Computer Science Education

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ncse20

Computer science teacher professional development and professional learning communities: a review of the research literature

Lijun Ni, Gillian Bausch & Rebecca Benjamin

To cite this article: Lijun Ni, Gillian Bausch & Rebecca Benjamin (2021): Computer science teacher professional development and professional learning communities: a review of the research literature, Computer Science Education, DOI: 10.1080/08993408.2021.1993666

To link to this article: https://doi.org/10.1080/08993408.2021.1993666

Published online: 19 Oct 2021.

COMPUTER SCIENCE EDUCATION
https://doi.org/10.1080/08993408.2021.1993666

Routledge
Taylor & Francis Group

REVIEW ARTICLE

Computer science teacher professional development and professional learning communities: a review of the research literature

Lijun Ni, Gillian Bausch and Rebecca Benjamin

Department of Educational Theory and Practice, University at Albany, State University of New York, Albany, USA

ABSTRACT

Background & Context: Many efforts have been dedicated to building computer science (CS) teacher capacity through offering professional development (PD) programs. Previous reviews indicated the need to offer more continual support for teachers. Recent research has shifted its focus to scaling up PD and sustaining teaching capacity by establishing PLCs for CS teachers.

Objective: This study aims to conduct a systematic literature review of recent research on K-12 CS teacher PD, with an explicit exploration of PLCs.

Method: Based on 48 selected articles of 41 programs, this study explored features that contributed to the effectiveness of PD, including (1) PD goals, (2) theoretical frameworks and PD models, (3) curriculum and pedagogy, (4) programming tools, (5) program structure and approach, and (6) PD evaluation. We also examined whether and how these programs were dedicated to establishing PLCs.

Findings: Findings indicate a considerable increase in the number of studies on CS teacher PD . More programs saw the promising roles of PLCs and explored a variety of approaches for community building and promoting teacher learning.

Implications: PLCs have immense potential for teacher development, including breaking teacher isolation and fostering collaboration. More research can enlighten the efforts for CS teacher preparation and development.

ARTICLE HISTORY
Received 18 January 2021
Accepted 12 October 2021

KEYWORDS
Teacher professional development; professional learning community; computer science teachers; review study

1. Introduction

The computer science education community has recently seen growing interest in and commitment to providing opportunities for all students to learn computer science (CS) in the United States (Code.org, CSTA & ECEP Alliance, 2020; Gallup & Google, 2020). The recent efforts for expanding access to CS education significantly increases the number of CS teacher professional development (PD) programs. In the past four years (2017–2020),

CONTACT Lijun Ni lni@albany.edu Department of Educational Theory and Practice, University at Albany, State University of New York, Albany, USA

over 170 organizations have announced CSforAll commitments focused on PD and continual support for K-12 CS educators, providing support to 186,658 CS teachers (CSforAll, 2020).

Many challenges have been reported in teaching CS in the K-12 setting. It is often difficult to get enough teachers and even harder to acquire teachers who specialize in this field (Haduong & Brennan, 2019; Yadav, 2017). There is also confusion around the certification requirements and the necessary knowledge and skills for teaching CS (Code.org, CSTA & ECEP Alliance, 2020; Sentance et al., 2014). The CS standards for education also differ between states, making it difficult for educators to know exactly what classes to offer and how to teach them. CS teachers often report feeling isolated, as they might be the only CS teacher in their buildings (Goode et al., 2020b; Yadav, 2017). This leads to a situation where they might become a catchall for any CS or related course, meaning that they might be teaching out of their comfort zone. The isolation CS teachers experience is also related to the dearth of PD opportunities available in the geographical regions where they teach (Simmonds et al., 2019). Additional challenges in teaching CS result from a lack of appropriate teaching space, trouble accessing technology, and unclear or changing directives from administrators who may not value CS as a standalone subject in the schedule (Price et al., 2016). All of these conditions have led to a shortage of CS teachers. This situation can put pressure on teachers in other areas to include CS in their curriculum, even though they do not always identify as CS teachers or perceive themselves as capable and effective CS educators.

Situated with the many challenges in K-12 CS education, professional development provides important opportunities for building teacher capacity together with unique challenges for both teachers and PD providers. Understanding the current state of CS teacher PD can shed lights on future directions for research and practice in this area. In this study, the main goal is to systematically review recently published literature on K-12 CS teacher PD, exploring important features of effective and sustainable PD programs.

Previous review studies

As K-12 CS education is a relatively new area of research, there have been only few studies reviewing the status of CS teacher PD programs in the past decade. Most recently, Menekse (2015) conducted a systematic literature review of studies published between 2004 and 2014. This study also cited one relevant report (Franke et al., 2013) and one conference proceeding paper (Liu et al., 2011) that reviewed CS PD programs in the United States. Menekse highlighted a few important findings from these two prior studies. The first landscape study (Franke et al., 2013) surveyed 65 unique high school CS PD providers with 76 programs for high school teachers. The landscape study primarily found that CS teacher PD in the U.S. was not coherent and did not specifically address the needs of teachers in terms of CS pedagogical content knowledge (PCK). The other study referenced (Liu et al., 2011) reviewed 11 programs in terms of their structure, and found most of them provided single opportunities for PD focusing on a particular topic or programming tool.

Menekse's (2015) review study included 21 journal articles and conference papers. The initial research provided comprehensive information on the contextual features of the PD programs, including the organization and funding resource, program structure, goal of

PD, and specific CS concepts and programming tools covered by the program. Furthermore, Menekse (2015) evaluated those studies by using six key factors to assess program effectiveness in terms of the potential to change teaching practices and promote student learning. These six factors included (1) PD duration, (2) support for implementation, (3) explicit focus on active learning methods, (4) explicit focus on PCK, (5) collaboration with local district or school administration, and (6) student learning data as a result of the provided PD program. The prior study (Menekse, 2015, p. 346) concluded that "a majority of the CS teacher professional development programs lacked the fundamental requirements for high quality and effective professional development to improve teacher practices and eventually enhance student learning". It highlighted three major barriers to effective and sustainable CS PD. First, the lack of collaboration between higher education institutions and local school organizations was one of the main obstacles to the sustainability of PD and its long-term effects on changing teachers' practices. Second, a majority of the programs were short and provided limited on-going support for teachers. Third, most PD programs lacked a clear focus on the discipline-specific PCK for teaching CS (Menekse, 2015).

Menekse's (2015) study also revealed that many published studies lacked information on two critical areas. First, most of these studies did not include evidence for the effectiveness of these programs in terms of the effect on teacher practices and student learning in actual classrooms. Second, the majority of these studies lacked theoretical frameworks, which did not provide connections and related discussion regarding the existing teacher PD literature (Menekse, 2015).

The previous study provided valuable information to help the CS education community understand the prior state of CS teacher PD programs. Presently, many more initiatives are working to expand CS education by increasing CS teacher capacity. Therefore, the current review study aims to provide a more current picture of efforts in this area.

Professional learning communities for CS teachers

The isolation of CS teachers calls attention to the need to provide PD that encourages teacher collaboration and the development of teacher networks (Cutts et al., 2017; Ni et al., 2011; Ryoo et al., 2016; Sentance & Humphreys, 2018). The notion of a Professional Learning Community, or PLC, highlights the potential that a group of people based inside and outside a school can mutually enhance each other's and students' learning, as well as school development (Stoll et al., 2006). The implementation of PLCs is considered one of the most powerful organizational strategies for achieving substantive instructional improvement and critical student learning outcomes (Woodland, 2016). PLCs have received much attention in educational literature throughout the past three decades, resulting in a plethora of definitions and a variety of descriptions. This study builds on the definition of PLC in the PLC literature review from Stoll et al. (2006): Professional Learning Communities consist of "a group of people sharing and critically interrogating their practice in an ongoing, reflective, collaborative, inclusive, learning-oriented, growth-promoting way operating as a collective enterprise" (p. 223).

The use of PLCs is a popular model of effective PD (Darling-Hammond et al., 2017). There are many benefits to having high-quality PLCs. Members of PLCs spend time implementing a cycle of improvement by looking at instruction, data, outcomes, and ways to better their instruction (Woodland, 2016). This process of inquiry supports student performance, but beyond that, it leads to higher teacher satisfaction (Mclaughlin & Talbert, 2006; Vescio et al., 2008; Woodland, 2016). Other benefits of effective teacher communities include increased teacher self-efficacy and an increase in collaboration, which minimizes teacher isolation. The process of reflecting and analyzing knowledge collaboratively with peers allows community participants to engage in a "culture of practice" (Lave & Wenger, 1991) that helps develop shared meanings, a sense of belonging, and increased understanding (Stoll et al., 2006), all of which contribute to student learning and higher achievement results. PLCs offer great promise in removing the barrier of teacher isolation and helping teachers build on their identity and capacity for CS education. Considering the challenges of CS teacher PD as identified by the previous studies (Menekse, 2015), research on PLCs indicates that developing and strengthening teacher support networks or PLCs may be a crucial step for the diffusion of K-12 CS education.

PLCs may operate at various levels, and the types of PLCs can vary significantly. Examples include PLCs focused on lesson study, collaborative action research, grade level meetings within schools, and discussion groups (Ryoo et al., 2016; Stoll et al., 2006). In this study, we are interested in examining whether and how recent CS teacher PD programs purposefully support the development of any type of PLCs with continual learning, ongoing support, and various collaboration opportunities. Thus, CS teacher PLCs can be a school-based instructional support network (e.g. Mazur & Woodland, 2018), but the designation of "PLC" could also refer to those collaborative and reflective communities built among CS educators from different schools who continually join PD events (e.g. Goode et al., 2020b).

This literature review studied current trends in CS PD for K-12 educators and how the recent work addressed some of those challenges identified by the previous review (Menekse, 2015). In addition to examining fundamental features of those programs, this study explicitly reviewed whether and how these programs were dedicated to establishing PLCs to sustain and scale up PD efforts.

2. Methodology

To systematically review recent studies on CS teacher PD and PLCs, we searched multiple databases, including ACM Digital Library, ERIC, and Taylor and Francis. A query was conducted for articles published between 2015 and July 2020. We included "teacher community" and "teacher support" as keywords to retrieve relevant studies that might not use the term PLC, but focused on building teacher communities and offering teacher support. The following keywords were used in the search: "**teacher professional development**" OR "**teacher network**" OR "**teacher community**" OR "**teacher support**" AND "**computer science**" OR "**computational thinking**". Additionally, we explicitly searched 11 peer-reviewed journals and conference proceedings focused on CS education. Figure 1 shows the flowchart of the paper selection process.

