Technology, Knowledge and Learning (2025) 30:2149–2173
https://doi.org/10.1007/s10758-025-09886-y

ORIGINAL RESEARCH

Moving Through the Messy Middle: Longitudinal Case Studies of Teachers' Computational Thinking Infusion

Robin Jocius[1] · Deepti Joshi[2] · Melanie Blanton[3] · Jennifer Albert[3] · W. Ian O'Byrne[4]

Received: 16 September 2024 / Revised: 25 May 2025 / Accepted: 17 July 2025 /
Published online: 12 August 2025

Abstract
TPACK (technological pedagogical content knowledge) has been widely used as a framework for understanding and creating teacher professional development experiences with technology, including teacher learning about programming and computational thinking (CT), which refers to the set of problem-solving practices inherent to the computer science discipline. However, one persistent gap in the knowledge base is the need for research that examines teachers' development of new pedagogical practices over an extended period. This study contributes a longitudinal perspective on teacher learning about computational thinking in relation to three TPACK domains: pedagogical content knowledge, technological content knowledge, and technological pedagogical content knowledge. Qualitative data from a five-year study of teachers' CT integration, including interviews, surveys, programming products, pedagogical artifacts, and practitioner research projects, were analyzed to unpack how two case study teachers (secondary math and science) came to understand CT and coding concepts, as well as how they utilized CT-infused teaching practices in their classrooms. Findings suggest that as teachers developed clear understandings of CT and more sophisticated knowledge of programming concepts, they were able to design scaffolded and sequential CT learning experiences for students and colleagues.

Keywords Computational thinking · Teacher learning · Computer science education · Professional development

1 Introduction

Over the past two decades, as many K-12 schools have shifted to one-to-one device models and new technological tools have made more personalized and responsive learning possible, there has been growing interest in teachers' professional learning experiences related to technology-based practices (Harris et al., 2009; Sauers & McLeod, 2018). More recently, as COVID-19 shuttered schools across the globe, there was a rapid acceleration in technology use that has fundamentally changed how we think about teaching and learning with technology (Ferdig et al., 2020). To ground understandings of how teachers grapple with new

Extended author information available on the last page of the article

and yet-unknown technologies, like programming environments, generative AI (artificial intelligence) tools, and virtual reality, Koehler and Mishra (2008) proposed the TPACK (technological pedagogical content knowledge) framework, which focuses on the different components of knowledge teachers need to teach with technology. Specifically, for educators interested in computational thinking (CT), or the set of problem-solving practices that relates to disciplinary practices in computer science (Wing, 2006), TPACK has served as a theoretical framework and analytic lens to develop effective professional learning experiences for teachers (Kong & Lai, 2021; Mouza et al., 2017).

Despite increasing attention to teacher learning and use of technology for pedagogical purposes, particularly considering new advances in online and hybrid teaching and generative AI tools (Whalen & Mouza, 2023), one persistent gap in the knowledge base is the need for research that examines teachers' development of new pedagogical practices over an extended period of time (Rodriguez Moreno et al., 2019). A longitudinal approach allows researchers to trace how teachers come to know a new practice, how they implement the practice with students, and how they transform that practice in light of their own experiences and perspectives (Hofer & Grandgennet, 2012). Longitudinal analyses are rare in the teacher education literature for a variety of reasons–schools and districts tend to offer teacher learning programs that are often brief; researchers often have limited access to teachers at the end of research projects; and teachers are leaving the profession at high rates, which can restrict access to long-term follow-up studies (Mouza et al., 2009).

This article contributes a longitudinal perspective on teacher learning about CT and the adoption of new pedagogical practices over time. In examining data from a five-year study, we unpack how two content area teachers (math and science) came to understand CT and how they gradually adopted CT-infused teaching practices. Our work addresses three research questions:

- Pedagogical Content Knowledge: How does teachers' understanding of computational thinking infusion change over time?
- Technological Content Knowledge: How does teachers' understanding of coding concepts and practices change over time?
- Technological Pedagogical Content Knowledge: How does teachers' classroom implementation of computational thinking infusion change over time?

Qualitative case study methods were used to analyze teacher surveys, interviews, reflections, pedagogical artifacts, and CT infusion projects. After examining the two cases and providing a cross-case comparison, we discuss implications for teacher professional development (PD) on CT, as well as the need for more longitudinal work on teacher technology learning more broadly.

2 Theoretical Framework

This study is grounded in a constructivist framing of the TPACK model (Olofson et al., 2016), which was initially conceived as TPCK (Mishler & Koehler, 2007). Drawing from Shulman's (1987) conceptualization of pedagogical content knowledge, which describes an amalgam of knowledge bases that teachers draw from to teach effectively, Koehler and

Mishra (2008) argued that successful technology integration requires more than just technical skills or subject matter expertise. They described a complex interaction between three core knowledge domains: technology knowledge (TK), pedagogical knowledge (PK), and content knowledge (CK). In Koehler and Mishra's (2008) revised TPACK model, there are seven components in total: content knowledge (CK), technological knowledge (TK), pedagogical knowledge (PK), pedagogical content knowledge (PCK), technological content knowledge (TCK), technological pedagogical knowledge (TPK), and technological pedagogical content knowledge (TPACK). TPACK represents the intersection of these domains; Koehler and Mishra (2008) emphasize the need for teachers to develop a nuanced understanding of how technology can enhance teaching and learning within specific content areas.

TPACK has been used to document and unpack teacher learning in a variety of contexts (Yeh et al., 2021), including pre-service teacher education (Chai et al., 2011; Lachner et al., 2021), graduate coursework for in-service teachers (Shin et al., 2009), and in-service teacher PD (Koh et al., 2017). In an effort to foreground the interactional components of technology integration, Olofson et al. (2016) reconceptualized TPACK as "an active process carried out by the teacher in which s/he constructs knowledge for teaching in the technology-rich setting" (p. 189). This framing recognizes that teachers' abilities to use new technologies, pedagogies, and content is relational and dependent on context. Others have used constructivist TPACK framings in order to examine teachers' perceptions of academic technologies (Koh et al., 2014) and beliefs (Dong, 2015), as well as the influence of social and contextual factors (Lai et al., 2022) on their technology integration practices. For the purposes of this study, we take up the constructivist framing to unpack how in-service teachers' TPACK changed over time as a result of ongoing PD and support.

Researchers have documented how in-service teacher PD and how the TPACK framework can ground studies in which teachers learn how to integrate technology, STEM content, and design thinking (Chai, 2019, p. 12). Research on TPACK and STEM PD has highlighted its potential for supporting instructional decision-making (Dalal et al., 2017), developing personalized learning PD (Chaipidech et al., 2021), and evaluating curriculum (Pareto & Willermark, 2019). While these studies suggest that TPACK can guide PD design and help teachers make decisions about technology-enhanced pedagogies, others caution that it is not a comprehensive PD program. Koh (2017), for example, argues that TPACK PD should be supplemented with activities to support teacher self-efficacy and collaboration. Similarly, Jaipal-Jamani and Figg (2015) found that teachers need professional learning experiences that introduce theoretical knowledge and opportunities to engage in classroom experimentation and reflection to employ TPACK in their classrooms.

It's important to note that there have been numerous critiques of and revisions to TPACK since its introduction. One of the major critiques is that by parsing out teacher learning into seven categories, the model is too convoluted for use in both practice and practice (Archambault & Barnett, 2010). Brantley-Dias and Ertmer (2013), for example, suggest that the model is both too "vague and too intricate" and that researchers, teacher educators, and teachers might choose to focus specifically on the components that best match their goals (p. 123). Further, in a critique specific to the ways in which researchers have employed TPACK, Saubern et al. (2020) suggested that studies have tended to focus on validating individual components rather than examining "holistically how the program helped teachers use technology more effectively to achieve teaching and learning goals" (p. 5).

So, for the purposes of this study, we specifically examine three elements (PCK, TCK, and TPACK) of the TPACK model in connection with teacher learning about CT infusion. Focusing on these particular components enabled us to better trace teacher learning across time and through multiple data sources. We have also carefully designed our study to analyze the interactions among TPACK elements, particularly in regard to teachers' implementation of new pedagogical practices over time.

3 Review of Related Literature

3.1 Computational Thinking Infusion in K-12 Education

Across the United States, national initiatives such as CSforAll (Wang, 2017) have aimed to broaden participation in computer science in the P-12 grades. As a result of greater attention to CS education, many states have adopted computer science standards for high, middle, and elementary school students (Code.org, CSTA, & ECEP Alliance, 2022). A critical component of these standards, and the foundation of CS education, is computational thinking (CT), which is an approach to problem-solving that builds specific knowledge and skills that can be utilized in both plugged-in and unplugged contexts (Shute et al., 2017; Wing, 2006). CT involves the development of content knowledge (pattern recognition, abstraction, decomposition, and algorithmic design), skills (debugging, remixing, and collaboration), and dispositions (perseverance, response to ambiguity, and creativity) (Dong et al., 2019; CSTA & ISTE, 2011).

