Reflecting on the impact of a course on inclusive strategies for teaching computer science. *IEEE Frontiers in Education Conference (FIE), 2018*, 1–9. https://doi.org/10.1109/FIE.2018.8659237
Joshi, A., Jain, A., Covelli, E., Yeh, J. H., & Andersen, T. (2019). A sustainable model for high-school teacher preparation in computer science. In *Proceedings-Frontiers in Education Conference, FIE, 2019-Octob* (June). https://doi.org/10.1109/FIE43999.2019.9028638
Kafai, Y. B., & Proctor, C. (2022). A revaluation of computational thinking in K–12 education: Moving toward computational literacies. *Educational Researcher, 51*(2), 1–6. https://doi.org/10.3102/0013189X211057904
Kay, J. S., & McKlin, T. (2014). The challenges of using a MOOC to introduce "absolute beginners" to programming on specialized hardware. In *Proceedings of the First ACM Conference on Learning @ Scale Conference* (pp. 211–212). https://doi.org/10.1145/2556325.2567886
Kay, J. S., Moss, J. G., Engelman, S., & McKlin, T. (2014). Sneaking in through the back door: Introducing k-12 teachers to robot programming. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (pp. 499–504). https://doi.org/10.1145/2538862.2538972
Kelter, J., Peel, A., Bain, C., Anton, G., Dabholkar, S., Horn, M. S., & Wilensky, U. (2021). Constructionist co-design: A dual approach to curriculum and professional development. *British Journal of Educational Technology, 52*(3), 1043–1059. https://doi.org/10.1111/bjet.13084
Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, J. R. (2020). Teacher change following a professional development experience in integrating computational thinking into elementary science. *Journal of Science Education and Technology, 29*(1), 174–188. https://doi.org/10.1007/s10956-019-09798-4
Kirkpatrick, D. (1996). Revisiting Kirkpatrick's four-level model. *Training & Development, 1*, 54–57.
Kite, V., & Park, S. (2022). Preparing inservice science teachers to bring unplugged computational thinking to their students. *Teaching and Teacher Education*. https://doi.org/10.1016/j.tate.2022.103904
Kong, S. C., Lai, M., & Li, Y. (2023). Scaling up a teacher development programme for sustainable computational thinking education: TPACK surveys, concept tests and primary school visits. *Computers & Education*. https://doi.org/10.1016/j.compedu.2022.104707
Kong, S.-C., Lai, M., & Sun, D. (2020). Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education, 151*, 103872. https://doi.org/10.1016/j.compedu.2020.103872
Lamprou, A., & Repenning, A. (2018). Teaching how to teach computational thinking. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education* (pp. 69–74). https://doi.org/10.1145/3197091.3197120
Lane, W. B., Galanti, T. M., & Rozas, X. L. (2023). Teacher re-novicing on the path to integrating computational thinking in high school physics instruction. *Journal for STEM Education Research, 6*(2), 302–325. https://doi.org/10.1007/s41979-023-00100-1
Lazarinis, F., Karachristos, C. V., Stavropoulos, E. C., & Verykios, V. S. (2019). A blended learning course for playfully teaching programming concepts to school teachers. *Education and Information Technologies, 24*(2), 1237–1249. https://doi.org/10.1007/s10639-018-9823-2
Lee, I. A., Psaila Dombrowski, M., & Angel, E. (2017). Preparing STEM teachers to offer New Mexico Computer Science for All. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 363–368). https://doi.org/10.1145/3017680.3017719
Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2019). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology, 29*(1), 1–8. https://doi.org/10.1007/S10956-019-09803-W
Lehmkuhl-Dakhwe, K. V. (2018). An instructional framework, model lessons, and professional learning program for science standards-aligned computing in 4th-12th grade classrooms. *IEEE Frontiers in Education Conference (FIE), 2018*, 1–5. https://doi.org/10.1109/FIE.2018.8658580
Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology, 25*(6), 860–876. https://doi.org/10.1007/S10956-016-9628-2
Li, Y., Schoenfeld, A. H., DiSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). On computational thinking and STEM education. *Journal for STEM Education Research, 3*(2), 147–166. https://doi.org/10.1007/S41979-020-00044-W
Liu, J., Lin, C., Hasson, E. P., & Barnett, Z. D. (2012). Computer science learning made interactive—A one-week alice summer computing workshop for K-12 teachers. *Frontiers in Education Conference Proceedings, 2012*, 1–6. https://doi.org/10.1109/FIE.2012.6462505
Liu, J., Lin, C.-H., Hasson, E. P., & Barnett, Z. D. (2011). Introducing computer science to K-12 through a summer computing workshop for teachers. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (pp. 389–394). https://doi.org/10.1145/1953163.1953277
Liu, J., Wilson, J., Hemmenway, D., Xu, Y., & Lin, C. (2015). Oh SNAP! A one-week summer computing workshop for K-12 teachers. In *2015 10th International Conference on Computer Science & Education (ICCSE)* (pp. 663–668). https://doi.org/10.1109/ICCSE.2015.7250329
Liu, Z. L. (2023). Integrating computational thinking into K-12 education: Bridging the gap between theories and practices. *STEM Education Review*. https://doi.org/10.54844/stemer.2023.0467
Lodi, M., & Martini, S. (2021). Computational thinking, between Papert and Wing. *Science & Education, 30*(4), 883–908. https://doi.org/10.1007/S11191-021-00202-5
Love, T. S., Bartholomew, S. R., & Yauney, J. (2022). Examining changes in teachers' beliefs toward integrating computational thinking to teach literacy and math concepts in grades K-2. *Journal for STEM Education Research, 5*(3), 380–401. https://doi.org/10.1007/s41979-022-00077-3
Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61. https://doi.org/10.1016/J.CHB.2014.09.012
Manizade, A., Buchholtz, N., & Beswick, K. (2023). *The evolution of research on teaching mathematics: International perspectives in the digital era*. Springer. https://doi.org/10.1007/978-3-031-31193-2
McChesney, K., & Aldridge, J. M. (2021). What gets in the way? A new conceptual model for the trajectory from teacher professional development to impact. *Professional Development in Education, 47*(5), 834–852. https://doi.org/10.1080/19415257.2019.1667412
McGonagle, A., Freake, H., Zinn, S., Bauerle, T., Winston, J., Lewicki, G., Jehnings, M., Khan-Bureau, D., & Philion, M. (2014). Evaluation of STRONG-CT: A program supporting minority and first-generation U.S. science students. *Journal of STEM Education, 15*(1).
Moher, D., Shamseer, L., Clarke, M., Ghersi, D., Liberati, A., Petticrew, M., Shekelle, P., & Stewart, L. A. (2015). Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement. *Systematic Reviews, 4*(1), 1–9. https://doi.org/10.1186/2046-4053-4-1
Mohr, S. C., & Shelton, K. (2017). Best practices framework for online faculty professional development: A Delphi study. *Online Learning Journal*, 21(4).
Morais, I., & Bachrach, M. S. (2019). Analyzing the impact of computer science workshops on middle school teachers. *IEEE Integrated STEM Education Conference (ISEC), 2019*, 57–61. https://doi.org/10.1109/ISECon.2019.8882115