A total of 655 articles were initially identified from the search with many duplicates across different search results. To be included in the review, articles were manually screened to determine the appropriateness of inclusion for the review. We only included research papers that explicitly studied PD and PLCs for K-12 CS educators. First, irrelevant

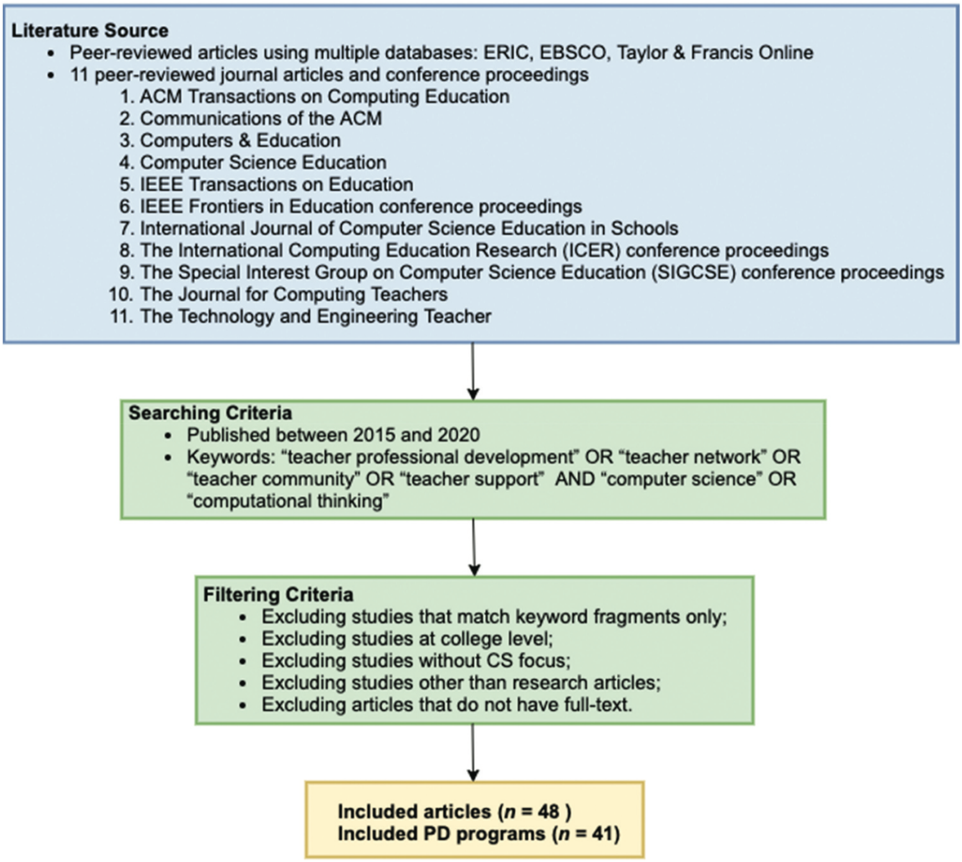

Figure 1. Flowchart of the study selection process.

articles were eliminated based on a screen of the article titles. We removed articles that contained only partial versions of the keywords (e.g. student learning community rather than professional learning community intended for teachers). Second, we reviewed the abstract to exclude articles that deviated from the main topic of PD and PLCs for K-12 CS educators. In this process, articles were filtered out which did not focus on CS education but referenced computer usage in PLCs. For the last step of screening for relevance, we read the articles and eliminated any additional studies that did not provide specific information on the PD programs, since they focused on other aspects of CS education (e.g. the design of the CS curriculum). Eventually, a total of 48 studies of 41 PD programs were included in this review. Table A1 in Appendix lists the included articles, providing information on the studied PD program, grade level, curriculum, whether it focused on building a PLC, and where the program was located.

The studies included in this review can be grouped into two categories: articles that presented PD programs with features that contributed to their effectiveness (as identified by Menekse, 2015) and a subset of the articles that included elements of PLCs supporting PD (n = 28). The articles were analyzed for content, and major themes were identified utilizing previous literature reviews as a guide. First, the following themes were used to code information on the PD programs: (1) PD goals; (2) theoretical framework and PD model; (3) curriculum and pedagogy; (4) programming tools; (5) PD structure and approach; (6) PD evaluation. In addition, we specifically examined whether and how these PD programs focused on PLC building, observing: (1) goal of PLCs; (2) definition of PLCs; (3) PLC format and approach; (4) PLC evaluation. Results are presented in Section 3 and Section 4.

3. Findings: computer science teacher professional development programs

Our analysis revealed that the total number of CS teacher PD programs has increased considerably over the past five years. While there were 21 published studies between 2004 and 2014 (Menekse, 2015), a total number of 48 studies of 41 PD programs were included in this review after careful selection and evaluation. Among these programs, 30 programs were from the United States, and 11 programs were located outside the U.S. Many programs served teachers of lower grades or across grade levels. In total, nine programs were designed for K-12 teachers in general. Nineteen programs served elementary school teachers, including six programs exclusively designed for elementary (primary) school teachers, three programs for both elementary and middle school teachers, and another one program for elementary and high school teachers. Four programs focused entirely on middle schools. PD programs serving high school teachers were still the most prevalent: in addition to the nine programs for K-12 teachers, another ten were designed exclusively for high school educators, with another eight for both middle and high school teachers and one for elementary and high school teachers.

3.1. Goal of professional development

The overarching goals of the 41 PD programs can be broadly categorized into three themes: (1) To build teacher capacity and self-efficacy; (2) To broaden participation in computing through K-12 CS education; (3) To develop a scalable and sustainable PD model. Figure 2 illustrates the main goals of these programs. Each program is represented into a number as labeled in Table A1.

First, most of the PD programs prioritized the goal of improving teachers' CS content knowledge base, as well as their pedagogical content knowledge and confidence in delivering CS instructions to students. This finding is aligned with Reding and Dorn (2017) recommendation on the three interdependent facets of knowledge that PD programs should help teachers build. The first facet, known as "explicit knowledge", is associated with the cognitive processes of learning. The second facet is the behavior, actions, and accumulated experiences of the learning process, known as "implicit knowledge". In addition, Reding and Dorn (2017) also mentioned the significance of dealing with the emotional component of teacher learning, known as the "emancipatory knowledge", which liberates teachers from fears and constraints that limit them from learning and embracing new knowledge. One example is the CS Education Research Group (CSER) Digital Technologies program (Falkner et al., 2018). This program adopted an "Ecosystem Approach", in which a suite of programs, such as Massive Open Online Courses (MOOCs), a lead teacher support program (called PL-in-a-box), and a technical support program (Lending Library), were created to operate the ecosystem. The program incorporated

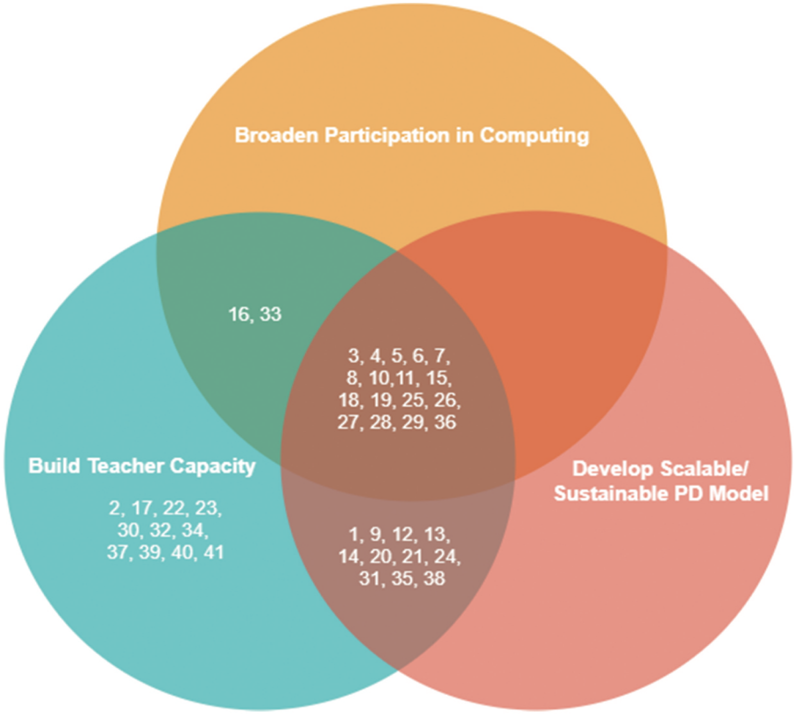

Figure 2. Goals of the CS professional development programs.

both formal and informal professional learning programs for CS educators to acquire both content and pedagogical content knowledge. The Lending Library was formed to alleviate the anxiety caused by potential technical issues. Likewise, other programs such as the Exploring Computer Science (ECS) program (Goode et al., 2019), the Beauty and Joy of Computing (BJC) PD (Price et al., 2016), and the WeTeach_CS project (Warner et al., 2019) also attempted to support some or all of these knowledge facets outlined by Reding and Dorn (2017). The previous review (Menekse, 2015) revealed the lack of clear focus on the discipline-specific PCK in the earlier PD programs (2004–2014). Our review indicates that 25 out of the 41 programs started to explicitly address this important need in developing CS teachers.

The second set of goals, to broaden participation in computing (BPC), has been one of the common PD goals for more than a decade (Menekse, 2015, p. 340). This goal is still prominent in the reviewed studies. Nineteen programs explicitly stated this goal, such as the ECS program (Goode et al., 2020a), Code.org's CS Fundamentals PD program (Roberts et al., 2018), and CS4Alabama (Gray et al., 2015).

This study revealed a third set of goals, focused on developing PD programs through establishing the infrastructure to further support teachers, and eventually to make the programs scalable and sustainable. Many successful PD programs followed community-based models. Some of these programs had an explicit community component that provided continual support for participants. For example, the ECS program (Goode et al., 2019) built an online PLC by having CS teachers work together remixing the ECS lessons to meet the needs of individual classes. The Professional Learning and Networking in Computing (PLAN C) program (Cutts et al., 2017) set up a national network of CS teacher groups acting as a reflection and discussion forum. Section 4 will further examine how those programs started to build PLCs for K-12 CS educators.

While many PD programs primarily focused on achieving these three sets of general goals, some programs also developed specific goals. For example, the ECS program also aimed to help teachers transition to an equity-based and inquiry-based classroom culture (Ryoo et al., 2016). A number of other programs also followed ECS on building equity and inquiry-based CS classrooms, such as the BJC PD (Price et al., 2016) and the Code.org CS Fundamentals PD program (Roberts et al., 2018). In addition, some programs aimed to develop effective practices to support teachers in integrating CS into other subjects, e.g. the Infusing Computing PD (Jocius et al., 2020). Last, one program, the Arts & Bots Math and Science Partnership program (Hamner et al., 2016), stated a special goal of helping teachers to understand and identify their students' talents through CS projects.

3.2. Theoretical frameworks and PD models

Prior literature affirms the importance of theoretical frameworks for PD programs. Menekse (2015) pointed out that many prior CS PD programs lacked the underpinning theoretical frameworks, which did not provide connections and discussions regarding existing teacher PD literature. Our review found that more studies have sought to establish their PD programs based on sound theories and existing PD models. Five theories have been identified from the review that guided the program design, including Critical Race Theory, Situated Learning, Community of Inquiry, Social Constructivism and Social Network Analysis. Another six PD models or approaches have been used to assist the design, execution and analysis of the PD programs. In this section, we introduce these existing theoretical frameworks and PD models that were applied by the authors as well as PD models that were adopted or adapted from the literature to organize or analyse the CS teacher PD programs.

- Critical Race Theory (CRT) was used as a framework to examine "the ways race and racism impact structures, discourses, and processes" (Goode et al., 2020a, p. 356). The study by Goode et al. (2020a) showed how the theory informed the ECS program in fostering conversations on race-infused, equity-based curriculum and teaching practice.
- Situated Learning Theory suggests that learning is a social process whereby knowledge is co-constructed, and such learning is situated in a specific context and embedded within a particular social and physical environment (Lave & Wenger, 1991, p. 40). It explains how participants' learning in a community of practice (CoP) occurs and changes as they are assimilated into that community. The Computing at School (CAS) program explicitly draws on this theory and models an innovative approach to professional learning for computing teachers based on community and peer support (Sentance & Humphreys, 2018). More PD programs has applied the concept of the CoP (e.g. Cooper et al., 2015; Kosmas, 2017) to foster PLCs for CS teachers, as discussed in Section 4.2.