Though CS education is a relatively new field, the cognitive framings of CS have been well established, and the social and cultural framings are expanding (Jacobs & Warschauer, 2018; Kafai et al., 2020). In an effort to encourage earlier and diverse experiences with CS and CT for all students, some states have adopted K-5 or K-8 Computer Science standards that are meant to be applied alongside disciplinary content in an integrated way. CT has transdisciplinary applications and, when taught in concert with content area objectives, can strengthen student understanding of disciplinary concepts (Grover & Pea, 2018; Jocius et al., 2022; Wing, 2006). In addition, CT concepts are woven within the Next Generation Science Standards (NGSS) and many states' math standards (NGSS Lead States, 2012; Weintrop et al., 2016).

Research has begun to explore the potential benefits of embedding CT concepts and practices into non-STEM content areas, including English Language Arts, social studies, art, world language, and dance (Jocius et al., 2021; Li et al., 2020; Yadav et al., 2016). Research suggests that CT integration, or infusion, can broaden participation and access to foundational computer science skills (Yadav et al., 2017), develop problem-solving and critical thinking skills (Burke et al., 2016), prepare students for participation and content creation in a digital world (Kafai et al., 2020), and support students' critical computational literacy skills (Hutchinson et. al., 2016; Kafai & Proctor, 2022). One challenge has been designing and developing professional learning supports for teachers so that they can implement CT infusion in their classrooms.

3.2 Professional Development on CT Infusion

Researchers (Hestness et al., 2018; Ketelhut et al., 2020; Li et al., 2020; Rich et al., 2021) have called for studies of teacher PD to identify models and best practices that can support successful CT integration. Initial studies have identified several features of PD that have shown promise in helping teachers develop self-efficacy in teaching CT-infused lessons: examining CT frameworks and modeling CT-infused learning practices (Hestness et al., 2018), engaging in discipline-specific tasks that connect CT to existing content standards (Blanton et al., 2023; Ketelhut et al., 2020), participating in a community of teachers and experts in CT integration (Caskurlu et al., 2021), and collaborative integration planning and lesson co-design (Biddy et al., 2021). Researchers have also pointed to the potential of sustained PD efforts that position teachers as co-designers of learning experiences. For example, in a study of CT infusion into upper elementary inquiry-based science lessons, Yadav et al. (2018) found that collaborative lesson planning and intensive PD increased teacher knowledge of CT skills and enabled teachers to seek CT "glimmers" as they integrated CT into appropriate lessons and activities (p. 377).

As Kong et al. (2023) argue, a critical next step for building sustainable professional learning experiences about CT is to investigate mechanisms for providing long-term teacher support to develop teacher knowledge over time and at scale. Others (Mouza et al., 2017) have called for more longitudinal work that attends to CT infusion across content areas and teaching contexts. Our work builds upon the existing research on TPACK, CT infusion, and CT PD to examine the development of teacher knowledge over time, in multiple content areas, and across multiple forms of knowledge (PCK, TCK, and TPACK). In their seminal study on teachers' learning to integrate technology-based practices, Harris and Hofer (2011) recommend centering the disciplinary curriculum, with the digital tools serving as a secondary consideration. We take a slightly different view in that we see CT (with or without the use of digital tools) as a thinking practice that can strengthen both teacher and student knowledge of disciplinary content. Specifically, our study is grounded in the premise that teachers need to experience CT infusion to think computationally using both unplugged and plugged methods. By strengthening a teacher's ability to think computationally and understand CT knowledge, skills, and dispositions, they can better make disciplinary connections and see disciplinary applications. Thus, grounded in the TPACK framework, we situated computational thinking at the forefront of professional learning experiences and aimed to support teachers' discipline-specific development through a CT lens.

4 Method

4.1 Context

Infusing Computing was a five-year grant project designed to support 360 middle and high school teachers from two Southeastern states as they learned to integrate CT into their classrooms (Jocius et al., 2022). Teachers represented various disciplines (math, science, ELA, social studies, and related arts) and could choose to attend the PD as individuals or part of school-based teams. The overarching aim of Infusing Computing was to broaden participation in computing by building a community of practice of teachers interested in CT and CS

infusion (Lave & Wenger, 1991). In terms of defining CT, we utilized the PRADA model (pattern recognition, abstraction, decomposition, and algorithms), which is a mnemonic device to make specific CT concepts more understandable for both teachers and students (Dong et al., 2019). From 2017–2021, in summer workshops and monthly academic-year activities, Infusing Computing teacher partners connected CT to disciplinary standards and teaching practices, created CT-infused lessons and units, and documented how they used and adapted lessons in their schools and classrooms. In the final year of the project (2021–2022), teachers applied for funds of 5,000-10,000 to implement their own CT infusion projects in their classrooms, schools, or communities (see Fig.1).

4.2 Participants and Case Selection

The goal of this qualitative study was to learn about changes in teachers' TPACK for CT infusion over time. A descriptive multiple case study methodology was chosen for the following reasons: the context of the Infusing Computing PD was of critical importance to understanding participants' experiences, the aim of the study was to address how and why questions about current classroom practices, and the phenomenon (CT infusion) and the context were inextricably linked (Yin, 2009). In order to trace teacher learning across the five years of the project, we decided to select cases from the group of teachers (n = 18) who participated in the final summer of Infusing Computing PD and received teacher-partner funds. Of the 18 teachers who completed practitioner inquiry projects, 13 had participated in at least three of the four summer PDs, and 7 had participated in all four summer sessions. Using purposive selection (Merriam, 1998), we chose two teachers who had participated in at least three summer PDs, successfully completed practitioner inquiry projects, represented different content area backgrounds and grade levels (middle and high school), and had varying levels of previous experience with CT prior to attending the PD (see Table 1). Ana (all teacher names are pseudonyms) attended four summer PD sessions (2018, 2019, 2020, 2021) and Charlotte attended three summer PD sessions (2019, 2020, 2021).

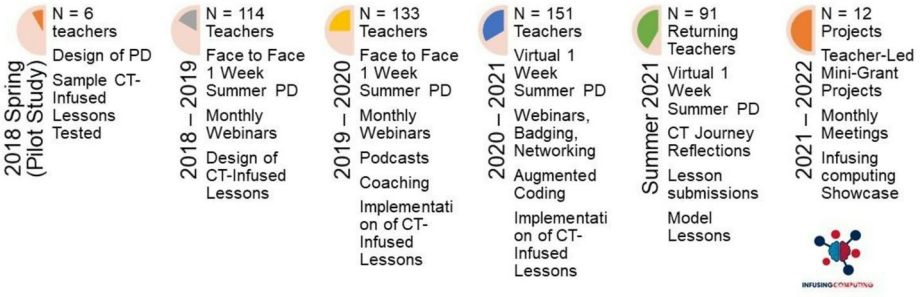

Fig. 1 Overview of infusing computing project

Table 1 Overview of case study participants

Name	Ana	Charlotte
Content Area and Teaching Placement	8th grade math	High school science (2019) Instructional Coach (2020–2022)
Self-Identified Race, Ethnicity, and Gender	White Hispanic female	White female
Years of Classroom Teaching Experience when Beginning Project (2018 and 2019)	11	24
Years of Classroom Teaching Experience at End of Study (2022)	15	27
Education	B.A., Communications	B.S., Secondary Education
School Context	STEAM focus	Computational Thinking focus
Previous CT Experience	None	Prior experience with coding in Scratch and limited experience with Python

4.3 Data Sources

4.3.1 Pre-PD and Post-PD Surveys (2018, 2019, 2020, 2021)

Pre-PD surveys were administered prior to summer PD each year and were used to gather demographic data (e.g., years of teaching experience, degree information, teaching contexts), document information about teachers' previous experiences with CT and programming, and track their goals for PD and CT infusion over time. We utilized validated items from the Pedagogical Discontentment Survey (Southerland et al., 2011) and open-ended items related to teachers' goals for attending PD. Returning participants also responded to questions about their implementation of CT-infused lessons. Pre-PD and post-PD survey items remained the same for each year, with minor adjustments made for returning participants, to enable longitudinal comparison.

4.3.2 Content Knowledge Assessment (2018, 2019, 2020)

For the first three years of summer PD, teachers completed a content knowledge assessment. Items were selected from the Exploring Computer Science Computational Thinking assessment, which was developed and validated by SRI (Snow et al., 2017). The assessment included 13 questions (12 multiple-choice or multiple answer and one open-ended question) and focused on teachers' knowledge of CT concepts (pattern recognition, abstraction, decomposition, and algorithms) and programming concepts (loops, variables). The assessment was administered at the beginning of the 2018 and 2019 PD sessions and at the conclusion of the 2020 session. It was not administered in 2021 due to the fact that all teachers were returning participants and the focus of the PD for that year was on curriculum creation.

