Developing K-8 Computer Science Teachers' CS Knowledge, Self-efficacy, and Attitudes through Evidence-based Professional Development

Gwen Nugent
University of Nebraska-Lincoln
Lincoln, NE, USA
gnugent@unl.edu

Keting Chen
University of Nebraska-Lincoln
Lincoln, NE, USA
ke-ting.chen8985@huskers.unl.edu

Leen-Kiat Soh
University of Nebraska-Lincoln
Lincoln, NE, USA
lksoh@cse.unl.edu

Dongho Choi
University of Nebraska-Lincoln
Lincoln, NE, USA
tom.choi@huskers.unl.edu

Guy Trainin
University of Nebraska-Lincoln
Lincoln, NE, USA
gtrainin2@unl.edu

Wendy Smith
University of Nebraska-Lincoln
Lincoln, NE, USA
wsmith5@unl.edu

ABSTRACT

Broadening participation in computer science (CS) for elementary students is a growing movement, spurred by computing workforce demands and the need for younger students to develop skills in problem solving and critical/computational thinking. However, offering computer science instruction at this level is directly related to the availability of teachers prepared to teach the subject. Unfortunately, there are relatively few primary/elementary school teachers who have received formal training in computer science, and they often self-report a lack of CS subject matter expertise. Teacher development is a key factor to address these issues, and this paper describes professional development strategies and empirical impacts of a summer institute that included two graduate courses and a series of Saturday workshops during the subsequent academic year. Key elements included teaching a high-level programming language (Python and JavaScript), integrating CS content and pedagogy instruction, and involving both experienced K-12 CS teachers and University faculty as instructors. Empirical results showed that this carefully structured PD that incorporated evidence-based elements of sufficient duration, teacher active learning and collaboration, modeling, practice, and feedback can successfully impact teacher outcomes. Results showed significant gains in teacher CS knowledge (both pedagogy and content), self-efficacy, and perception of CS value. Moderating results – examining possible differential effects depending on teacher gender, years of teaching CS, and geographic locale – showed that the PD was successful with experienced and less experienced teachers, with teachers from both rural and urban locales, and with both males and females.

CCS CONCEPTS

• **Social and professional topics → K-12 education**.

ITiCSE 2022, July 8–13, 2022, Dublin, Ireland.

ACM ISBN 978-1-4503-9201-3/22/07…$15.00
https://doi.org/10.1145/3502718.3524771

KEYWORDS

Teacher professional development, K-8 education, research

ACM Reference Format:
Gwen Nugent, Keting Chen, Leen-Kiat Soh, Dongho Choi, Guy Trainin, and Wendy Smith. 2022. Developing K-8 Computer Science Teachers' CS Knowledge, Self-efficacy, and Attitudes through Evidence-based Professional Development. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education Vol 1 (ITiCSE 2022), July 8–13, 2022, Dublin, Ireland.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3502718.3524771

1 INTRODUCTION

The push for CS in younger grades is accelerating. In 2020, the number of states with K–12 computer science standards increased from 34 to 37, with an additional states developing standards informed by the CSTA K-12 standards[5]. This growth is supported by a recent survey indicating that 90% of parents want their children to study computer science[22], as well as a Microsoft study showing that 88% of teachers believe computer science is critical to future workplace success. Teachers reported that beyond simply coding, computer science can teach students problem solving and reasoning. Additionally, 83% of teachers surveyed believe coding builds students' creativity[23].

Elementary and middle schools' capacity for offering computer science courses is directly related to the availability of teachers prepared to teach the subject[17]. Despite the fact that 40 states have adopted computer science teacher certifications[5], there are relatively few elementary school teachers who have received formal training in computer science[36, 41]; and elementary teachers self-report a lack of CS subject matter expertise[19]. Professional development is key; a comprehensive survey of pre-college computer science education conducted by Google concluded that "teacher development is a key factor in the success of CSEd"[4]. Today there are growing numbers of professional development opportunities for teachers (see csteachers.org/page/quality-pd); but a comprehensive review of published CS PD studies[26] showed that programs serving high schools were most prevalent. There remains a critical need for more K-8 educators trained in teaching computer science.

Adding to this teacher scarcity is the lack of empirical research as to what constitutes effective CS K-8 instruction and what pedagogies and instructional strategies are most appropriate to foster student learning for this age group[12, 35]. Although growing, the research base and availability of appropriate teacher instruments for K-8 CS instruction is also clearly deficient in comparison to mathematics or science[13]. Research on factors that moderate teacher CS knowledge, attitudes and self-efficacy are also lacking. Research on moderating effects has focused on students, with considerable work looking at gender differences[21, 39, 42], and some work examining ethnicity[25]. This research tends to be limited to descriptive statistics describing the CS teacher force, e.g., 64% female, 75% white, 56% teaching at the high school level[19].

To help address such deficiencies, the CS for All Initiative was introduced by the U.S. Office of Science and Technology Policy[15]. This effort seeks to accelerate efforts to expand CS in K-12 schools and bring together federal agencies to support CS teacher professional development. This paper presents professional development strategies and empirical impact from one of this effort, focusing on teacher professional development for K-8 CS.