Mouza, C., Pollock, L., Pusecker, K., Guidry, K., Yeh, C.-Y., Atlas, J., & Harvey, T. (2016). Implementation and outcomes of a three-pronged approach to professional development for CS Principles. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (pp. 66–71). https://doi.org/10.1145/2839509.2844585

Mumcu, F., Uslu, N. A., & Yıldız, B. (2023). Teacher development in integrated STEM education: Design of lesson plans through the lens of computational thinking. *Education and Information Technologies, 28*(3), 3443–3474. https://doi.org/10.1007/s10639-022-11342-8

Neutens, T., & Wyffels, F. (2018). Bringing computer science education to secondary school: A teacher first approach. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 840–845). https://doi.org/10.1145/3159450.3159568

Ouyang, Y., Hayden, K. L., & Remold, J. (2018). Introducing computational thinking through non-programming science activities. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 308–313). https://doi.org/10.1145/3159450.3159520

Ozturk, Z., Dooley, C. M., & Welch, M. (2018). Finding the hook: Computer science education in elementary contexts. *Journal of Research on Technology in Education, 50*(2), 149–163. https://doi.org/10.1080/15391523.2018.1431573

Peters-Burton, E., Rich, P. J., Kitsantas, A., Laclede, L., & Stehle, S. M. (2022). High school science teacher use of planning tools to integrate computational thinking. *Journal of Science Teacher Education, 33*(6), 598–620. https://doi.org/10.1080/1046560X.2021.1970088

Peters-Burton, E., Rich, P. J., Kitsantas, A., Stehle, S. M., & Laclede, L. (2023). High school biology teachers' integration of computational thinking into data practices to support student investigations. *Journal of Research in Science Teaching, 60*(6), 1353–1384. https://doi.org/10.1002/tea.21834

Pokorny, K., & White, N. (2012). Computational thinking outreach: Reaching across the K-12 curriculum. *Journal of Computing Sciences in Colleges, 27*(5), 234–242.