4.3.3 Lesson Plans, Programming and Pedagogical Artifacts, and Director's Cuts (2018, 2019, 2020, 2021)

Lesson plans, programming and pedagogical artifacts, and director's cut videos, in which teachers unpacked their lessons in 2 to 15-min videos, were used to examine growth of teachers' technological pedagogical content knowledge over time. Ana created materials for all four years, while Charlotte created materials for three years.

4.3.4 Practitioner Inquiry Project Artifacts (2022)

As previously mentioned, the culminating activity for the final year of the project was a teacher grant project (Jocius et al., 2024) in which teachers applied for funds to complete CT infusion projects of their choice. Numerous project artifacts, including grant applications, teacher-collected data, showcase presentations, infographic overviews, and implementation materials (PD activities and slides, resources used with students in after-school programs), were collected and analyzed.

4.3.5 Interviews (2019, 2020, Summer 2021, Fall 2021, 2022)

Semi-structured interviews were held at five points in the project–following the summer PD in 2019, 2020, and 2021, prior to practitioner inquiry project implementation, and following practitioner inquiry project implementation. Interviews were conducted by various members of the project team. Post-PD interview questions focused on teachers' definitions of CT, personal goals for CT infusion, experiences with CT-infused lesson implementation, barriers and accelerators for CT infusion, and PD experiences. Pre-practitioner inquiry project interviews focused on teachers' initial plans and questions about logistics, while post-practitioner inquiry projects centered on teacher reflections, research processes, and project impact.

4.4 Data Analysis

This study utilized a longitudinal case study design that focused on identifying patterns to build explanations of teacher learning and CT infusion practices over time (Merriam, 1998). First, all data were transcribed using multimodal transcription techniques (Bezemer & Mavers, 2011). Data analysis proceeded in four phases, with the first three phases aligned with the research questions and the fourth phase focusing on a holistic and cross-case comparison among teachers. Table 2 includes an overview of the phases of data analysis.

4.4.1 Pedagogical Content Knowledge

The primary sources of data used to respond to the first research question were open-ended survey response items and interviews collected at five points, after each summer PD and prior to and following the completion of practitioner inquiry projects. All interviews were transcribed and broken into idea units (Gee, 2011). Then, using line-by-line open qualitative coding techniques (Charmaz, 2015; Saldaña, 2021), two members of the research team independently coded the data, engaged in relating to teacher goals for the PD and imple-

Table 2 Overview of data analysis

Research question	Focus of analysis	Data sources
RQ1	Pedagogical Content Knowledge	• Pre-PD and Post-PD Surveys • Post-PD Interviews (CT definitions, PD goals) • Pre- and post-practitioner inquiry project interviews (project goals)
RQ2	Technological Content Knowledge	• Content knowledge assessment (Snow et al., 2017) • Programming products
RQ3	Technological Pedagogical Content Knowledge	• Lessons and artifacts (slides, programming products, student handouts, Director's Cuts) created during the PD • Practitioner inquiry project artifacts • Post-PD interviews (lesson implementation experiences, barriers and accelerators, PD experiences) • Pre- and post-practitioner inquiry project interviews (project experiences and outcomes)

mentation experiences over time. Recursive cycles of open and axial coding resulted in five themes: programming tools, CT as a thinking practice, discipline-specific CT infusion, barriers to CT infusion, and collaboration. We then created charts to map teachers' responses to enable longitudinal comparisons.

4.4.2 Technological Content Knowledge

To answer the second research question, we investigated teachers' content assessment responses and block-based (Snap*!* and Python) programming products created by the teachers during the PD. For the content assessments, we conducted item analyses for each teacher to detail the number and types of questions answered correctly and incorrectly. To analyze the programming products, we examined the CT concepts used in the Snap blocks, such as pattern recognition implemented through the use of loops, abstraction with the use of variables and user-defined blocks, and decomposition with the use of if-else blocks. Also, we paid attention to the complexity of the project by measuring the total number of blocks used and the categories of blocks.

4.4.3 Technological Pedagogical Content Knowledge

The primary sources of data for the analysis of technological pedagogical content knowledge were lesson plans and artifacts teachers created during the PD and refined over the course of classroom implementation. The focus of analysis was on teachers' conceptualizations of CT integration, their analysis of student and teacher learning as a result of their lessons or practitioner inquiry projects, and their reflections on how their pedagogical practices changed over time. After creating comparison charts for each teacher, we used thematic analysis and constant comparative methods (Charmaz, 2015), which resulted in six themes: lesson focus, use of CT concepts, disciplinary concepts, programming tools used, collaboration with other PD participants or school-based colleagues, and scaffolds for student CT learning. Then, we triangulated this analysis of artifacts with post-PD interview responses

related to lesson implementation experiences, barriers and accelerators, and PD experiences, as well as the pre- and post-inquiry project interviews.

4.4.4 Data Analysis Matrices and Cross-Case Comparisons

In a method similar to Mouza et al.'s (2009) work on longitudinal analysis of teachers' technology integration, we created matrices for each teacher that displayed features of their pedagogical content knowledge, technological content knowledge, and technological pedagogical content knowledge at five points in time: Summer 2018 (Ana only), Summer 2019, Summer 2020, Summer 2021, and May 2022. Then, we looked across the matrices to enable comparisons of the two case studies.

4.4.5 Trustworthiness

Throughout each phase of data analysis, members of the research team met to discuss emergent themes and the use of data sources. As the data were collaboratively analyzed by all members of the research team, inter-rater agreement was not calculated. We did keep audit trails in reflective journals, triangulated multiple sources of evidence to analyze data in relation to each research question, and conducted member checks with teachers and asked them to reflect on their own learning.

5 Findings

5.1 Ana

5.1.1 Pedagogical Content Knowledge

Ana began attending Infusing Computing in Summer 2018 and had no previous experience with coding or CT as either a learner or teacher. Analysis of her survey responses across the four years of the project mark a definite shift in her thinking about the generic use of coding tools to intentional infusion of CT concepts and practices. For example, in a 2018 pre-PD survey, her vision for infusing CT was vague: "We are planning a 'getting to know you' to start the year off." As she came to better understand CT, her focus shifted from using the technology to leveraging the practices for content instruction. For example, in her 2019 pre-PD survey, she said, "I want to be able to use this to teach my content, not just as a wrap up to a unit like I did last year." She also mentioned that one of the barriers that she had to overcome was learning how to "use it to integrate into my curriculum instead of using it for the sake of using it." So, there was growth from 2018 to 2019 in the specificity of her response, as well as her apparent comfort with CT terms such as "debugging."

Prior to her third summer of PD participation in 2020, she began focusing more intentionally on the CT concepts, with an aim of "helping students use the CT concepts in solving math problems." By 2021–2022, Ana's final year of the program, she had sought out numerous opportunities to infuse CT and had even developed an after-school Coding Club for students. Her 2021 pre-PD survey outlined the various ways she had used CT in her classroom and school during the previous year: "hour of code, coding club, used the sum-

mer lessons." She noted that viewing students' growth in Coding Club was "awesome" and that students "went so far with it." These responses demonstrate how Ana came to see the balance between helping students learn how to use the technology and coming to understand CT as a process for thinking.

5.1.2 Technological Content Knowledge

On the initial content assessment administered in 2018, Ana scored 10.5 points out of a possible 13 points (80.8%), which was higher than the average overall teacher score (7.49 points; 57.6%) on that year's assessment. While she scored on all items focused on programming components correctly (one of just three teachers to do so), she missed items related to defining and operationalizing CT concepts such as pattern recognition and algorithms. In 2019, Ana scored 11 of 13. While still higher than the teacher average for that year, she missed two of the same items related to algorithms and pattern recognition. In 2020, she scored all questions correctly.

Growth in Ana's technological content knowledge was also evident in analysis of the programming products that she created over time (see Fig. 2 for timeline and focus of Ana and Charlotte's work in each year of the project). In 2018, Ana created a Snap*!* narrative program to introduce herself to her class. It included animations that developed interactions with the user by soliciting feedback through questioning. Her program, which incorporated images of her family, utilized four variables and three different types of blocks, including multiple examples of "when" blocks, but included very little variety in terms of custom block types, loops, or other more complex programming concepts (see Fig. 3).

In 2019, 2020, and 2021, Ana worked with colleagues from her school and/or others that she met at the PD to create interdisciplinary lessons and programming products. In 2019, she collaborated with an ELA teacher from her school, Hazel, to create a Snap*!* program to introduce students to STEAM. As compared to her 2018 product, changes in her approach to the task, as well as the types of programming concepts represented, demonstrate her increasing knowledge of programming features and scaffolds. For example, rather than asking students to interact with the completed project, her 2019 project required students to debug and modify code. Using the Snap*!* annotation tool, which allows users to mark particular blocks with notes, she tasked students with editing, removing, and reordering blocks; identifying loops; and creating their own custom blocks (see "Fix Me" exercises in Fig. 4). In total, she used seven categories of blocks, including two loops, four custom blocks, and five if/then/else blocks.