- Community of Inquiry (CoI) has been expanded and applied to online learning contexts (Garrison et al., 2000). The Mobile CSP project structures its online PD around the CoI model (Rosato et al., 2017). This model suggests three elements in the online learning environment that need to be addressed: social presence, cognitive presence, and teaching presence. This approach also reinforces the idea of establishing strong learning communities that include collaboration and reflection for CS teacher PD.
- Social Constructivism: The Applied Mathematics, Physics, and Engineering Design (RAMPED) program embraced a social constructivism framework, where interactions between people allowed the pre-collegiate teachers to create connections, develop content understanding, and develop CS self-efficacy (Borowczak & Burrows, 2019; Vygotsky, 1978). In that study, the social constructivism approach guided the researchers in developing a collaborative learning environment for teachers as well as the data collection and analysis on teacher content knowledge and perceptions.
- Social Network Analysis (SNA) is a framework used to analyze and assess social resource and network development for PD programs (e.g. Mazur & Woodland, 2018). Based on SNA, Reding and Dorn (2017) developed a quantitative study to examine teachers' network development in the Strategic Problem-based Approach to Rouse Computer Science (SPARCS) program and identify potential teacher leaders in order to sustain the social capital of the PD cohort.
- The Code, Connect, Create (3 C) Model (Jocius et al., 2020) was designed to teach computational thinking (CT) to content area teachers and support them infusing CT into their classrooms, applying a situative perspective on teacher learning and professional development (Borko, 2004). The 3 C PD model is centered around three primary components: Code (Bootcamp), Connect (connecting disciplinary content and pedagogy to CT), and Create (the development of CT-infused learning segments).
- The Collective Impact Models are viable solutions to complex social sector challenges through uniting interdisciplinary partners towards a common goal (Kania & Kramer, 2011). To bring high-quality PD opportunities to teachers in rural areas, the WeTeach_CS program implemented five key components of collective impact models: a backbone organization supporting the entire system, a common agenda (of broadening participation in computing), shared measurement systems, mutually reinforcing activities, and continuous communication across all stakeholders (Warner et al., 2019).
- The Ecosystem Approach (Falkner et al., 2018) was designed to support CS teacher PD, combining informal and formal learning opportunities through integrated, flexible, structured, and sustained support. This approach was developed based on Thomas and Autio (2014) adoption of organizational theory that conceptualized the organizational environment and characteristics of an ecosystem.
- The Five Tensions Negotiated was developed after the ScratchEd project (Brennan, 2015). Brennan identified five sets of tensions in CS PD to design and evaluate PD programs. Hickmott and Prieto-Rodriguez (2018) further identified an extra tension beyond the five tensions to enhance the constructivist learning experience.

- The Interconnected Model of Professional Growth (IMPG) is a model of professional growth that encompasses four interconnected and non-linear domains (external domain, personal domain, domain of practice, and domain of consequence) (Clarke & Hollingsworth, 2002). The Science Teaching Inquiry Group in Computational Thinking (STIGCT) program operationalized the IMPG model in the context of its CT PD to understand the professional growth of the elementary school teachers in the project (Ketelhut et al., 2020).
- The Lloyd and Cochrane (2006) model of professional development was initially developed for face-to-face teacher professional learning in Information and Communications Technology (ICT). This model defines four major characteristics of successful professional learning around context, time, community, and personal growth. The CSER program expanded these characteristics to design its self-directed MOOC courses (Falkner et al., 2017).

3.3. Curriculum and pedagogy

The reviewed studies indicated that the PD programs worked on preparing teachers to teach a variety of CS curricula. Figure 3 shows the number of programs that focused on the different types of curricula: curriculum created by the PD program (project), standardized CS curriculum, integrating CS and CT into existing curricula. Six studies did not provide specified curriculum information. Two of these programs (Mouza et al., 2016; Pollock et al., 2017) represented two types as they structured the PD around both the Advanced Placement (AP) CS Principles course for full implementation and infusing CS Principles modules into STEM.

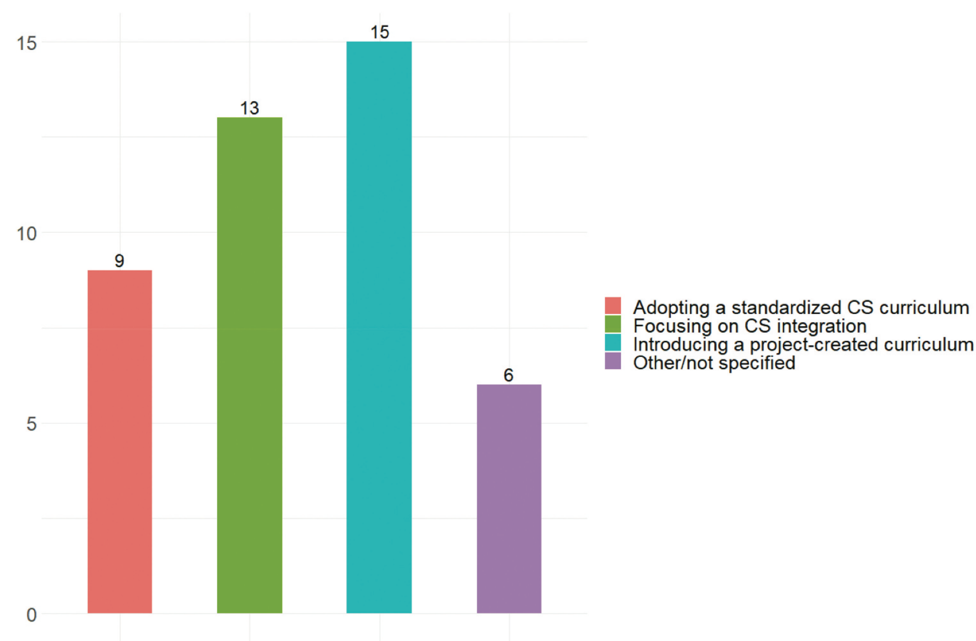

Figure 3. Curriculum introduced in the PD programs.

First, many programs focused on using standardized curricula to outline PD content and motivate teachers. An example of this is the AP CS Principles curriculum in the United States, which was developed by the College Board to "introduce students to the foundational concepts of the field and challenge them to explore how computing and technology can impact the world" (College Board, 2020). As summarized in Table A1, several PD programs were designed to prepare teachers to teach this new course and the AP CS A course (e.g. Granor et al., 2016; Hamlen et al., 2018; Leyzberg & Moretti, 2017). Furthermore, other countries also have their own national-level CS curriculum, such as the Australian Curriculum: Digital Technologies (CS Learning Area) (Falkner et al., 2018), and the UK's latest version of ICT curriculum, which prioritizes instruction in computer science and digital literacy (Brown et al., 2014).

Several PD programs developed their own curriculum or learning materials. A few curricula were designed for the AP CS Principles course, such as the BJC curriculum (Price et al., 2016) and the Mobile CSP curriculum (Rosato et al., 2017). The ECS project (Goode et al., 2020a) also created a high school introductory CS course (Exploring Computer Science), which was structured to facilitate inquiry-based and equity-based instructional practices. The Code.org CS Fundamentals curriculum (Roberts et al., 2018) included activities that fostered collaboration, student engagement, and equity, with the goal of broadening participation in CS. The NM-CSforAll project (Lee et al., 2017) developed a curriculum that integrated fundamental CS concepts, modeling and simulation, and the study of complex adaptive systems.

Noticeably, many PD programs did not provide a specific CS curriculum but supported teachers in creating and implementing curricular materials for integrating CS and other subjects. For example, the Arts & Bots program (Hamner et al., 2016) provided teachers with an opportunity to discuss the integration of robotics projects into their disciplinary classrooms. The SPARCS program (Reding & Dorn, 2017) also utilized PD sessions to encourage teachers to constantly consider how the new learning (of CS concepts) could be adapted to their classrooms relative to their specific content focus.

This review also explored the curriculum pedagogies introduced in the PD programs. We found that 14 out of the 41 programs employed or introduced specific pedagogies for CS. Since Ryoo et al. (2016) first introduced inquiry-based and equity-based learning to CS education through the ECS program, these pedagogical approaches have been adopted by many other PD programs. For instance, the BJC PD (Price et al., 2016) and the STIGCT program (Ketelhut et al., 2020) both employed these pedagogies, showing an example of how widespread these new pedagogies have become across the CS education community. Problem-based learning (PBL) was another learner-centered pedagogy that many PD providers desired to infuse into CS classrooms, which focused on solving real-world problems with computational tools. For example, the SPARCS program (Reding & Dorn, 2017) introduced the PBL pedagogy and provided continued support in response to the teachers' challenges when learning and implementing this new pedagogy.

3.4. Programming tools

This study also discovered that many PD programs used more than one programming language to train CS educators, as multiple programming languages and tools were used in K-12 CS education worldwide. For instance, the Progra-MEER program (Neutens &

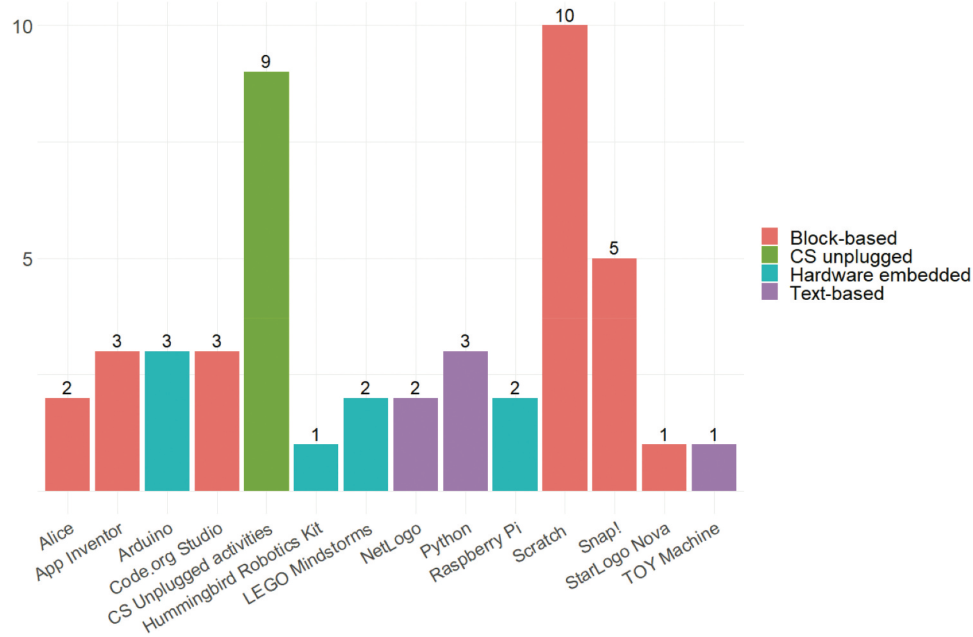

Figure 4. Distribution of programming tools used by the PD programs.

Wyffels, 2018) used Scratch, App Inventor, and Arduino. Figure 4 demonstrates the number of programs using different programming tools and materials. Among these tools, block-based programming environments, such as App Inventor, code.org's game lab, and Scratch, were popular among programs for K-8 CS teachers. Text-based programming tools were mainly used at the high school level, e.g. Python. In addition, the PD programs exhibited a trend of introducing physical computing tools. A number of programming tools were designed to engage students in programming with hardware (e.g. robotics kits and small single-board computers).

Nine programs introduced CS unplugged activities. For example, the CSER Digital Technologies Education program (Falkner et al., 2018) and the Integrating CT PD program (Yadav et al., 2018) used CS Unplugged activities to develop computational thinking skills with elementary school teachers. Noticeably, five of these programs used both plugged and unplugged activities in their CS curricula (Celepkolu et al., 2020; Jocius et al., 2020; Mouza et al., 2016; Price et al., 2016; Roberts et al., 2018).