2 MATERIALS AND METHODS

2.1 Evidence-based Professional Development

The project's professional development (PD) was developed using strategies from theory and empirical research, including deepening teachers' knowledge of both content and pedagogy, active teacher engagement in learning opportunities, teacher collaboration[7, 9, 20], as well as use of didactic instruction, modeling, practice, and feedback strategies to achieve desired experiential and learning outcomes [28, 32].

The professional development was conducted in two separate years (2019 and 2020), with two separate cohorts of teachers. An integration of pedagogy and content was accomplished through two one-week summer graduate-level courses for K-8 teachers focusing on CS content (CS course) and CS pedagogy (Education course). In 2019 the content course highlighted the importance of providing the teachers with CT concepts instruction[44]. It was taught by a university computer science professor and dealt with fundamental CS topics (i.e., simple Input/Output, data structures, arrays, functions, search and sort) and computational thinking (CT) topics (i.e., decomposition, pattern recognition, abstraction, generalization, algorithm design, and evaluation). The class was supported by graduate and undergraduate students in computer science. The course used python-involved lectures, hands-on group activities, reflections, and homework assignments. The use of a higher level programming language was chosen to provide teachers with a more in-depth understanding of CS concepts and skills. However, there were also group activities based on Computational Creativity Exercises (CCE), designed to develop the teachers' CT skills through collaboration[30]. These exercises are akin to "CS Unplugged" exercises for open-ended problem solving using computational thinking and creative thinking skills[24]. The end-of-course project allowed teachers to pick one CS concept and CT topic and create a lesson for their targeted grade level.

In 2020, because of school and university closures due to COVID-19, the course was taught remotely by a high school computer science teacher using Zoom technology. The instructor had an undergraduate degree in computer science and a master's in mathematics teaching. Javascript replaced Python as the programming language. Otherwise, the content remained the same across the two years, but the 2019 course included more in-depth explanation of key CS concepts. Cohort two was supported by the CS college-level students involved in the previous year; assistance also included an elementary and middle school teacher from the 2019 cohort. Instructors set up a separate communication channel on Slack to coordinate activities in real-time and made use of the chat function to offer additional explanations and resources to complement instruction by the on-camera instructor. The course continued group activities and discussions by using Zoom breakout groups, with each group facilitated by an instructor or a teaching assistant.

The Education course in both years was taught by master elementary and middle school CS teachers and focused on CS pedagogy and how to teach the CS concepts of loops, variables, conditionals, and functions at the elementary and middle school levels. The instruction concentrated on giving teachers experiences with instructional strategies they could use in their classrooms. In 2019 the course was taught immediately after completion of the content course; in the following year the two courses were taught simultaneously, with the CS course in the morning and the education course in the afternoon to allow more content-pedagogy integration. Another significant change was moving all the computational thinking teaching from the programming class to the pedagogy class. The changes made for Zoom delivery included moving in-class activities to the discussion board on Canvas. Participants were asked to complete an activity, post a reflection, and reply to classmate reflections. This process allowed participants to take breaks from synchronous online learning throughout the day. Robots were sent to participants so they could explore and program them at home. Teachers also utilized online robot tutorials and simulations in place of having participants use robots in person.

All teachers met throughout the year as a part of a continuous improvement process to ensure that instruction was improved and adapted to new technological opportunities. These communities of practice or professional learning communities built avenues for exchanging information and strategies. This approach has been shown to be an effective method of supporting CS teachers[26, 27, 38], breaking the isolation for CS teachers who are often the only computer science teacher in their school and changing teachers' attitudes and self-efficacy[6, 37]. These were Saturday meetings, held initially face-to-face and then virtually and led by the computer science public school teachers who served as facilitators in the summer courses.

In total our teachers attended approximately 100 hours of PD for which their time was paid. This duration is in line with previous research showing that more than 45 hours is needed to show a significant effect[43].

2.2 Research Questions

The goal of the research was to determine the impact of the PD on key teacher outcomes. The study investigated two research questions:

(1) What is the impact of the PD on teacher's a) knowledge of computer science concepts and computational thinking, b) CS self-efficacy, and c) CS attitudes?
(2) Are there differential effects depending on teacher demographics (gender, years of teaching CS, geographic locale) and cohort?

2.3 Participants

Twenty-nine teachers participated in 2019 (cohort 1); 24 teachers participated in year two (cohort 2). The two cohorts were similar in terms of basic demographics (see Table 1). Most of the teachers did not have college STEM majors (57%), and 62% had not studied a programming language prior to participating in the PD. The majority of participants were female. Teachers in cohort 1 had 16 years of teaching experience and 5 years of teaching CS; cohort two had 17 years teaching experience with 7 years teaching CS. Both groups had a high percentage of master's degrees. Cohort 2 average percent of free and reduced lunch per school was slightly higher than that of cohort 1. The major differences between the two cohorts were in rural/urban and grade level distribution. Participants in 2019 were primarily from an urban district with an established K-8 CS curriculum. The remaining 11 were predominantly from rural districts in the same state. In cohort 2, most came from rural districts (38%) or towns (42%). Cohort 1 had a fairly even split between elementary and middle school teachers; cohort 2 was primarily middle school.