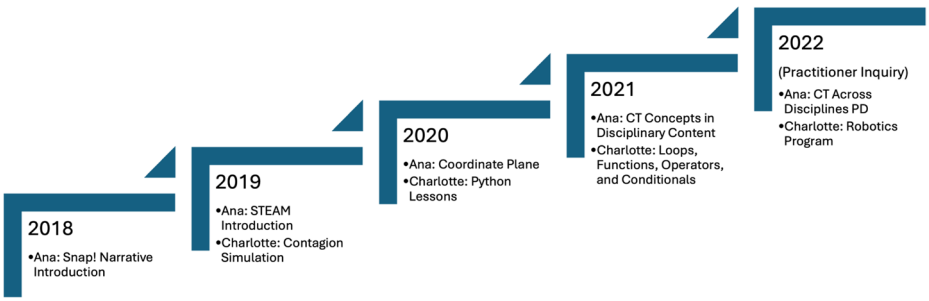

Fig. 2 Timeline of Ana and Charlotte's programming projects over time

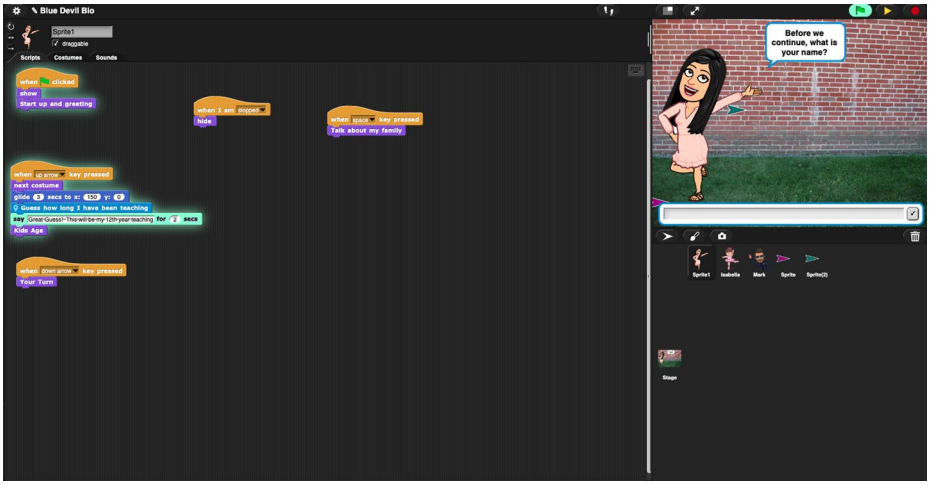

Fig. 3 Code Snippet from Ana's Snap! Project Created During the 2018 PD

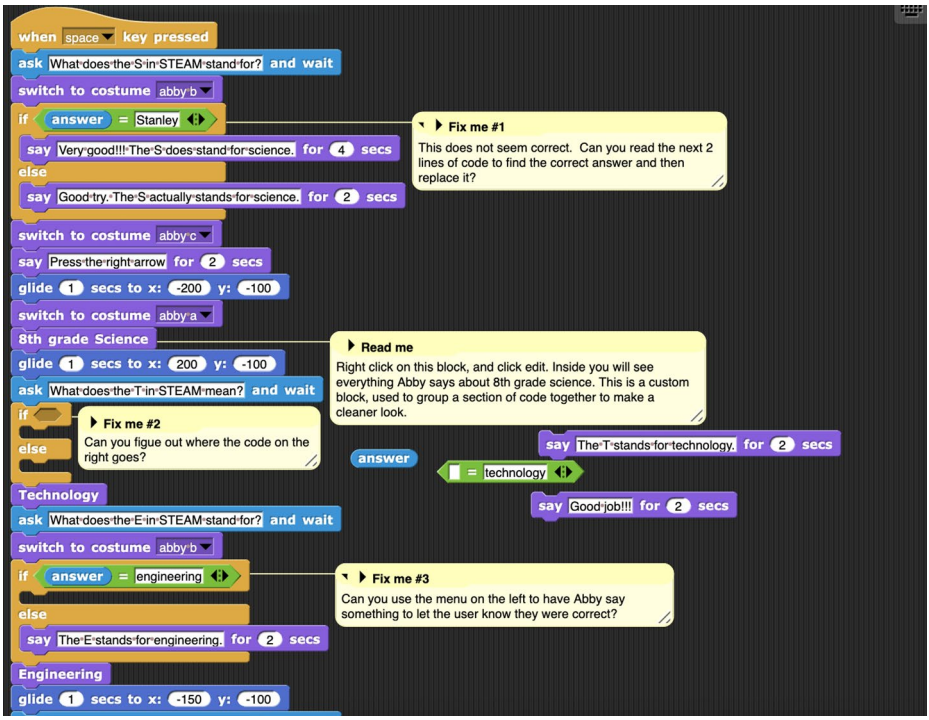

Fig. 4 Code Snippet from Ana's Snap! Project Created During the 2019 PD

In 2020, Ana worked with two colleagues to create an interdisciplinary Snap*!* program in which students decomposed code to review parts of the coordinate plane and break down CT concepts. The program includes seven sprites with code blocks associated for each sprite, different variables to capture the coordinates and other pieces of information, 13 custom blocks, and a Boolean operator. By including several sprites (see Fig. 5) and new custom blocks, she was able to create a scaffolded learning experience for students that integrated CT, programming, and disciplinary knowledge.

In 2021, Ana created a narrative Snap*!* program and matching game to explicitly introduce students to four CT concepts: pattern recognition, abstraction, decomposition, and algorithms. Her goal was to use these concepts to guide them through the process of modifying and creating code to develop their own introduction programs. Using nine sprites, several types of programming blocks, including custom blocks (see Fig. 6) that used parameters to develop flexibility in the use of algorithms, students are tasked with using multiple sprites, recording sounds, and developing their own custom blocks to create narrative stories.

5.1.3 Technological Pedagogical Content Knowledge

In 2018, after completing her first year of PD, Ana was able to successfully use her model introduction to share an example of programming, but she wasn't able to implement the component where students coded their own introductions due to time constraints. However, as she reflected in a post-PD interview in 2019, she decided to build additional coding projects into her "curriculum to have them create a Snap*!* presentation about polynomials. And they had to teach me something about polynomials, but it had to be interactive. So they had to have questions and then a certain response to go to each type of answer." She noted that she struggled due to her own lack of technological knowledge:

> The struggle was when I taught my segment on polynomials. It was around the February, March timeframe. So, it had been a while [since the summer 2018 PD]…and I kind of knew what I was doing, but I totally forgot some of it and I'm sitting up there

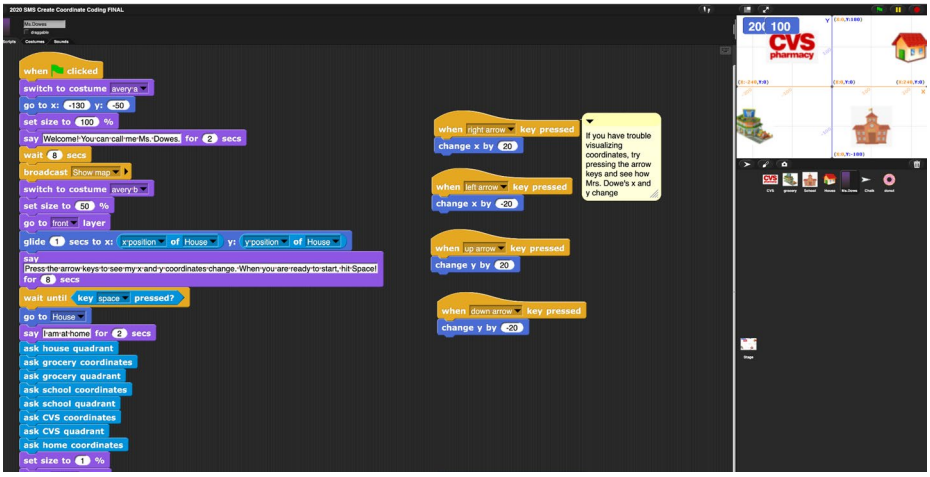

Fig. 5 Code Snippet from Ana's Snap! Project Created During the 2020 PD

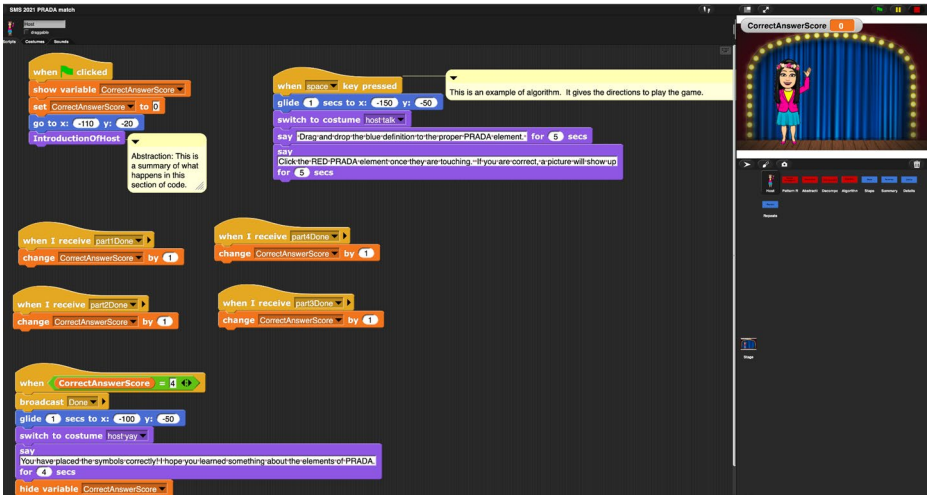

Fig. 6 Code Snippet from Ana's Snap! Project Created During the 2021 PD

trying to show them little things and I'm like, crap, I don't remember how to do this, so I look like a ding dong.