3.5. Program structure and approach

This review analyzed the PD structure and approach in order to observe the primary format and key strategies employed in the CS teacher PD programs. Traditional formats include in-person workshops and summer institutes (Menekse, 2015, p. 337). Many of the reviewed programs delivered PD fully or partially in-person (n = 37). Noticeably, to address the geographical restrictions of teacher communities, the ECS program (Goode et al., 2020a) developed residential PD. ECS teachers from areas not served by regional hubs traveled to the location where the PD took place to participate in the PD and build a

CS teaching community. Most programs were structured into multiple sessions, with 7 of the 41 programs offering only one-time workshops lasting one week or less. Along with in-person PD events, many programs (*n* = 30) provided hybrid learning opportunities combing face-to-face workshops with a variety of learning activities such as online courses, webinars, coaching and community meetings (Gray et al., 2016; Morelli et al., 2015; Warner et al., 2019). For example, the JBC program offered an online Python course and required participants to finish the course prior to the in-person PD (Reimer et al., 2018). Three PD programs took place fully online through community-centric MOOCs (Falkner et al., 2017), email-based learning (Haduong & Brennan, 2019), and an online Community of Practice (Kosmas, 2017).

Effective teacher PD practices involve opportunities for sustained learning and time allowances for teachers to integrate and reflect (Falkner et al., 2018). Following up with formal workshops and courses, professional learning opportunities can be provided through PLCs that support teachers with continual access to peer support. Many (28 out of the 41) recent CS teacher PD programs initiated online or offline learning communities for teacher collaboration and communication (e.g. Brown et al., 2014; Kosmas, 2017; Price et al., 2016). Section 4.3 provides more information on the format these PD programs used to build PLCs for CS educators.

Dynamic and rigorous approaches to support teacher learning are crucial to the success and sustainability of a PD program (Menekse, 2015). Activities and approaches applied in the reviewed PD programs varied remarkably. We observed 13 major approaches from these programs (see Table 1). The ECS program initially created several approaches to train teachers, including the Teacher-Learner-Observer (TLO) model, the in-class coaching model, and building online PLCs. These approaches were then adopted by other CS PD programs. For instance, the Code.org CS Fundamentals PD used the TLO model (Roberts et al., 2018); The SPARCS program established a PLC during "teacher institute weeks" (Reding et al., 2016); The Partner4CS program (Mouza et al., 2016) adapted the in-class coaching method. Notably, Flatland et al. (2018) reported a PD model that embedded college CS faculty into high school classrooms to coach teachers in teaching a college-level programming course. Another pedagogical approach, pair-programming, was used by several other programs (Mouza et al., 2016; Price, et al., 2016). In addition, several innovative approaches surfaced from this study, such as student academy (Reding et al., 2016), project scoring (Kao et al., 2020), and lesson design contest

Table 1. Approaches employed by the PD programs.

PD Approach	Example
1. In-classroom coaching/support: teacher support provided in their classrooms	Ryoo et al., 2016
2. Teacher-Learner-Observer model: a rehearsal-based approach	Goode et al., 2020a
3. Peer-instruction model: teacher teaching teachers	Price, et al., 2016
4. Lead/master teacher model: lead teacher supporting their peers	Cutts et al., 2017
5. Paring computing industry professional with classroom teachers	Granor et al., 2016
6. Embedding higher education faculty into high school classrooms	Flatland et al., 2018
7. Student academy model: teachers co-teach lessons and plan for curriculum	Reding et al., 2016
8. Team/individual assignments: teachers were given the same assignment as their students	Lee et al., 2017
9. Building an online PLC to support collaborative professional learning	Goode et al., 2020a
10. Project scoring: teachers work on rubric writing and scoring activities	Kao et al., 2020
11. Workshop with culminating competition: teacher lesson design competition	Alghamdi et al., 2019
12. Sponsoring teachers attending conferences for networking development	Gray et al., 2016
13. Pair-programming: teachers learn coding with pair-programming	Mouza et al., 2016

(Alghamdi et al., 2019). Overall, the lead teacher model was the most popular approach utilized by 14 out of the 41 programs (e.g. Cutts et al., 2017; Falkner et al., 2018; Gray et al., 2016). Some PD programs referred to this approach as “teacher mentor” or “master teacher” (e.g. Ketelhut et al., 2020; Sentance & Humphreys, 2018).

3.6. PD evaluation

Evaluation can help PD programs determine whether the PD activities are achieving their goals. This section presents results on how the studies evaluated the effectiveness of their PD programs, focused on (1) the types of evaluation (instruments used); (2) the areas of evaluation, using Guskey’s (2002) five levels of PD evaluation to categorize the evaluation areas and discuss the evaluation results.

Types of evaluation

All the reviewed studies except one study (Cooper et al., 2015) provided evaluation information. Various instruments were used in these studies to evaluate the effectiveness of PD. Traditionally, researchers used self-reported tools, such as surveys and interviews, for PD evaluation (Menekse, 2015). Our results align with the previous findings: surveys (n = 40) and interviews (n = 16) were still the most popular tools (see Figure 5). Many studies used surveys and interviews to learn about teachers’ PD experience (e.g. Falkner et al., 2017) and assess their confidence in implementing the CS curriculum (e.g. Celepkolu et al., 2020).

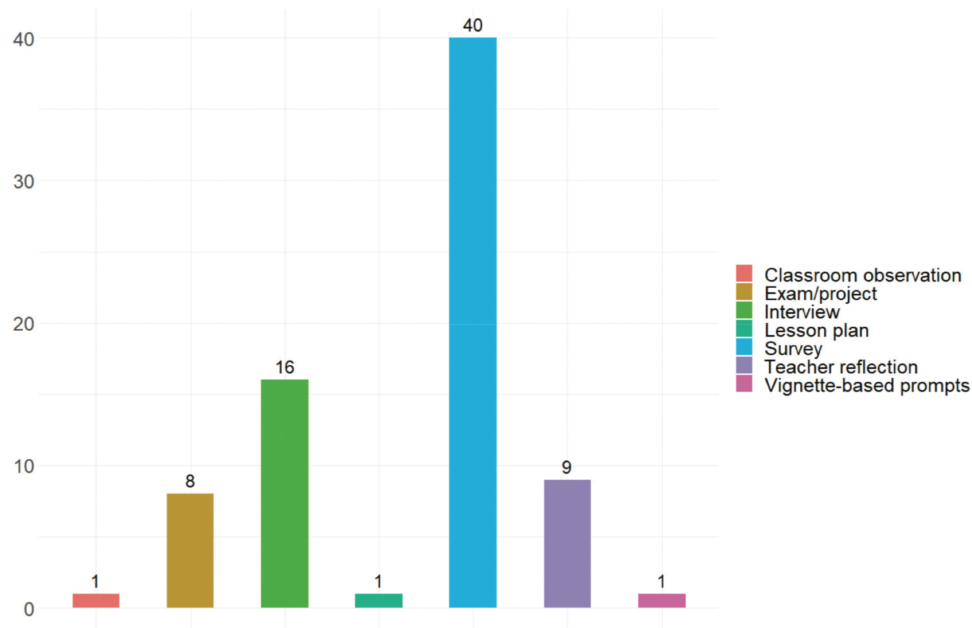

Figure 5. Percentage for evaluation methods used in the CS PD programs.

Another type of instrument used to assess PD programs was the teacher's reflection and daily diary. For instance, the STIGCT program (Ketelhut et al., 2020) used teachers' reflections to understand their PD experience. The data was analyzed using the Interconnected Model of Professional Growth, which developed non-linear and interconnected domains of practice and consequences to represent teachers' growth pathways. Teachers' development occurred through reflection between and within each of the domains. The SPARCS program (Reding & Dorn, 2017) used daily online journal prompts to understand teachers' experience and refine the program accordingly. Another innovative method, vignette-based assessment prompts, asked teachers to reflect on a specific CS teaching scenario in order to measure their understanding of computational thinking in the context of teaching (Yadav et al., 2018). Classroom observation (Granor et al., 2016) and analysis of lesson plans (Ahmed et al., 2020) were two other methods used to assess teacher learning and classroom implementation. Additionally, performance-based assessments (e.g. exams and projects) were also used to evaluate teachers' content knowledge (Borowczak & Burrows, 2019; Celepkolu et al., 2020; Lee et al., 2017).

In addition to collecting data directly from teachers, some programs used student learning outcomes to demonstrate the overall impact of PD. For example, both the NM-CSforAll project (Lee et al., 2017) and the Utah Exploring Computer Science Initiative (Hu et al., 2017) asked students to evaluate their confidence in CS knowledge in a Likert scale survey. In the Mobile CSP PD program (Morelli et al., 2015; Rosato et al., 2017), researchers combined the survey results of students' attitudes towards the CS course and the summative student assessment (the final exam) to show the success of the PD program. The TEALS program (Granor et al., 2016) used the AP CS A exam to assess the learning outcomes of their TEALS classrooms.

A few studies also tracked participants after they completed the PD. For example, the We_Teach CS program applied the comparative interrupted time series to assess the impact of the PD program on teachers' change over time (Warner et al., 2019). Similarly, the Infusing Computing program (Jocius et al., 2020) and the Alice Community of Practice program (Cooper et al., 2015) also developed follow-up measures with teachers' classroom implementation.

Areas of evaluation and results

Effective PD evaluations require the collection and analysis of the five critical levels of information: (1) participants' reactions, (2) participants' learning, (3) organization support and change, (4) participants' use of new knowledge and skills, and (5) student learning outcomes (Guskey, 2002). In this review, we used Guskey's (2002) five levels of PD evaluation framework to analyze the evaluation results reported in these studies. We examined the specific areas (levels) each study focused on (see Figure 6), identified additional areas, and summarized where the PD program achieved its goals with any reported challenges.

Overall, most studies evaluated PD effectiveness in areas that fit into Guskey's five critical levels of PD evaluation (see Figure 6). Most programs examined multiple levels. For Level 1, those programs reported overall positive participant experience, such as enjoying the PD activities, easy access to the curriculum resources, and appreciating the networking and program support (e.g. Cutts et al., 2017; Mouza et al., 2016). Those types of feedback can be used to improve the PD design and delivery. For Level 2, the studied PD

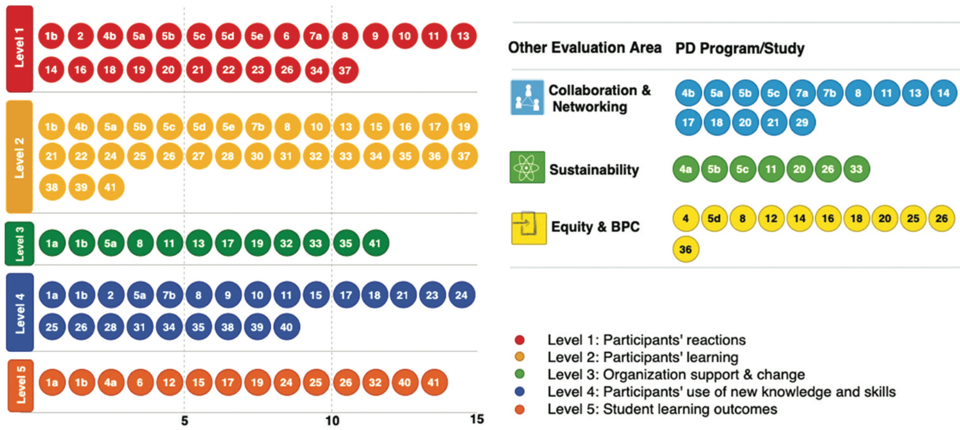

Figure 6. Evaluation areas of the CS PD programs.

programs also recorded substantial findings indicating participants were able to develop content knowledge and pedagogical (content) knowledge (e.g. Goode et al., 2019; Lee et al., 2017; Ryoo et al., 2016), as well as to increase their confidence in teaching CS (e.g. Borowczak & Burrows, 2019; Jocius et al., 2020; Reimer et al., 2018). A few programs did struggle to develop teacher capacity (especially PCK) within the short timeframe of the PD. These programs suggested continual PD to support teachers in their professional growth and building confident in their classrooms (e.g. Goode et al., 2020b; Neutens & Wyffels, 2018; Sentance & Humphreys, 2018).