Pollock, L., Mouza, C., Czik, A., Little, A., Coffey, D., & Buttram, J. (2017). From professional development to the classroom. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 477–482). https://doi.org/10.1145/3017680.3017739

Rachmatullah, A., & Wiebe, E. N. (2023). Changes and sources of changes of middle school teachers' self-efficacy for teaching science in a computationally rich environment: A mixed-methods study. *Journal of Science Teacher Education, 34*(2), 132–156. https://doi.org/10.1080/1046560X.2022.2035990

Reding, T. E., & Dorn, B. (2017). Understanding the "teacher experience" in primary and secondary CS professional development. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 155–163). https://doi.org/10.1145/3105726.3106185

Relevo, R. (2012). Chapter4: Effective search strategies for systematic reviews of medical tests. *Journal of General Internal Medicine, 27*, 28–32. https://doi.org/10.1007/s11606-011-1873-8

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*. https://doi.org/10.1016/j.compedu.2021.104196

Roth, K. J., Bintz, J., Wickler, N. I. Z., Hvidsten, C., Taylor, J., Beardsley, P. M., Caine, A., & Wilson, C. D. (2017). Design principles for effective video-based professional development. *International Journal of STEM Education, 4*(1), 1–24. https://doi.org/10.1186/S40594-017-0091-2/TABLES/5

Sabin, M., Smith, A., DuBow, W., & Deloge, R. (2018). Creative computing challenge: teacher professional learning to enhance non-computing career and technical education curricula with engaging computational practices for all students. In *2018 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. https://doi.org/10.1109/RESPECT.2018.8491714

Scherer, R., Siddiq, F., & Viveros, B. S. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology, 111*(5), 764–792. https://doi.org/10.1037/EDU0000314

Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a phenomenology of computational thinking in STEM education. In M. Khine (Ed.), *Computational thinking in the STEM disciplines: Foundations and research highlights* (pp. 49–72). Springer International Publishing. https://doi.org/10.1007/978-3-319-93566-9_4

Sherwood, H., Yan, W., Liu, R., Martin, W., Adair, A., Fancsali, C., Rivera-Cash, E., Pierce, M., & Israel, M. (2021). Diverse approaches to school-wide computational thinking integration at the elementary grades: A cross-case analysis. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 253–259). https://doi.org/10.1145/3408877.3432379

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

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Simmonds, J., Gutierrez, F. J., Casanova, C., Sotomayor, C., & Hitschfeld, N. (2019). A teacher workshop for introducing computational thinking in rural and vulnerable environments. In *SIGCSE 2019-Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1143–1149). https://doi.org/10.1145/3287324.3287456

Simmonds, J., Gutierrez, F. J., Meza, F., Torrent, C., & Villalobos, J. (2021). Changing teacher perceptions about computational thinking in grades 1–6, through a national training program. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 260–266). https://doi.org/10.1145/3408877.3432542

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). https://doi.org/10.1145/3408877.3432482

Tengler, K., Kastner-Hauler, O., & Sabitzer, B. (2021). Tell, draw and code – Teachers' intention to a narrative introduction of computational thinking. In E. Barendsen & C. Chytas (Eds.), *Informatics in schools: Rethinking computing education* (pp. 29–42). Springer International Publishing. https://doi.org/10.1007/978-3-030-90228-5_3

Tran, Y. (2019). Computational thinking equity in elementary classrooms: What third-grade students know and can do. *Journal of Educational Computing Research, 57*(1), 3–31. https://doi.org/10.1177/0735633117743918

Tsouccas, L., & Meletiou-Mavrotheris, M. (2017). Enhancing the Technological, Pedagogical and Content Knowledge (TPACK) of in-service primary teachers in the use of tablet technologies. In *Proceedings of the 16th World Conference on Mobile and Contextual Learning* (pp. 1–8). https://doi.org/10.1145/3136907.3136951

van Veen, K., Zwart, R., & Meirink, J. (2012). What makes teacher professional development effective? A literature review. In M. Kooy & K. van Veen (Eds.), *Teacher learning that matters: International perspectives* (pp. 23–41). Routledge. https://doi.org/10.4324/9780203805879-8

Vieira, C., & Magana, A. J. (2013). Using backwards design process for the design and implementation of computer science (CS) principles: A case study of a Colombian elementary and secondary teacher development program. *IEEE Frontiers in Education Conference (FIE), 2013*, 879–885. https://doi.org/10.1109/FIE.2013.6684950

von Wangenheim, A., Gresse Von Wangenheim, C., Pacheco, F. S., Hauck, J. C. R., & Ferreira, M. N. F. (2017). Motivating teachers to teach computing in middle school: A case study of a physical computing taster workshop for K-12 teachers. *International Journal of Computer Science Education in Schools, 1*(4), 35–49. https://doi.org/10.21585/ijcses.v1i4.17

Wang, C., Shen, J., & Chao, J. (2022). Integrating computational thinking in STEM education: A literature review. *International Journal of Science and Mathematics Education, 20*(8), 1949–1972. https://doi.org/10.1007/s10763-021-10227-5

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Kemi, J., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*, 127–147. https://doi.org/10.1007/s10956-015-9581-5

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Wing, J. M. (2010). Computational thinking: What and why? *The Link Magazine, 6*, 20–23.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education, 14*(1), 1–16. https://doi.org/10.1145/2576872

Zhang, Y., Ng, O. L., & Leung, S. (2023). Researching computational thinking in early childhood STE(A)M education context: A descriptive review on the state of research and future directions. *Journal for STEM Education Research, 6*(3), 427–455. https://doi.org/10.1007/s41979-023-00097-7