As she moved into her second summer of PD in 2019, her aim was to grow more confident with coding and to create a lesson that would introduce rather than review content. In the post-PD interview, she described the greatest growth as being in her understanding of the CT concepts. As she said following the 2019 summer PD, "I've got the experience behind me, I feel more confident with coding. I love the way they [project facilitators] have presented the lessons and the different ways to teach it. And that's given me more to take back."

In 2020, her third year with the project, Ana incorporated a series of CT activities to help students learn how to "decompose a script of code to review the parts of the coordinate plane and technical terms." In a series of lessons, students participated in scaffolded paired programming activities, including recognizing patterns in the coding of sprites and completing a Parson's problem, which is an activity common in computer science that tasks users with blocks that are out of order. Within the Snap*!* interface, she included notes for students to guide their thinking, demonstrating pedagogical uses of built-in Snap*!* features.

During the Summer 2021 PD, in addition to the Snap*!* narrative introduction to the four CT elements, Ana worked collaboratively with Hazel to develop a lesson that tasked students with identifying CT concepts in the world around them. The lesson plan also incorporated small-group discussions for students to make interdisciplinary connections and aimed to set a foundation for later content-specific, CT-infused lessons. This work was critical for the development of their collaborative practitioner inquiry project; several elements that they created for students, such as introduction PPT slides and small-group discussion prompts, were also used within the PD they created for teachers. In her final interview following the practitioner inquiry projects, Ana reflected on her desire to continue to build her knowledge and comfort with CT infusion: "I want to continue to build my personal bank of lessons with my students and be more purposeful, like with making sure to repeatedly talk about PRADA throughout the school year."

6 Summary

As Ana developed pedagogical content knowledge, her focus shifted from more generic forms of CT instruction to more content-specific ways of infusing CT into her math instruction. As she noted in her surveys and reflections, a big piece of this work was to think critically about ways to enhance existing instruction using CT, rather than using it as a standalone piece or an "add-on" activity. Growth in her technological content knowledge aided her abilities to infuse CT into her instruction and the development of her technological pedagogical content knowledge. As she described, rather than focusing on the "little" components of Snap*!*, she was able to instead devote more time and energy to developing meaningful learning experiences that co-develop both content and CT in a symbiotic process.

6.1 Charlotte

6.1.1 Pedagogical Content Knowledge

Charlotte, who began attending Infusing Computing PD in 2019, had previous experience with programming tools like Scratch and Code.org curriculum components. At the onset of her participation in the project, she was working as a high school science teacher and had also taught one semester of an AP Computer Science course. Like Ana, Charlotte's responses to surveys detailed how her thinking about CT changed over the course of her work with Infusing Computing. She attended as part of a school-based team, and her initial description of goals for CT infusion in the 2019 pre-PD survey focused generally on implementing lessons: the "team and I will be co-teaching several lessons in the fall." In 2020, she accepted a new role as an instructional coach, and her primary responsibilities included supporting teachers with technology. Her goals for that second year were primarily focused on the use of coding tools: "Although we use code.org curriculum, I want to add SNAP and Python this year to give the kids a little taste of the different languages and relate it back to the pseudocode they will see on their exam."

Like Ana, in her 2021 pre-PD survey, she described a shift in focus from tools to CT as a thinking process. She also described supporting others in bringing CT to their classrooms through "multiple lessons, CT vocab, infused projects with students, present[ing] PRADA to admins" and said that she had come to see CT as "a powerful tool to facilitate higher-order thinking." In addition to growth in her knowledge about CT as a tool for engendering disciplinary thinking, she referenced the importance of sharing the benefits of CT infusion with multiple stakeholders, including school and district leaders. This desire to help others build expertise with CT content, which was likely influenced by shifts in her role, led to her choice of practitioner inquiry projects (after-school coding club to build a pipeline of middle and high school students interested in CT and CS).

6.1.2 Technological Content Knowledge

Charlotte entered the PD with prior experience in CS education and one semester of teaching an AP CS course, but she scored a 5.5 out of 13 on her content assessment for 2019, her first year with the project. This was two points below the average teacher score for that year, and she missed items related to both CT concepts (algorithm, pattern recognition, decom-

position) as well as programming items (variables, functions, repeat blocks). In 2020, she scored 8.5 of 13, which was again slightly below the teacher average for that year. While she did better with CT concepts, only missing one item related to algorithms, she still missed three items related to programming concepts and applications.

Over her participation in the project, there were shifts in the type of programming products Charlotte created, both in terms of the content focus and sophistication of programming concepts (see Fig. 2). In 2019, while serving as a science teacher, Charlotte collaborated with colleagues from her school to develop an interdisciplinary lesson sequence taught in both the biology and CS classrooms. She built upon a Snap*!* contagion simulation developed by the Infusing Computing team. In her adapted lesson, students were asked to examine patterns in the code and solve a Parson's Problem, in which they had to put the blocks in the correct order (see Fig. 7). She also developed a series of supplemental materials to address science standards related to population growth and the impact on the biosphere. The resulting program included two different types of custom "when" blocks, an "if" block, and a loop used to create clones that spread a contagion across the United States.

In 2020, Charlotte attended the Python sessions, which were added to the PD that year to give experienced participants an opportunity to explore a different programming language. As she mentioned in her final interview, she had very limited previous experience with Python, so the experience and resources were crucial to developing her lesson. In her lesson reflections, she referenced her shift to an instructional coach role as being one of the primary determining factors in choosing to complete a lesson focused on programming concepts. Her final product was a scaffolded Google Colab lesson in which students worked through existing code to practice programming concepts (e.g., adding logic using if/then statements and exploring loops). She used several components of Colab notebook exercises used within the PD, but incorporated annotations and support for students throughout her work. In 2021, Charlotte refined the 2020 introduction to Python lesson, and then developed multiple lessons that leveraged Python to help students explore more complex programming concepts like lists, mathematical operators, and decision structures (see Fig. 8).

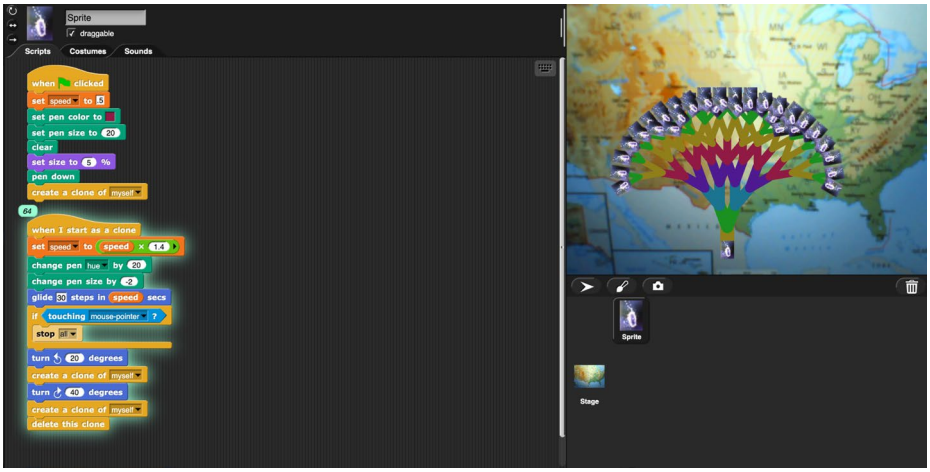

Fig. 7 Charlotte's Snap! Contagion project

Write Code:

In the code block below write code to do the following:

- Create a list to store 10 scores
- Write a loop to calculate the total of all the scores
- Find the average score by diving the total by the length of the list of scores.
- Optional: Convert the average score to a percentage and use an if-elif-else block to assign a grade.

```
## Write your code below
scores = [99, 98, 95, 93, 87, 83, 85, 77, 79, 100]
total = 0

for score in scores:
  total = total + score
  print('t = ', total)

average = round(total/len(scores))

print('total = ', total)
print('average = ', average)

if average >= 90:
  grade = 'A'
elif average <= 89 and average >= 80:
  grade = 'B'
else:
  grade = 'C'

print(grade)
```

Fig. 8 Charlotte's Python coding introduction lesson

6.1.3 Technological Pedagogical Content Knowledge

In Charlotte's 2019 project, her students were tasked with using Snap*!* to develop models for the spread of a zombie contagion using exponential functions. In the math classroom, after creating tree diagrams in response to a contagion scenario, students interacted with the Snap*!* simulation that Charlotte and her colleagues modified (see Fig. 7). The biology component involved the investigation of different variables that affected the growth of the contagion. Then, as part of a computer science lesson, students examined patterns in the code and solved a Parson's Problem. However, as her team planned for implementation of the contagion-focused lesson in early March of 2020, COVID-19 began to spread across the globe. Although she wasn't able to implement the lesson in full due to the switch to primarily asynchronous remote teaching in Spring 2020, several of her colleagues were able to implement it in future years, as the content was participant relevant during the pandemic.