For Level 3, the PD programs recorded nearly unanimous feedback that organizational support was insufficient and presented a significant challenge in continuing and sustaining the programs, except for the Introduction to Coding Workshop program (Carter & Crockett, 2018). Two main challenges observed were the lack of resources (e.g. access to technical equipment, Cutts et al., 2017) and administrative support within their schools (e.g. Morelli et al., 2015; Pollock et al., 2017; Ryoo et al., 2016). Despite the teachers' best efforts, the lack of administrative support prevented them from implementing what they learned from the PD. Those organizational barriers also discouraged teachers from continuing their participation in the PD (Price et al., 2016).

For Level 4, about half of the programs reported findings in this area with both positive results and challenges for teachers to apply what they learned from the PD. First, some programs found the PD had a big impact on participants' teaching practice, even with "transformational changes" (Cutts et al., 2017). The ECS program found the increased application of inquiry and equity-based teaching practices (Ryoo et al., 2016). Teachers from the STIGCT program (Ketelhut et al., 2020) came up with creative ways of incorporating CT activities into their subjects. Meanwhile, a few programs also reported that teachers' CS implementations were limited to the activities and scope of the PD with few adaptations and minimal systemic change in implementation behaviors. Teachers reported that either they were not given enough time to process the new knowledge (Neutens & Wyffels, 2018), or the PD did not offer in-depth CS knowledge needed for spontaneous examples during classroom implementation and for fully developing a series of integrated lessons for ongoing implementation (Jocius et al., 2020).

For Level 5, 13 PD programs measured the student outcomes in terms of student performance and attitudes. The findings were universally positive and demonstrated enhanced student interest (Morelli et al., 2015) and confidence in learning CS (e.g. Hu et al., 2017), significant growth in CS knowledge (e.g. Lee et al., 2017), and test score (e.g. Granor et al., 2016). Meanwhile, it is worth noting that only a few studies (Granor et al., 2016; Hamlen et al., 2018; Morelli et al., 2015; Rosato et al., 2017) explicitly examined the impact on students in relation to whether the teachers had participated in the PD or not and to what extent they adopted the curriculum from the PD program. As one example, the Mobile CSP PD program (Morelli et al., 2015) found that in courses where the participating teachers covered more course materials from the PD program, the students performed better in their final exams. Similarly, Hamlen et al. (2018) reported that students who received instructions from teachers who participated in the summer PD gained a significant increase in both content knowledge and confidence.

Beyond the five levels of PD evaluation, we also observed results connected to three areas: collaboration and networking, sustainability, and equity and BPC (see Figure 6). Both collaboration and sustainability are two critical characteristics of effective PD (Darling-Hammond et al., 2017). Equity (and broadening participation in computing) is another important goal for many CS PD programs, as presented in Section 3.1. Many programs examined these areas as indicators of their effectiveness. First, these programs strived to build PLCs to support collaboration and continual professional learning. A few studies discussed findings regarding how collaboration and network building contributed to teachers' professional learning. For example, the ECS project found that their community-centered PD broke teacher isolation through teacher collaboration in authentic practices, which offered time and space to build their content knowledge, pedagogical skills and inclusive belief system (Goode et al., 2019; Ryoo et al., 2016). Similarly, several programs also found that their efforts to sustain the professional learning or scale up the program were (potentially) successful (Carter & Crockett, 2018; Cutts et al., 2017; Falkner et al., 2018; Goode et al., 2020b), and identified areas of improvement (Neutens & Wyffels, 2018). Section 4.4 presents more information on PLC evaluations.

Eleven programs explicitly discussed results on equity and BPC with either students or teachers, as evidence for PD effectiveness. For example, the CSER PD (Falkner et al., 2018) analyzed the demographics of their participants, showing diverse participation of teachers in the program. The WeTeach_CS PD (Warner et al., 2019) observed more growth in the number of certified teachers from rural areas, narrowing the gap of CS teacher supply between urban and rural areas. Several studies also analyzed students' learning outcomes by demographics (e.g. Gray et al., 2015; Hamlen et al., 2018; Neutens & Wyffels, 2018). For example, one study found Hispanic or Latino students had low confidence in programming skills, but their confidence significantly increased through the course, similar to their peers (Hamlen et al., 2018).

4. Findings: building professional learning communities

The second part of this review specifically examined the studies that highlighted building PLCs for K-12 CS educators. We classified the PD programs in regard to their designation and primary focus on PLCs for CS teachers. Sixteen of the 41 programs explicitly focused on establishing nationwide or local PLCs to support CS educators. Approximately 30% (12 out of 41) of the programs organized some activities to provide continual support and enhance teacher collaboration during and outside the PD, which showed the prototype of a PLC. Overall, about 70% of the 41 PD programs explicitly or implicitly stated they had either initiated a PLC or included some activities to promote teacher collaboration and network building. Another three studies proposed to build a PLC in the future, informed by their PD evaluation results.

4.1. Goal of PLCs

Across the review, three major areas emerged as the specific goals of a CS teacher PLC. The first commonly identified goal for PLCs was to break the professional isolation and encourage collaboration and network-building among CS teachers. For example, the ECS program (Goode et al., 2019) found that organizing teachers to work collaboratively in authentic practices eliminated the isolation that many CS teachers felt, and connected educators across schools. Many studies also aimed to support teachers by setting up a network of PD groups focusing on reflective discussions throughout their classroom implementations. For instance, the Arts & Bots Math and Science Partnership (Hamner et al., 2016) project and the NM-CSforAll program (Lee et al., 2017) suggested that curriculum resources shared in online communities provided a great deal of support to educators during curriculum implementation. PLCs were developed as a platform for teacher collaboration and helped the process of finding and selecting resources, as well as using and revising them. Similarly, the CSER PD (Falkner et al., 2018) concluded that PLCs should empower each teacher as "decision-makers and co-creators" of CS curriculum (p. 306).

Another common goal for building PLCs was to provide ongoing support for teachers inside and outside PD. In other words, PLCs create sustainable value for PD programs. For example, the Progra-MEER program (Neutens & Wyffels, 2018) intended to spark a "community of practice" through helping all stakeholders of schools recognize the significance of CS and encourage them to collaborate with each other. Likewise, a few other programs (Falkner et al., 2018; Gray et al., 2015; Reimer et al., 2018) also built sustainable communities to scaffold CS teaching and improve teachers' professional learning experiences.

The third main goal for PLCs in these studies was to strengthen teachers' CS content knowledge and especially PCK through resource sharing, curriculum co-designing, group discussion and reflection on curriculum implementations. Prior literature recommended that PLCs should emphasize acquisition methods and content knowledge (Stoll et al., 2006). Liberman et al. (2012) reviewed Shulman's (1986, 1987) discussions on knowledge acquisition, and stated that teachers' communities facilitate both the acquisition of content knowledge and PCK. They further asserted that the presence of PCK is what distinguishes a novice from an expert teacher in K-12 CS education. In the reviewed studies, many PLCs explicitly focused on strengthening teachers' CS PCK via different strategies, such as peer instruction (Cutts et al., 2017), discussion forums (Goode et al., 2020b), CS material repositories (Falkner et al., 2017) and teacher collaboration for curriculum development and implementation (Hamner et al., 2016; Price et al., 2016). A few programs used pilot lessons to achieve this goal, which could be completed with either students (Reding & Dorn, 2017) or with other educators (Roberts et al., 2018; Goode, et al., 2020b). Hamner et al. (2016) developed a unique strategy, a "tip sheet", to improve teachers' PCK and promote their CS teaching. The tip sheet was an ever-evolving list of recommendations for implementation tips from both teacher participants and researchers.

4.2. Definition of PLCs

Many PD programs used the term "PLC", but did not provide a specific definition (e.g. Borowczak & Burrows, 2019; Falkner et al., 2018). Only the ECS program (Ryoo et al., 2016) provided an explicit definition of PLC, referring to the same literature (Stoll et al., 2006, p. 223) as this study. The ECS program summarized the concept of PLC as serving three main functions: engaging educators in reflective practice, problem-solving, and expanding educators' views (Goode et al., 2019). In that study, the researchers tried to design a sustainable PD program that functioned as a PLC. Other studies used relative terms for PLC, such as teacher professional learning network (Cutts et al., 2017), instructional support network (Mazur & Woodland, 2018), and community of practice (Kosmas, 2017; Sentance & Humphreys, 2018).

Although there was a lack of a unified definition of PLC established in CS education literature, many PD programs built their PLCs based on existing theoretical frameworks. For example, Falkner et al. (2017) designed a MOOC PLC informed by Lloyd and Cochrane's model of professional learning, which provided a comprehensive framework for building a community in terms of contextualization, time, personal growth, and community. They further developed a community-centric ecosystem approach to PD addressing the sustainability and scalability required by the CS community (Falkner et al., 2018). Sentance and Humphreys (2018) built the Computing at School (CAS) PLC based on the Situated Learning Theory (Lave & Wenger, 1991). This theory explains how participants' learning in a community of practice occurs and changes as they are assimilated into that community. PLCs can apply the community of practice model to foster CS teacher learning through interaction and experience sharing with "like-minded" teachers. For example, Kosmas's (2017) PD program helped teachers gain CS expertise through assimilation in a CS teachers' online community of practice.

4.3. PLC format and approach

This study explored the formats and approaches that the CS PD programs used to build PLCs. In terms of format, the PD programs built community networks through face-to-face meetings, online activities, or a hybrid format that includes both in-person and online interactions. Among the 28 programs with PLC elements, ten programs held PLCs in a hybrid format, nine programs ran face-to-face activities, and another nine PLCs were entirely online.

Online PLCs can provide alternative or additional teacher learning and collaboration opportunities that allow flexible participation. For example, the PLAN C program (Cutts et al., 2017) built a repository of classroom-ready materials with an online forum for teachers to discuss how to use CS teaching materials available in the repository, which attracted hundreds of teachers from 32 areas in Scotland. Moreover, MOOC classes were integrated into online PLCs, offering teachers a self-paced and personalized approach for professional learning and community engagement. The CSER program (Falkner et al., 2018) attracted a large group of CS educators to their online PLC due to the comprehensive MOOC community. Meanwhile, a face-to-face PLC was favored by some PD programs. These programs valued in-person interactions and believed that online PLCs could not replicate the engagement, teachers' confidence building, and buy-in that resulted from in-person interactions. For example, Hickmott and Prieto-Rodriguez (2018) reported that face-to-face PD was particularly important for novice CS teachers and teachers with low confidence. In hybrid PLCs, face-to-face and online activities can complement each other to provide a better professional learning experience. For example, the Scalable Game Design Project (Webb et al., 2017) offered a hybrid program to accommodate those teachers who could not travel. The project also built various widgets for online discussions and provided facilitator support for teachers to implement the game-design focused CS curriculum.

This review study found that the programs featured four types of strategic activities for building PLCs, to encourage teacher collaboration and enhance community building:

Developing CS teacher leaders

Teacher leaders (lead/master teachers) have been recognized as crucial to the sustainability of CS teacher PD programs (e.g. Cutts et al., 2017; Falkner et al., 2018; Price et al., 2016). Lead teachers can facilitate high-quality discussions within the community, focusing on their own teaching practice. They can also help set up a local teacher hub in their areas to scale up PD efforts. Cutts et al. (2017) explicitly recommended investment in developing a network of subject-specific teacher leaders with expertise in PCK and conceptual frameworks for the subject as well as leadership and facilitation skills. Reding et al. (2016) used SNA techniques to identify emergent teacher leaders within a PD cohort.