In 2020, her second year with the project, Charlotte was preparing to take on her new role as a district instructional coach. For the summer 2020 PD, in addition to her participant role, she helped to facilitate small-group discussions for the science teachers. This experience helped her to better understand CT infusion and to get ready to take on a similar role in her district. As she said in a post-PD interview,

> I think a light bulb went off for me when we went through the PRADA. And they say that the highest level of learning is teaching something and while I was learning something to teach it to someone else or to pull information out of someone else, I

> think that helped me take it to a level where, hey, this can fit anywhere. It doesn't matter who you are.

In Summer 2020, Charlotte also participated in Python bootcamp sessions and collaborated with four other teachers that she had met the previous year at Infusing Computing to create a lesson introducing students to Python. In addition to introducing each of the CT elements and giving students opportunities to reflect and identify real-world examples, the lesson highlighted conceptual understandings of programming and distinguished Python characteristics from those found in other programming languages. Students came to define pseudocode, which is the scripting language used on the AP exam to make sure that students are understanding programming concepts. In an interview, she talked about incorporating reflective activities "to pull out any misconceptions that they may have or to have them explain in detail certain parts of the problem to make sure they completely understand it," which connected to the shift in her definition of computational thinking and the importance of unpacking the differences between programming and CT.

Her 2021 PD project built on her 2020 lesson to create a "working understanding of loops, functions, operators, IF/ELSE conditionals." She collaborated with one colleague who had also worked on the 2020 lesson. In her Director's Cut, Charlotte said,

> We wanted to incorporate more problem-solving skills, and we wanted these problem-solving skills not to be just something that we touched on and then ran away from for the rest of the year. We wanted it to be something where it's shown to them, it's modeled for them, and then they get to reflect on their thinking.

The lesson goals aligned with Charlotte's shift in focus from CT tools to problem-solving processes; as she mentioned in the director's cut, "We wanted to make it a habit for these children, and a habit in that they think about their thinking and their problem-solving practices every day."

Charlotte's practitioner inquiry project aimed to expand access to CT and CS for students in both middle and high schools within her school and district. As she described in her application for the project, "My mission is for every student to have at least some coding experience prior to leaving high school. I want them to see, control, and understand the power of thinking like a computer can have in almost all aspects of life." She built an after-school robotics program that integrated engineering practices and provided a scaffolded introduction to both CT and CS. As she mentioned in her final presentation, the club drew on disciplinary practices developed in STEM classrooms. In the club sessions, after Charlotte taught mini-lessons that introduced the PRADA concepts, students experimented with different programming tools, such as Misty and Sphero robots. The club, which served 80 students, also helped students to "make computer science creative" by having students "use robots to draw a picture or do a short play or dance."

7 Summary

Charlotte began the Infusing Computing PD with more experience than Ana with both computational thinking and programming. However, she followed a similar trajectory in that her thinking shifted away from focusing on programming tools and languages toward focusing on the thinking processes involved with CT. As she moved from a role as a disciplinary teacher to instructional coach, there were corresponding shifts in the focus of her instructional design from discipline-specific lessons to materials that would support teachers and broaden participation through the robotics program. Unlike Ana, Charlotte participated in sessions that expanded her existing knowledge of multiple programming languages (Snap*!* and Python). She was able to leverage this knowledge to focus on CT and programming concepts and skills that are not specific to one particular knowledge, but instead build a foundation for students to utilize across multiple contexts and with multiple tools.

8 Discussion

Using TPACK as a theoretical lens, this study examines two longitudinal case studies of teachers' CT infusion practices over a five-year project. We analyzed teacher growth in three primary areas–content knowledge, technological knowledge, and pedagogical content knowledge, in line with previous research on teacher learning about new pedagogies involving technology (Mouza et al., 2017). We also compared teacher learning across cases to describe how teachers coming from different content area backgrounds (math and science) and with different levels of programming experience developed expertise in CT infusion over time.

In relation to pedagogical content knowledge, we found that both teachers demonstrated shifts in their thinking about CT as related to coding tools to CT as a thinking practice with value for supporting students' development of disciplinary thinking. This is in alignment with previous research demonstrating that teachers must have sufficient knowledge about CT to consider how to integrate it with existing practices (Hestness et al., 2018). This study shows that teachers' understandings may gradually develop over time. Ana, for example, started the PD with general goals for using coding tools with students; in the initial years of the project, she planned activities to introduce students to coding tools, whereas in the later years, she focused on full integration of CT concepts and practices with disciplinary learning goals. Over time, her focus shifted toward the introduction of CT concepts to promote content area learning. This perspective gave her the confidence to share her knowledge with colleagues in the Infusing Computing project and within her school. Charlotte, who came into the PD with previous coding knowledge and skills, as well as experiences in teaching AP computer science, also began the PD with a focus on coding tools, which shifted to a focus on CT as a thinking practice and a commitment to building a pipeline of students interested in CS.

Both teachers also demonstrated growth in terms of their technological content knowledge and abilities to use coding tools, which prior research has recognized as a critical component of teacher learning about CT infusion (Caskarlu et al., 2021). Ana, who began the PD without prior programming experience, improved over time in her ability to recognize and connect to CT concepts and their use of increasingly complex programming components

in their Snap*!* programs (e.g., if/then/else blocks, custom blocks, variables, and operators). Charlotte, who learned to code in Snap*!* during her first year and then took the Python sessions in her final two years, developed the ability to use more complex concepts like loops, operators, and variables. Importantly, as each teacher's technological content knowledge increased, they also developed increasingly complex activities for students to engage in. In their first years in the PD, both focused on the development of interactive simulations or narratives for students to interact with. Over time, they incorporated student coding activities, such as solving Parson's Problems, debugging code, and coding.

Finally, we examined shifts in teachers' technological pedagogical content knowledge and found differences in how they conceptualized and implemented CT-infused practices. Ana began the PD by focusing on generalized CT and programming skills. For example, in the first year of teaching lessons, she asked students to create content-agnostic Snap*!* narratives that introduced themselves to the class. As they developed greater familiarity with CT and coding tools, she developed both plugged-in and unplugged lessons that fully integrated CT with disciplinary learning goals. Her practitioner inquiry project, a PD retreat that introduced CT to their colleagues, utilized unplugged lessons and activities to build a foundation for CT infusion at their school. Charlotte, who began the PD with more experience with both CT and CS, initially focused on collaborating with other teachers to build an interdisciplinary lesson. As her role at her school changed from teacher to instructional coach, the lessons and activities she designed during the PD also changed to focus on CT as a thinking process with broad applicability for multiple disciplines. In the final year of the project, like Ana, she aimed to broaden participation in CT infusion, but her focus was not on teachers, but on building an after-school robotics program for students across middle and high school.

Our analysis of Ana and Charlotte's cases also helped us to consider how professional contexts can shape the form and function of disciplinary CT infusion. For her first three years of participation, Ana represents a fairly straightforward example of disciplinary infusion in the mathematics classroom–she was able to embed CT concepts like pattern recognition, abstraction, and algorithms to support students' understanding of mathematical practices and processes. Charlotte's case is more complex. In her first year of PD participation, she focused specifically on science content standards. However, as she moved into her coaching role, her instructional focus shifted to more general CS and CT activities that could be implemented in a variety of classrooms and informal learning contexts. While the lessons Charlotte created in her second and third years lack a discipline-specific lens, her work does help us understand how CT can be infused in adjacent disciplines, including engineering and CS, particularly when teachers assume leadership or coaching roles that transcend a single content area. Interestingly, in both Ana and Charlotte's final year of participation, as they developed practitioner inquiry projects, both chose to focus on broadening participation in CT infusion, which may have led to more content-agnostic approaches that focused on CT and coding knowledge rather than direct links to disciplinary standards. This suggests that disciplinary CT infusion isn't a static process, but one that interacts with evolving professional responsibilities and institutional contexts.

Examining teacher learning across time also forced us to confront some of the assumptions we had made at the outset of the study regarding how teachers learn to "do" CT infusion. For example, we initially thought that having teachers design their own CT infusion lessons would provide a more authentic context for teacher learning; this is a common practice in in-service teacher PD (Coenders & Verhoef, 2019) and has been proposed by

other researchers as one potential strategy to mitigate elementary teacher concerns about connections between disciplinary content and CT (Ketelhut, 2020). We purposefully chose to provide a small set of sample lessons in the first year so that teachers would have models for designing their own lessons, but quickly found that they often preferred to adapt lessons taught during the PD. This was reflected in Ana's use of PD activities and materials in her classroom and eventually in her practitioner inquiry projects, as well as Charlotte's adaptation of CT introduction materials. As we've talked as a project team about how we would structure similar projects in the future, in the early years (Y1 and Y2), we would instead ask teachers to take a pre-existing, field-tested lesson, adapt it for their teaching contexts, and then reflect on the experience with colleagues who implemented the same lesson. As we've begun to test this model in related projects, we've found that providing common experiences with particular lessons can be a positive experience (Blanton et al., 2023).

We also found that the process of building deep understandings of CT infusion takes much longer than we originally believed. While some teachers were ready for classroom implementation in Y1, others took two or three years to build core content understandings of CT, which impacted their abilities to successfully implement CT-infused lessons. Even teachers like Ana, who felt comfortable with implementing lessons in the first year, noted that continued participation in PD and work with students caused them to question nascent understandings of CT infusion. In some cases, this caused teachers to forego classroom coding activities in favor of unplugged lessons that featured CT concepts. Ana, for example, started in Y1 with comprehensive coding projects, and then decided to do unplugged lessons in Y3 and Y4 due to the need to slowly build a strong foundation of CT.

CT infusion is not "one size fits all." Each teacher brings different lived experiences, pedagogical practices, and disciplinary norms with them to any professional learning experience and PD must provide opportunities to draw on teachers' existing knowledge as well as introduce new ideas. CT infusion, which bridges familiar disciplinary practices with often unfamiliar computer science concepts, practices, and dispositions, requires teachers to navigate the alignment of CT-infused curriculum with disciplinary standards, available technology, their own developing understandings of CT, and pedagogical approaches.

9 Conclusion

Examining teachers' journeys to CT infusion over time has allowed us to examine the interrelationships of content, technological, and pedagogical content knowledge. Seeing how teachers take up these new practices, and how their understanding of CT as a tool for supporting disciplinary learning, highlights the potential of CT infusion in a variety of content area classrooms. We also believe it highlights the need to prioritize future research that examines teacher learning not just immediately following a PD experience, but that follows teachers as they grapple with new learning and implement new pedagogical practices in their classrooms, schools, and districts.

In order for CT to be fully embedded into disciplinary classrooms, to create spaces where terms like algorithm and decomposition and debugging are used frequently and accurately, and where students have chances to experience both plugged-in and unplugged CT learning experiences that help them build a strong foundation in both CT and content knowledge, teachers need meaningful and sustained learning experiences. While stand-alone workshops

of a brief duration can introduce teachers to CT and expose them to new tools, it's in the messy middle of PD, teaching, and reflection that teachers can do the work needed to build self-efficacy and pedagogical content knowledge for CT infusion.

Funding This material is based upon work supported by the National Science Foundation under grant numbers 1742351 and 1742332.

Data availability Some data cannot be shared publicly due to participant privacy and Institutional Review Board restrictions. Data excerpts are available upon request to the corresponding author.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Bezemer, J., & Mavers, D. (2011). Multimodal transcription as academic practice: A social semiotic perspective. *International Journal of Social Research Methodology, 14*(3), 191–206.

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

Blanton, M., Jocius, R., Albert, J., Joshi, D., & Andrews, A. (2023). Dragons, squishy circuits, and computational thinking: Integrating scientific literacies into elementary classrooms. Language Arts 100(4), 269–281. https://doi.org/10.58680/la202332307

Brantley-Dias, L., & Ertmer, P. A. (2013). Goldilocks and TPACK: Is the construct 'just right?' *Journal of Research on Technology in Education, 46*(2), 103–128.

Burke, Q., O'Byrne, W. I., & Kafai, Y. B. (2016). Computational participation: Understanding coding as an extension of literacy instruction. *Journal of Adolescent & Adult Literacy, 59*(4), 371–375.

Caskurlu, S., Yadav, A., Dunbar, K., & Santo, R. (2021). Professional development as a bridge between teacher competencies and computational thinking integration. In *Computational Thinking in Education* (pp. 136–150). Routledge.

Chai, C. S. (2019). Teacher professional development for science, technology, engineering and mathematics (STEM) education: A review from the perspectives of technological pedagogical content (TPACK). *The Asia-Pacific Education Researcher, 28*(1), 5–13.

Chai, C. S., Koh, J. H. L., Tsai, C. C., & Tan, L. L. W. (2011). Modeling primary school pre-service teachers' Technological Pedagogical Content Knowledge (TPACK) for meaningful learning with information and communication technology (ICT). *Computers & Education, 57*(1), 1184–1193.

Chaipidech, P., Kajonmanee, T., Chaipah, K., Panjaburee, P., & Srisawasdi, N. (2021). Implementation of an andragogical teacher professional development training program for boosting TPACK in STEM education. *Educational Technology & Society, 24*(4), 220–239.

Chalmers, C. (2018). Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction, 17*, 93–100.

Charmaz, K. (2015). *Constructing grounded theory* (2nd ed.). Sage.

CSTA, & ECEP Alliance (2022). *2022 State of computer science education: understanding our national imperative.* Retrieved from https://advocacy.code.org/stateofcs

Computer Science Teachers Association (CSTA) & International Society for Technology in Education (ISTE). (2011). Operational definition of computational thinking. Available: http://www.csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf

Coenders, F., & Verhoef, N. (2019). Lesson Study: Professional development (PD) for beginning and experienced teachers. *Professional Development in Education, 45*(2), 217–230.

Dalal, M., Archambault, L., & Shelton, C. (2017). Professional development for international teachers: Examining TPACK and technology integration decision making. *Journal of Research on Technology in Education, 49*(3–4), 117–133.

Dong, Y., Cateté, V., Jocius, R., Lytle, N., Barnes, T. Albert, J., Joshi, D., Robinson, R., & Andrews, A. (2019). *PRADA: A practical model for integrating computational thinking in K-12 education*. In Proceedings of the 50th ACM technical symposium on computer science education (SIGCSE '19) (pp. 906-912). Association for Computing Machinery (ACM).

Dong, Y., Chai, C. S., Sang, G. Y., Koh, J. H. L., & Tsai, C. C. (2015). Exploring the profiles and interplays of pre-service and in-service teachers' technological pedagogical content knowledge (TPACK) in China. *Journal of Educational Technology & Society, 18*(1), 158–169.

Ericson, B., Adrion, W. R., Fall, R., & Guzdial, M. (2016). State-based progress towards computer science for all. *ACM Inroads, 7*(4), 57–60.

Ferdig, R. E., Baumgartner, E., Hartshorne, R., Kaplan-Rakowski, R., & Mouza, C. (Eds.). (2020). *Teaching, technology, and teacher education during the COVID-19 pandemic: Stories from the field*. Association for the Advancement of Computing in Education.

Gee, J. (2011). *How to do discourse analysis: A toolkit*. Routledge.

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. *Computer Science Education: Perspectives on Teaching and Learning in School, 19*(1), 19–38.

Harris, J. B., & Hofer, M. J. (2011). Technological pedagogical content knowledge (TPACK) in action: A descriptive study of secondary teachers' curriculum-based, technology-related instructional planning. *Journal of Research on Technology in Education, 43*(3), 211–229.

Harris, J., Mishra, P., & Koehler, M. (2009). Teachers' technological pedagogical content knowledge and learning activity types: Curriculum-based technology integration reframed. *Journal of Research on Technology in Education, 41*(4), 393–416.

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

Hofer, M., & Grandgenett, N. (2012). TPACK development in teacher education: A longitudinal study of preservice teachers in a secondary MA Ed. program. *Journal of Research on Technology in Education, 45*(1), 83–106.

Hutchison, A., Nadolny, L., & Estapa, A. (2016). Using coding apps to support literacy instruction and develop coding literacy. *The Reading Teacher, 69*(5), 493–503. https://doi.org/10.1002/trtr.1440

Jacob, S. R., & Warschauer, M. (2018). Computational thinking and literacy. *Journal of Computer science Integration,* 1(1). https://doi.org/10.26716/jcsi.2018.01.1.1

Jaipal-Jamani, K., & Figg, C. (2015). A case study of a TPACK-based approach to teacher professional development: Teaching science with blogs. *Contemporary Issues in Technology and Teacher Education, 15*(2), 161–200.