Building a resource repository

Establishing a national or local central hub to share materials among teachers was one of the salient features among all PLCs. PD programs with a robust PLC tended to develop repositories for sharing lesson plans, curricular resources, and classroom implementation experience (e.g. Falkner et al., 2018).

MOOCs for self-paced learning

Many PLCs offered MOOCs to support teachers' self-paced learning and classroom implementation. For instance, Falkner et al. (2018) designed a community-centered MOOC to support CS teachers' professional learning addressing the new national curriculum in Australia. Similarly, the BJC PD program (Price et al., 2016) also created its own MOOC classes.

Online forums

PLCs use various online platforms and social media tools to organize professional forums. For instance, Falkner et al. (2017) used Google+ to support their community-centered MOOCs. The PLAN C program (Cutts et al., 2017) used Moodle as a virtual learning environment to train their lead teachers. The BJC PD program (Price et al., 2016) used Piazza to share teaching practices within the community. The Infusing Computing program (Jocius et al., 2020) utilized Slack to encourage teacher collaboration. The WeTeach_CS community (Warner et al., 2019) had Facebook and Twitter feeds to develop regular communications among teachers.

4.4. PLC evaluation: benefits and challenges

This review also examined how the studies evaluated the nature and effectiveness of PLCs for CS teacher PD. Generally, PLC evaluations were based on meeting their overall goals. We observed three key focus areas for the evaluations: (1) teachers' engagement in the PLC activities, (2) PLC's impact on teachers' pedagogy, (3) the extent of collaboration and network building among teachers. For example, the ECS program (Goode et al., 2019) and the CSER program (Falkner et al., 2018) measured teachers' engagement in PLCs by examining how actively teachers corresponded and created online discussion topics, along with the length and categories of the teachers' contributions. The ECS program (Ryoo et al., 2016) also used end-of-year surveys to evaluate how the PLC impacted teachers' inquiry-based and equity-based pedagogies. The SPARCS program (Reding & Dorn, 2017; Reding et al., 2016) developed an SNA survey to understand the extent of collaboration among teachers and the network development, based on self-reported relationships and a metrics calculation of network development. The effectiveness of PLCs was generally measured through the PD evaluation instruments, such as surveys and interviews.

Through program evaluations, the reviewed studies also discussed the benefits and effects of PLCs on CS teacher PD as well as the challenges PD providers faced. Echoing the main PLC goals, the review studies reported that PLCs brought various benefits to CS teachers. First, one of the prominent findings is that PLCs broke the isolation for CS teachers (e.g. Hamner et al., 2016; Ryoo et al., 2016). Second, PLCs played significant roles in changing teachers' attitudes and increasing their confidence in teaching CS in K-12 classrooms (Cutts et al., 2017; Ryoo et al., 2016). Third, PLCs enhanced CS content knowledge and especially PCK through community activities. The community-centered MOOCs (Falkner et al., 2017) provided an example of how a MOOC community enhanced teachers' PCK and eventually teachers were able to apply the knowledge into their classrooms. Remarkably, the MOOC community engaged more teachers in self-paced learning with enough time to personalize the curriculum and routinely engage in applying course content into their professional contexts.

There were also a few challenges PD providers faced in building PLCs for K-12 CS educators. We summarized the challenges in terms of how to (1) invite all stakeholders, (2) engage teachers with full participation, and (3) sustain the network.

Invite all stakeholders

In total, only four programs (out of 41) included (or mentioned) administrative participation in the PLC to support teacher buy-in and curriculum implementation. Ryoo et al. (2016) found that creating a cultural change within the school was not only teachers' responsibility but also the responsibility of school administrators. Mazur and Woodland (2018) concluded that organizational infrastructure to support formal and informal social interactions and access to social capital was often left unattended by school administrators (p. 17). Barriers to engaging teachers in PD and PLCs included administrative challenges when administrators did not participate or were not prioritizing CS. For example, having a CS course as an elective, teachers found themselves needing more support from the "district office" to fit the CS course within the schedule (Lee et al., 2017). Sometimes teachers became less invested in the PLC when they did not know whether they could implement what they learned from the PD into their classrooms as it was unclear what classes they would be teaching the following year (Flatland et al., 2018).

Engage teachers with full participation

Another challenge of building PLCs is that support networks did exist but contained "low density, few ties and high numbers of isolates" (Mazur & Woodland, 2018). Even though some projects broke the isolation and teacher participants were encouraged to communicate in a virtual community, they seldom contributed content. Both the BJC PD program (Price et al., 2016) and the Alice Community of Practice program (Cooper et al., 2015) reported that PD facilitators mostly fulfilled the responsibility of content contribution. Teacher participants preferred to take on the role of a consumer of the provided materials and feedback rather than to contribute actively.

Sustain the network

Sustaining PLCs can be another challenge for PD providers. Few studies explicitly discussed how to maintain PLCs. The Progra-MEER PD program (Neutens & Wyffels, 2018) acknowledged the need to create a sustainable community of practice, but did not provide evidence of establishing the community or discuss any challenges they were facing in maintaining the PLC. Falkner et al. (2017) mentioned the less frequent participation in their MOOC community forum, indicating a potential challenge for sustaining the online PLC.

5. Conclusion and discussion

This study has expanded on previous review studies on PD programs for K- 12 CS educators in terms of the scope (the diversity of the studies) and the new research focus on PLCs. Overall, this review study yielded an extensive set of findings that can inform future research and practice on building CS teacher capacity through professional development and establishing professional learning communities for K-12 CS educators. Meanwhile, it is important to note that not all the studies included details regarding the PD programs, so this review study is limited by the information available in the papers. This section highlights significant findings from this study and discusses those findings linking to the previous review study and teacher PD literature.

First, results from this study show that opportunities for CS teacher PD have grown remarkably in all aspects. In the past five years, the number of CS PD programs has almost doubled, compared to the last decade (2004–2014). Notably, the number of programs designed for elementary teachers has increased significantly, whereas previous literature discovered that there were no PD programs specially designed for elementary school teachers (Menekse, 2015). Among the reviewed studies, many PD programs in the U.S. have been preparing teachers to teach several different versions of the recently launched AP CS Principles course. Other programs have focused on either disseminating their own

CS curriculum or supporting CS integration into other subjects. These PD programs presented a collection of diverse programming languages and tools for K-12 CS education.

Second, this review has observed that recent CS PD programs embedded more features of effective PD as recommended by the literature. Effective PD is defined as "structured professional learning that results in changes in teacher practices and improvements in student learning outcomes (Darling-Hammond et al., 2017, p. 2). Drawing from extensive reviews into effective PD programs, researchers recommend that effective PD should be content-focused, collaborative, sustained, and related to learner outcomes (Cordingley et al., 2015). Darling-Hammond et al. (2017) further identified seven widely shared features of effective PD. Such PD (1) is content-focused, (2) incorporates active learning, (3) supports collaboration, (4) uses models of effective practice, (5) provides coaching and expert support, (6) offers feedback and reflection, (7) is of sustained duration. While the field of CS education is still young, recent CS PD programs present many of these characteristics of effective PD throughout their PD goals, program design, and evaluation results with some additional features specific for CS education.

In terms of program goals, CS PD programs continue to build teacher capacity and broaden participation in computing through K-12 CS education. A new goal of developing a scalable and sustainable PD model is rapidly emerging. Many programs have realized that effective and sustainable PD needs to provide teachers with ongoing support; therefore, building a professional learning community is now an essential component of PD. This goal is well-aligned with what the literature recommends for effective PD (Cordingley et al., 2015; Darling-Hammond et al., 2017). In addition, many studies have sought robust theories to organize, execute, and assess PD programs. These findings also expand on the results of the former review study by Menekse (2015), which found that many CS PD programs lacked an underpinning theoretical framework.

Results from this review also indicate new features in the area of PD evaluation. First, recent CS PD programs looked at multiple levels of information from the five critical levels of PD evaluation (Guskey, 2002). Together with data from teachers, student data were used in several programs as evidence of the success of PD programs in promoting teaching practices and fostering student learning. In contrast, the previous review study (Menekse, 2015) reported the lack of data on the effect of PD to inform program design and implementation as a critical problem in the prior studies. Meanwhile, a few programs still struggled to develop teacher capacity within the short timeframe of the PD. Moreover, the lack of administrative support was a big barrier for teachers to implement what they learned from the PD. Insufficient organizational support also presented a significant challenge for sustaining the PD. Beyond the five critical levels of PD evaluation (Guskey, 2002), many of the reviewed studies discussed three additional evaluation areas linking to their program goals: collaboration and networking, sustainability, and equity and BPC. These three areas are critical characteristics of effective PD and important goals of CS teacher PD programs. Together with the five critical levels, our computing education community can use these eight areas to guide the design and evaluation of PLC-focused, effective PD programs for CS teachers.

Another significant finding from this study is that recent CS teacher PD programs represented some "reform type of professional development" with innovative approaches to organize PD programs and build PLCs. In the prior study, Menekse (2015) reported that all the PD programs studied between 2004 and 2014 were structured as traditional forms of PD, such as summer schools and workshops, without any "reform type of professional development such as coaching or mentoring among those studies" (Menekse, 2015, p. 337). This study reveals that recent PD programs have adopted diverse approaches to structure PD, including the lead teacher model and in-classroom coaching. These strategies can be powerful in engaging teachers in active and contextualized professional learning (Darling-Hammond et al., 2017). In addition, some of the approaches increased the collaborations between higher education and local school districts, which breaks what Menekse (2015) claimed to be one of the main obstacles for sustainable PD programs. Moreover, many programs explored various strategies to foster community building and professional learning, capitalizing on the significant roles of PLCs in offering flexible, sustainable, and effective PD. These programs also incorporated several elements of effective PD, such as supporting collaboration, providing coaching and expert support, and offering feedback and reflection (Darling-Hammond et al., 2017).

Although PLCs are recommended as a popular model of effective PD (Darling-Hammond et al., 2017), PD providers can face significant challenges in building PLCs for CS teachers. Some challenges identified from the reviewed studies include inviting all stakeholders into the PLC, engaging teachers with full participation, and sustaining the PLC network. This review of recent studies on CS teacher PD suggests that an effective PLC needs to provide both "human capital" and "system infrastructure" to support and sustain CS teachers' professional learning. For system infrastructure, PLCs can be equipped with a repository for resource sharing, a platform supporting self-paced learning, and an environment encouraging collaboration and reflection among participants. For human capital, teacher leaders are essential in providing mentoring, fostering an open and productive discussion forum, and creating a local teacher hub to scale up PD efforts. It is also critical to engage all stakeholders in local PLCs to support teacher buy-in and CS curriculum implementation. Our CS education community can benefit from more research in these areas to develop sustainable PD models, including establishing PLCs for K-12 CS educators, in order to build strong CS teacher capacity.

Acknowledgments

This work was supported by the National Science Foundation under Grant No.1923461. Any opinions, findings, and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would like to thank the CS Pathways project team and the reviewers, who provided constructive feedback for an earlier version of this paper.

Disclosure statement

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

Funding

This work was supported by the National Science Foundation under Grants No.1923461.

References

Ahmed, G., Nouri, J., Zhang, L., & Noren, E. (2020). Didactic methods of integrating programming in mathematics in primary school. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 261–267). Portland, OR: ACM.

Alghamdi, F., Nylen, A., & Pears, A. (2019). Teachers' awards: An incentive for pedagogical development in Saudi Arabia. In *Proceedings of the institute of electrical and electronics engineers* (pp. 1–5). San Mateo, CA: IEEE.