Jocius, R., O'Byrne, I., Blanton, M., Albert, J., Joshi, D. & Robinson, R. (2021). Leveraging virtual professional development to build computational thinking literacies in English language arts classrooms. *Contemporary Issues in Technology and Teacher Education, 21*(4), 626-654. https://citejournal.org/volume-21/issue-4-21/english-language-arts/leveraging-virtual-professional-development-to-build-computational-thinking-literacies-in-english-language-arts-classrooms

Jocius, R., O'Byrne, W.I., Albert, J. Joshi, D., Blanton, M., Robinson, R., Andrews, A., Barnes, T., Cateté, V. (2022). Building a virtual community of practice: Teacher learning for computational thinking infusion. *TechTrends, 66*, 547–559. https://doi.org/10.1007/s11528-022-00729-6

Jocius, R., Albert, J., O'Byrne, I., Joshi, D., Robinson, R., & Blanton M. (2024). Computational thinking infusion as transformative teaching: Investigating content area teacher perspectives and practices. *Computer Science Education 34*(2), 222–251. https://doi.org/10.1080/08993408.2023.2210458

Kafai, Y., Proctor, C., & Lui, D. (2020). From theory bias to theory dialogue: Embracing cognitive, situated, and critical framings of computational thinking in K-12 CS education. *ACM Inroads, 11*(1), 44–53.

Kafai, Y. B., & Proctor, C. (2022). A revaluation of computational thinking in K–12 education: Moving toward computational literacies. *Educational Researcher, 51*(2), 146-151.

Ketelhut, D. J., Mills, K., Hestness, E., Cabrera, L., Plane, J., & McGinnis, J. R. (2020). Teacher change following a professional development experience in integrating computational thinking into elementary science. *Journal of Science Education and Technology, 29*, 174–188.

Koehler, M.J., & Mishra, P. (2008). Introducing TPCK. In AACTE Committee on Innovation and Technology (Eds.), *Handbook of technological pedagogical content knowledge (TPCK) for educators* (pp. 3-30). Routledge.

Koh, J. H. L., Chai, C. S., & Lim, W. Y. (2017). Teacher professional development for TPACK-21CL: Effects on teacher ICT integration and student outcomes. *Journal of Educational Computing Research, 55*(2), 172–196.

Koh, J. H. L., Chai, C. S., & Tsai, C. C. (2014). Demographic factors, TPACK constructs, and teachers' perceptions of constructivist-oriented TPACK. *Journal of Educational Technology & Society, 17*(1), 185–196.

Kong, S. C., & Lai, M. (2021). A proposed computational thinking teacher development framework for K-12 guided by the TPACK model. *Journal of Computers in Education, 9*, 379–402.
Kong, S. C., Lai, M., & Li, Y. (2023). Scaling up a teacher development programme for sustainable computational thinking education: TPACK surveys, concept tests and primary school visits. *Computers & Education, 194*, Article 104707.
Lachner, A., Fabian, A., Franke, U., Preiß, J., Jacob, L., Führer, C., & Thomas, P. (2021). Fostering preservice teachers' technological pedagogical content knowledge (TPACK): A quasi-experimental field study. *Computers & Education, 174*, Article 104304.
Lai, C., Wang, Q., & Huang, X. (2022). The differential interplay of TPACK, teacher beliefs, school culture and professional development with the nature of in-service EFL teachers' technology adoption. *British Journal of Educational Technology, 53*(5), 1389–1411.
Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation*. Cambridge University Press.
Li, Y., Schoenfeld, A. H., diSessa, A. A., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational thinking is more about thinking than computing. Journal for STEM Education Research, 3(1), 1-18.
Merriam, S. B. (1998). *Qualitative research and case study applications in education*. Jossey-Bass.
Mishra, P., & Koehler, M. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *Teachers College Record, 108*(6), 1017–1054.
Mishra, P., & Koehler, M. J. (2007). Technological pedagogical content knowledge (TPCK): Confronting the wicked problems of teaching with technology. In *Society for Information Technology & Teacher Education International Conference Proceedings* (pp. 2214-2226). Association for the Advancement of Computing in Education (AACE).
Mouza, C. (2009). Does research-based professional development make a difference? A longitudinal investigation of teacher learning in technology integration. *Teachers College Record, 111*(5), 1195–1241.
Mouza, C., Yang, H., Pan, Y. C., Ozden, S. Y., & Pollock, L. (2017). Resetting educational technology coursework for pre-service teachers: A computational thinking approach to the development of technological pedagogical content knowledge (TPACK). *Australasian Journal of Educational Technology*. https://doi.org/10.14742/ajet.3521
Niess, M. L. (2011). Investigating TPACK: Knowledge growth in teaching with technology. *Journal of Educational Computing Research, 44*(3), 299–317. https://doi.org/10.2190/EC.44.3.c
Olofson, M. W., Swallow, M. J., & Neumann, M. D. (2016). TPACKing: A constructivist framing of TPACK to analyze teachers' construction of knowledge. *Computers & Education, 95*, 188–201.
Ottenbreit-Leftwich, A., Liao, J. Y. C., Sadik, O., & Ertmer, P. (2018). Evolution of teachers' technology integration knowledge, beliefs, and practices: How can we support beginning teachers' use of technology? *Journal of Research on Technology in Education, 50*(4), 282–304.
Pareto, L., & Willermark, S. (2019). TPACK in situ: A design-based approach supporting professional development in practice. *Journal of Educational Computing Research, 57*(5), 1186-1226.
Rich, P. J., Mason, S. L., & O'Leary, J. (2021). Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. *Computers & Education, 168*, Article 104196.
Rodríguez Moreno, J., Agreda Montoro, M., & Ortiz Colón, A. (2019). Changes in teacher training within the TPACK model framework: A systematic review. *Sustainability, 11*(7), 1870. https://doi.org/10.3390/su11071870
Saldaña, J. (2021). *The coding manual for qualitative researchers* (4th ed.). Sage.
Saubern, R., Henderson, M., Heinrich, E., & Redmond, P. (2020). TPACK–time to reboot? *Australasian Journal of Educational Technology, 36*(3), 1–9.
Sauers, N. J., & McLeod, S. (2018). Teachers' technology competency and technology integration in 1: 1 schools. *Journal of Educational Computing Research, 56*(6), 892–910.
Shin, T., Koehler, M., Mishra, P., Schmidt, D., Baran, E., & Thompson, A. (2009, March). Changing technological pedagogical content knowledge (TPACK) through course experiences. In *Society for information technology & teacher education international conference* (pp. 4152–4159). Association for the Advancement of Computing in Education (AACE).
Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142-158.
Snow, E., Tate, C., Rutstein, D., & Bienkowski, M. (2017). Assessment design patterns for computational thinking practices in exploring computer science. Technical report. SRI International.
Southerland, S. A., Sowell, S., Blanchard, M., & Granger, E. M. (2011). Exploring the construct of pedagogical discontentment: A tool to understand science teachers' openness to reform. *Research in Science Education, 41*, 299–317.
Wang, J. (2017). Is the US education system ready for CS for all? *Communications of the ACM, 60*(8), 26–28.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*, 127–147.
Whalen, J., & Mouza, C. (2023). ChatGPT: Challenges, opportunities, and implications for teacher education. *Contemporary Issues in Technology and Teacher Education, 23*(1), 1–23.
Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://doi.org/10.1145/1118178.1118215
Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends, 60*, 565–568.
Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM, 60*(4), 55–62.
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
Yeh, Y. F., Chan, K. K. H., & Hsu, Y. S. (2021). Toward a framework that connects individual TPACK and collective TPACK: A systematic review of TPACK studies investigating teacher collaborative discourse in the learning by design process. *Computers & Education, 171*, Article 104238.
Yin, R. K. (2009). *Case study research: Design and methods*. Sage.

Authors and Affiliations

Robin Jocius[1] · Deepti Joshi[2] · Melanie Blanton[3] · Jennifer Albert[3] · W. Ian O'Byrne[4]

✉ Robin Jocius
robin.jocius@gmail.com

Deepti Joshi
djoshi@citadel.edu

Melanie Blanton
Mlblanto1@citadel.edu

Jennifer Albert
jennifer.albert@citadel.edu

W. Ian O'Byrne
obyrnei@cofc.edu

[1] Department of Curriculum and Instruction, University of Kentucky, 103 Dickey Hall, Lexington, KY 40506, USA

[2] Department of Cyber and Computer Sciences, The Citadel, 171 Moultrie St., Charleston, SC 29412, USA

[3] Zucker Family School of Education, The Citadel, 171 Moultrie St., Charleston, SC 29412, USA

[4] Department of Teacher Education, College of Charleston, 86 Wentworth Street, #234, Charleston, SC 29424, USA