Allen, L. (2018). Computer science education: online content modules and professional development for secondary teachers in West Tennessee–A case study. *I-manager's Journal of Educational Technology*, 15(2), 8–15. https://doi.org/10.1007/978-3-030-12385-7_38

Borko, H. (2004). Professional development and teacher learning: mapping the terrain. *Educational Researcher*, 33(8), 3–15. https://doi.org/10.3102/0013189X033008003

Borowczak, M., & Burrows, A. (2019). Ants go marching–integrating computer science into teacher professional development with NetLogo. *Education Sciences*, 9(1), 1–16. https://doi.org/10.3390/educsci9010066

Brennan, K. (2015). Beyond technocentrism: supporting constructionism in the classroom. *Constructivist Foundations*, 10(3), 289–296. http://constructivist.info/10/3/289 .

Brown, N. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: the resurgence of computer science in UK schools. *ACM Transactions on Computing Education*, 14(2), 1–22. https://doi.org/10.1145/2602484

Byrne, J. R., Fisher, L., & Tangney, B. (2015). Computer science teacher reactions towards raspberry pi continuing professional development (CPD) workshops using the bridge21 model. In *Proceedings of the 10th international conference on computer science & education* (pp. 267–272). Cambridge: IEEE.

Carter, L., & Crockett, C. (2018). Preparing K-12 teachers to inspire future coders: it doesn't have to be complex. In *Proceedings of the 13th international conference on computer science & education* (pp. 1–7 . San Hose: IEEE. 10.1109/FIE.2018.8658762

Celepkolu, M., O'Halloran, E., & Boyer, K. E. (2020). Upper elementary and middle grade teachers' perceptions, concerns, and goals for integrating CS into classrooms. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 965–970). Portland, OR: ACM.

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

Code.org, CSTA, & ECEP Alliance. (2020). 2020 State of computer science education: Illuminating disparities. Retrieved from October 14, 2020. https://advocacy.code.org/stateofcs

College Board. (2020). AP computer science principles: the course. Retrieved from September 19, 2020. https://apcentral.collegeboard.org/courses/ap-computer-science-a/course

Cooper, S., Rodger, S., Schep, M., Stalvey, R. H., & Dann, W. (2015). Growing a K-12 community of practice. In *Proceedings of the 46th ACM technical symposium on computer science education* (pp. 290–295). Kansas City, MO: ACM.

Cordingley, P., Higgins, S., Greany, T., Buckler, N., Coles-Jordan, D., Crisp, B., Saunders, L., & Coe, R. (2015). *Developing Great Teaching: Lessons from the international reviews into effective professional development.* London: Teacher Development Trust.http://tdtrust.org/about/dgt

CSforAll. (2020). *CSforAll commitments*. Retrieved from November 17, 2020. https://www.summit.csforall.org/commitments

Cutts, Q., Robertson, J., Donaldson, P., & O'Donnell, L. (2017). An evaluation of a professional learning network for computer science teachers. *Computer Science Education*, 27(1), 30–53. https://doi.org/10.1080/08993408.2017.1315958

Darling-Hammond, L., Hyler, M. E., & Gardner, M. (2017). *Effective Teacher Professional Development*. Learning Policy Institute.

Falkner, K., Vivian, R., Falkner, N., & Williams, S. A. (2017). Reflecting on three offerings of a community-centric MOOC for K-6 computer science teachers. In *Proceedings of the 48th ACM technical symposium on computer science education* (pp. 195–200). Seattle, Washington: ACM. DOI: 10.1145/3017680.3017712.

Falkner, K., Vivian, R., & Williams, S. A. (2018). An ecosystem approach to teacher professional development within computer science. *Computer Science Education*, 28(4), 303–344. https://doi.org/10.1080/08993408.2018.1522858

Flatland, R., Goldstein, I., Egan, M. L., Vandenerg, S., Fryling, M., & Small, S. (2018). Connecting colleges/universities and local high schools: A new model for high school CS teacher development. In *Proceedings of the 49th ACM technical symposium on computer science education* (pp. 958–963). Baltimore, MD: ACM. DOI: https://dl.acm.org/doi/10.1145/3159450.3159570 .

Franke, B., Century, J., Lach, M., Wilson, C., Guzdial, M., Chapman, G., & Astrachan, O. (2013). Expanding access to K-12 computer science education: research on the landscape of computer science professional development. In *Proceedings of the 44th ACM technical symposium on computer science education* (pp. 541–542). Denver, CO: ACM.

Gallup & Google. (2020). *Current perspectives and continuing challenges in computer science education in U.S. K-12 schools*. Retrieved from November 16, 2020. https://csedu.gallup.com/home.aspx

Garrison, D. R., Anderson, T., & Archer, W. (2000). Critical inquiry in a text-based environment: computer conferencing in higher education model. *The Internet and Higher Education*, 2(2–3), 87–105. https://doi.org/10.1016/S1096-7516(00)00016-6

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

Goode, J., Peterson, K., Malyn-Smith, J., & Chapman, G. (2019). Online professional development for high school teachers: building the capacity of a national cadre of teachers to broaden participation in computing. In *Proceedings of the institute of electrical and electronics engineers* (pp. 1–6). San Mateo, CA: IEEE.

Goode, J., Peterson, K., Malyn-Smith, J., & Chapman, G. (2020b). Online professional development for high school computer science teachers: features that support an equity-based professional learning community. *Computing in Science & Engineering*, 22(5), 51–59. https://doi.org/10.1109/MCSE.2020.2989622

Granor, N., Delyser, L. A., & Wang, K. (2016). TEALS: teacher professional development using industry volunteers. In *Proceedings of the 47th ACM technical symposium on computer science education* (pp. 60–65). Memphis, TN: ACM.

Gray, J., Corley, J., & Eddy, B. P. (2016). An experience report assessing a professional development MOOC for CS principles. In *Proceedings of the 47th ACM technical symposium on computing science education* (pp. 455–460). Memphis, TN: ACM.

Gray, J., Haynie, K., Packman, S., Boehm, M., Crawford, C., & Muralidhar, D. (2015). A mid-project report on a statewide professional development model for CS principles. In *Proceedings of the 46th ACM technical symposium on computer science education* (pp. 380–385). Kansas City, MO: ACM.

Guskey, T. (2002). Does it make a difference? evaluating professional development. *Educational Leadership*, 59(6), 45–51. https://uknowledge.uky.edu/edp_facpub/7 .

Haduong, P., & Brennan, K. (2019). Helping K-12 teachers get unstuck with scratch: the design of an online professional learning experience. In *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 1095–1101). Minneapolis, MN: ACM.

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). Baltimore, MD: ACM.

Hamner, E., Cross, J., Zito, L., Bernstein, D., & Mutch-Jones, K. (2016). Training teachers to integrate engineering into non-technical middle school curriculum. In *Proceedings of the institute of electrical and electronics engineers* (pp. 1–9). Portland: IEEE.

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

Hu, H. H., Heiner, C., Gagne, T., & Lyman, C. (2017). Building a statewide computer science teacher pipeline. In *Proceedings of the 48th ACM technical symposium on computer science education* (pp. 291–296). Seattle, Washington: ACM.

Jocius, R., Joshi, D., Dong, Y., Robinson, R., Catete, V., Barnes, T., Albert, J., Andrews, A., & Lytle, N. (2020). Code, connect, create: the 3C professional development model to support computational thinking infusion. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 971–977). Portland, OR: ACM.

Kania, J., & Kramer, M. (2011). Collective impact. *Stanford Social Innovation Review*, (Winter) , 36–41. https://senate.humboldt.edu/sites/default/files/senate/Chair%20Written%20Report%201-23-2018.pdf .

Kao, Y., Nolan, I., & Rothman, A. (2020). Project scoring for program evaluation and teacher professional development. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 1133–1138). Portland, OR: ACM.

Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, J. R. (2020). 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

Kosmas, P. (2017). Online sharing of knowledge among in-service teachers for professional development purposes. In *Proceedings at the 9th international conference on education technology and computers* (pp. 162–166). Barcelona: ICETC.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge university press.

Lee, I. A., Dombrowski, M. P., & Angel, E. (2017). Preparing STEM teachers to offer New Mexico computer science for all. In *Proceedings of the 48th ACM technical symposium on computer science education* (pp. 363–368). Seattle, Washington: ACM.

Leyzberg, D., & Moretti, C. (2017). Teaching CS to CS teachers: addressing the need for advanced content in K-12 professional development. In *Proceedings of the 48th ACM technical symposium on computer science education* (pp. 369–374). Seattle, Washington: ACM.

Liberman, N., Kolikant, Y. B. D., & Beeri, C. (2012). “Regressed experts” as a new state in teachers’ professional development: Lessons from computer science teachers’ adjustments to substantial changes in the curriculum. *Computer Science Education*, 22(3), 257–283. https://doi.org/10.1080/08993408.2012.721663

Liu, J., Hasson, E. P., Barnett, Z. D., & Zhang, P. (2011). A survey on computer scienceK-12 outreach: teacher training programs. In *Proceedings of the 2011 frontiers in education conference* (pp. T4F-1-6). Aizu: ICIET.

Lloyd, M., & Cochrane, J. (2006). Celtic knots: interweaving the elements of effective teacher professional development in ICT. *Australian Educational Computing*, 21(2), 16–19 https://search.informit.org/doi/10.3316/aeipt.159908 .

Mazur, R., & Woodland, R. H. (2018). A fringe topic in a fragile network: how digital literacy and computer science instruction is supported (or not) by teacher ties. *ACM Transactions on Computing Education*, 18(4), 1–20. https://doi.org/10.1145/3218361

Mclaughlin, M. W., & Talbert, J. E. (2006). *Building school-based teacher learning communities: Professional strategies to improve student achievement*. Teachers College Press.

Menekse, M. (2015). Computer science teacher professional development in the United States: A review of studies published between 2004 and 2014. *Computer Science Education*, 25(4), 325–350. https://doi.org/10.1080/08993408.2015.1111645

Morelli, R., Uche, C., Lake, P., & Baldwin, L. (2015). Analyzing year one of a CS principles PD project. In *Proceedings of the 46th ACM technical symposium on computer science education* (pp. 368–373). Kansas City, MO: ACM.

Mouza, C., Czik, A., Little, A., Coffey, D., & Buttram, J. (2017). From professional development to the classroom: findings from CS K-12 teachers. In *Proceedings of the 48th ACM technical symposium on computer science education* (pp. 477–482). Seattle, WA: ACM. DOI: 10.1145/3017680.3017739.

Mouza, C., Pollock, L., Pusecker, K., Guidry, K., Yeh, C., 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 computer science education* (pp. 66–71). Memphis, TN: ACM. DOI: 10.1145/2839509.2844585.

Nakajima, T. M., & Goode, J. (2019). Transformative learning for computer science teachers: examining how educators learn e-textiles in professional development. *Teaching and Teacher Education*, 85, 148–159. https://doi.org/10.1016/j.tate.2019.05.004

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). Baltimore, MD: ACM. DOI: https://dl.acm.org/doi/10.1145/3159450.3159568 .

Ni, L., Guzdial, M., Tew, A. E., Morrison, B., & Galanos, R. (2011). Building a community to support HS CS teachers: the disciplinary commons for computing educators. In *Proceedings of the 42nd ACM technical symposium on computer science education* (pp. 553–558). Dallas, TX: ACM. https://doi.org/10.1145/1953163.1953319 .

Pollock, L., Mouza, C., Czik, A., Little, A., Coffey, D., & Buttram, J. (2017). From professional development to the classroom: findings from CS K-12 teachers. In *Proceedings of the 2017 ACM technical symposium on computer science education* (pp. 477–482). Seattle, Washington: ACM. DOI: https://dl.acm.org/doi/abs/10.1145/3017680.3017739 .

Price, T. W., Catete, V., Albert, J., Barnes, T., & Garcia, D. D. (2016). Lessons learned from "BJC" CS principles professional development. In *Proceedings of the 47th ACM technical symposium on computer science education* (pp. 467–472). Memphis, TN: ACM. DOI: 10.1145/2839509.2844625.

Ravitz, J., Stephenson, C., Parker, K., & Blazevski, J. (2017). Early lessons from evaluation of computer science teacher professional development in google's cs4hs program. *ACM Transactions on Computing Education*, 17(4), 1–16. https://doi.org/10.1145/3077617

Reding, T. E., & Dorn, B. (2017). Understanding the "teacher experience" in primary and secondary CS professional development. In *Proceedings of the 48th ACM technical symposium on computer science education* (pp. 155–163). Seattle, Washington: ACM. DOI: https://doi.org/10.1145/3105726.3106185 .

Reding, T. E., Dorn, B., Grandgenett, N., Siy, H., Youn, J., Zhu, Q., & Engelmann, C. (2016). Identification of the emergent leaders within a CSE professional development program. In *Proceedings of the 47th ACM technical symposium on computer science education* (pp. 37–44). Memphis, TN: ACM. DOI: https://dl.acm.org/doi/10.1145/2978249.2978254 .

Reimer, Y. J., Coe, M., Blank, L. M., & Braum, J. (2018). Effects of professional development on programming knowledge and self-efficacy. In *Proceedings of the 13th international conference on computer science & education* (pp. 1–8). San Jose: IEEE.

Roberts, M., Prottsman, K., & Gray, J. (2018). Priming the pump: reflections on training K-5 teachers in computer science. In *Proceedings of the 49th ACM technical symposium on computer science education* (pp. 723–728). Baltimore, MD: ACM.

Rosato, J., Lucarelli, C., Beckworth, C., & Morelli, R. (2017). A comparison of online and hybrid professional development for CS principles teachers. In *Proceedings for the 22nd annual conference on innovation and technology in computer science education* (pp. 140–145). Bologna: ITiCSE.

Ryoo, J., Goode, J., & Margolis, J. (2016). It takes a village: Supporting inquiry- and equity- oriented computer science pedagogy through a professional learning community. *Computer Science Education*, 25(4), 351–370. https://doi.org/10.1080/08993408.2015.1130952

Sentance, S., Humphreys, S., & Dorling, M. (2014). The network of teaching excellence in computer science and master teachers. In *Proceedings for the 9th workshop in primary and secondary computing education* (pp. 80–88). Berlin: WiPSCE.

Sentance, S., & Humphreys, S. (2018). Understanding professional learning for Computing teachers from the perspective of situated learning. *Computer Science Education*, 28(4), 345–370. https://doi.org/10.1080/08993408.2018.1525233

Shulman, L. S. (1986). Those who understand: knowledge growth in teaching. *Educational Researcher*, 15(2), 4–14. https://doi.org/10.3102/0013189X015002004

Shulman, L. S. (1987). Knowledge and teaching: foundations of the new reform. *Harvard Educational Review*, 57(1), 1–22. https://doi.org/10.17763/haer.57.1.j463w79r56455411

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 *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 1143–1149). Minneapolis, MN: ACM. DOI: https://dl.acm.org/doi/10.1145/3287324.3287456 .

Stoll, L., Bolam, R., McMahon, A., Wallace, M., & Thomas, S. (2006). Professional learning communities: A review of the literature. *Journal of Educational Change*, 7(4), 221–258. https://doi.org/10.1007/s10833-006-0001-8

Thomas, L., & Autio, E. (2014). The fifth facet: the ecosystem as an organizational field. *Academy of Management Proceedings*, 1(2014), 10306. https://doi.org/10.5465/ambpp.2014.10306abstract

Vescio, V., Ross, D., & Adams, A. (2008). A review of research on the impact of professional learning communities on teaching practice and student learning. *Teaching and Teacher Education*, 24(1), 80–91. https://doi.org/10.1016/j.tate.2007.01.004

Vygotsky, L. (1978). Interaction between Learning and Development. In Cole, M, John-Steiner, V, Scribner, S, and Souberman, E (Eds.,),*Mind and Society: The Development of Higher Psychological Processes*. Cambridge, MA: Harvard University Press, 79–91.

Warner, J. R., Torbey, R., Fletcher, C. L., & Garbrecht, L. S. (2019). Increasing capacity for computer science education in rural areas through a large-scale collective impact model. In *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 1157–1163). Minneapolis, MN: ACM.

Webb, D. C., Nickerson, H., & Bush, J. B. (2017). A comparative analysis of online and face- to-face professional development models for CS education. In *Proceedings of the 48th ACM technical symposium on computer science education* (pp. 621–626). Seattle, Washington: ACM.

Woodland, R. H. (2016). Evaluating PK–12 professional learning communities. *American Journal of Evaluation*, 37(4), 505–521. https://doi.org/10.1177/1098214016634203

Yadav, A. (2017). Computer science teacher professional development: towards a research agenda on teacher thinking and learning. In *Proceedings for the 12th workshop in primary and secondary computing education* (pp. 1–2). Nijmegen: WiPSCE. DOI: https://doi.org/10.1145/3137065.3137066 .

Yadav, A., Krist, C., Good, J., & Caeli, E. N. (2018). Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education*, 28(4), 371–400. https://doi.org/10.1080/08993408.2018.1560550

Appendix

Table A1. Summary of reviewed studies.

	PD Program	Study	Grade Level	Curriculum	PLC	Country
1	Mobile Computer Science Principles (CSP) PD	1a. Morelli et al. (2015) 1b. Rosato et al. (2017)	High School	Mobile CSP	Contain PLC components	US
2	Raspberry Pi Continuing PD	Byrne et al. (2015)	Secondary School	Raspberry Pi curriculum	Not Referenced	Ireland
3	Alice Community of Practice	Cooper et al. (2015)	Middle & High School	Alice programming curriculum	Focus on PLC building	US
4	CS4 Alabama	4a. Gray et al. (2015) 4b. Gray et al. (2016)	High School	CS4HS MOOC	Focus on PLC building	US
5	Exploring Computer Science (ECS) PD	5a. Ryoo et al. (2016) 5b. Goode et al. (2019) 5 c. Goode et al. (2020a) 5d. Goode et al. (2020b) 5e. Nakajima and Goode (2019)	High School	ECS curriculum	Focus on PLC building	US
6	Technology Education and Literacy in Schools (TEALS)	Granor et al. (2016)	High School	AP CS A or subset of the BJC curriculum	Contain PLC components	US
7	Strategic Problem-based Approach to Rouse Computer Science (SPARCS)	7a. Reding et al. (2016) 7b. Reding and Dorn (2017)	Middle & High School	CS infusion & problem-based learning	Focus on PLC building	US
8	Beauty and Joy of Computing (BJC) PD	Price et al. (2016)	Elementary, Middle & High school	BJC curriculum	Focus on PLC building	US
9	Arts & Bots Math and Science Partnership	Hamner et al. (2016)	Middle & High School	Robotics projects integrated into traditional disciplines	Contain PLC components	US
10	Partner4CS PD	Mouza et al. (2016)	Middle & High School	AP CS Principles & CS-STEM infusion	Contain PLC components	US
11	Professional Learning and Networking in Computing (PLANC)	Cutts et al. (2017)	Secondary School	Not Specified	Focus on PLC building	Scotland
12	Scalable Game Design Project	Webb et al. (2017)	Elementary, Middle & High school	Scalable Game Design curriculum	Contain PLC components	US
13	An Online Community of Practice for PD	Kosmas (2017)	Primary & Secondary School	Not Specified	Focus on PLC building	Cyprus
14	Community-centric MOOCs for PD	Falkner et al. (2017)	Elementary School	Australian Curriculum: Digital Technologies (CS)	Focus on PLC building	Australia
15	Utah Exploring Computer Science Initiative	Hu et al. (2017)	Middle School	ECS curriculum	Contain PLC components	US
16	A week-long summer workshop	Leyzberg and Moretti (2017)	High School	AP CS curricula	Not Referenced	US

(*Continued*)

Table A1. (Continued).

	PD Program	Study	Grade Level	Curriculum	PLC	Country
17	Delaware Partner4CS	Pollock et al. (2017)	Elementary, Middle & High school	CS-STEM integration & AP CS Principles	Propose to build PLC	US
18	Google CS4HS	Ravitz et al. (2017)	High School	Four programs: 2 with face-to-face course, 1 online, 1 hybrid	Focus on PLC building	US
19	New Mexico Computer Science for All (NM-CSforAll)	Lee et al. (2017)	High School	NMCSforAll curriculum	Focus on PLC building	US
20	CSER Digital Technologies	Falkner et al. (2018)	Elementary & Middle School	Australian Curriculum: Digital Technologies (CS)	Focus on PLC building	Australia
21	Computing at School (CAS)	Sentance and Humphreys (2018)	Primary & Secondary School	English national curriculum for computing	Focus on PLC building	UK
22	Online Content Modules: CS in Middle Grades	Allen (2018)	Middle School	Five middle school CS content modules	Not Referenced	US
23	Constructionist Learning PD	Hickmott and Prieto-Rodriguez (2018)	Primary & High School	Australian Digital Technologies curriculum	Propose to build PLC	Australia
24	Embedded CS PD with college faculty into high schools	Flatland et al. (2018)	High School	College-level introductory CS course	Focus on PLC building	US
25	A summer institute on CS Principles	Hamlen et al. (2018)	Middle School	AP CS Principles	Focus on PLC building	US
26	Progra-Meer PD	Neutens and Wyffels (2018)	Secondary School	CS integration into STEM	Contain PLC components	Belgium
27	Code.org CS Fundamentals PD	Roberts et al. (2018)	Elementary School	CS Fundamentals curriculum	Contain PLC components	US
28	Joy and Beauty of Computing (JBC) PD	Reimer et al. (2018)	High School	JBC curriculum	Contain PLC components	US
29	District digital literacy and computer science (DLCS) instructional support network	Mazur and Woodland (2018)	Elementary School	Integrating DLCS	Focus on PLC building	US
30	PD for integrating computational thinking (CT)	Yadav et al. (2018)	Elementary School	CT-infused inquiry lessons	Not Referenced	US
31	Getting Unstuck online Scratch PD	Haduong and Brennan (2019)	Elementary, Middle & High school	Scratch projects	Contain PLC components	US
32	PD connected with the Educational Excellence Award	Alghamdi et al. (2019)	Elementary, Middle & High school	Not Specified	Not Referenced	Saudi
	Arabia Incorporating coding	33	Introduction to Coding workshop	Carter and Crockett, 2018)	Elementary, Middle & High school	Not
	Referenced into K-12 curriculum	US				
34	Robotics, Applied Mathematics, Physics, and Engineering Design (RAMPED) PD	Borowczak and Burrows (2019)	Elementary, Middle & High school	CS with real world problems	Propose to build PLC	US

(*Continued*)

Table A1. (Continued).

	PD Program	Study	Grade Level	Curriculum	PLC	Country
35	Workshop for introducing CT in rural areas	Simmonds et al. (2019)	Elementary & Middle School	Introducing CT with Scratch	Contain PLC components	Chile
36	WeTeach_CS PD	Warner et al. (2019)	High School	Not Specified	Focus on PLC building	US
37	5-Day PD workshop	Celepkolu et al. (2020)	Elementary & Middle School	Integrating CS using Snap!	Not Referenced	US
38	Infusing Computing PD	Jocius et al. (2020)	Middle & High School	CT integration	Contain PLC components	US
39	Science Teaching Inquiry Group in Computational Thinking (STIGCT)	Ketelhut et al. (2020)	Elementary School	Integrating CT into elementary science	Not Referenced	US
40	Developing teaching with programming content supported by Lesson Study	Ahmed et al. (2020)	Primary School	Integrating programming in mathematics	Not Referenced	Sweden
41	PD with student project scoring	Kao et al. (2020)	Middle School	Not Specified	Not Referenced	US