2.4 Instruments

Teacher knowledge of computer science was measured by two previously validated instruments that had been used in beginning undergraduate CS courses. One instrument focused on CS concepts such as selection statements, functions, and sorting[29]; the second focused on computational thinking [CTCAST: [31]]. The alpha for the computational thinking test with undergraduates was .73; for the CS concepts test it was .77. For use with the K-8 teachers in this project, alphas were .60 (CS concepts) and .51 (computational thinking).

Computer science *self-efficacy* was determined through a project-developed 22-item instrument measuring two constructs: a) self-efficacy in teaching computer science (16 items, e.g., *I can assist all students who are having trouble mastering specific programming/computer science skills*) and b) self-efficacy in their CS content and skills (6 items; e.g., *I can design and iteratively develop/refine CS programs*). Self-efficacy refers to an individual's belief in his or her capacity to execute behaviors necessary to produce specific performance attainments[2, 3]. This construct has been shown to be related to educational performance for a variety of outcomes[33], including computer science[36]. Items were rated on a 0–100% self-efficacy scale and were developed to align with objectives of each of the summer courses. Overall alpha for the instrument was .92, with alpha = .94 for self-efficacy in CS pedagogy and .91 for self-efficacy in CS content. Correlations of these two constructs was .49 providing evidence that they are related, but separate constructs.

The *attitudinal* items used a Likert scale (1: strongly disagree, 2: disagree, 3: neutral, 4: agree, 5: strongly agree) to measure two constructs a) personal interest in CS and b) the perceived value/real-world connections of CS. Sample interest questions included "I am interested in learning more about CS" and "I enjoy solving CS problems." Sample value questions included "Tools and techniques from CS can be useful in the study of other subjects" and "Having background knowledge and understanding of computer science is valuable in and of itself." The instrument was developed by adapting the *Computing Attitudes Survey*[8], which was validated with CS undergraduates. Overall alpha for both cohorts of teachers for this project was .83, with personal interest .86 and CS value .60. Correlations of these two constructs was .60, showing that they are related but separate constructs.

2.5 Research Design and Data Analysis

Following the IRB approved procedures, data were collected at three time points: pre-PD (Time 1), post PD (Time 2, average 3.26 weeks after Time 1) and end of the school year (Time 3, average 51.63 weeks after Time 1). Pre-PD represented a baseline measure taken early summer prior to the professional development; post-PD occurred on the last day of the summer PD; and time 3 was in late spring of the following school year. Research question 1 focusing on overall teacher impacts utilized a piece-wise repeated measures model looking at effects for specific time segments: pre to post summer PD (effects of summer PD) and post summer PD to end of year (effect of Saturday meetings and experience teaching CS throughout the school year). Hierarchical linear modeling[34] was performed to analyze repeated measures of teacher outcomes while accounting for nested data structure (i.e., repeated measures nested within teachers and teachers nested within districts). Separate slope terms across pre-PD to post-PD period and post-PD to follow-up period were included and estimated in the model using piecewise coding scheme suggested by Raudenbush & Byrk[34]. In addition, individually varying time points were used to account for variation in the time between assessments across teachers. Teacher cohort (2019-2020 vs 2020-2021), years of teaching (continuous), gender (male vs female), locale (urban/city vs rural/town) variables were included as covariates at timepoints for all outcomes. Then, moderation effects of each variable were examined separately by including interaction terms for only one moderator and slope term at a time.

To address the research question 2, which dealt with the moderating effects of each identified variable on the teacher outcomes, we conducted a priori planned comparisons between moderator groups (for continuous variable, scores for +.5 sd and -.5 sd group were computed for grouped comparison) for specific time segments: effects during pre-PD to post-PD period and effects during post-PD to follow-up period. Long-term (pre-PD to follow-up) effects are reported as the sum of short-term (pre-PD to post-PD) and post-PD effects. P-values were adjusted using Šidák correction method for multiple comparison to control for family-wise type-I error rates[40]. All data analysis procedures were done using SAS version 9.4 software.

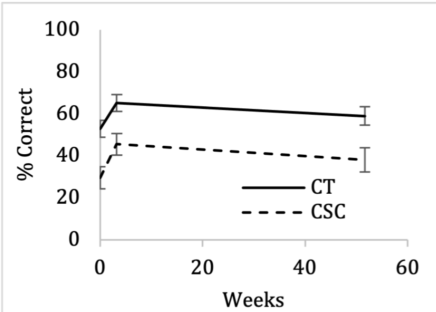

Figure 1: CS Knowledge Subscores

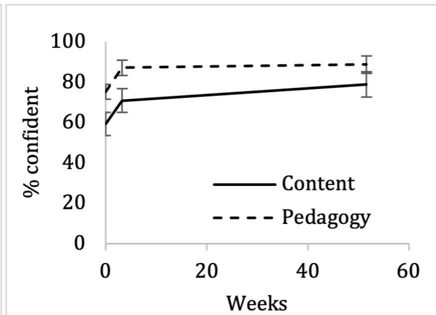

Figure 2: CS Self-efficacy Subscores

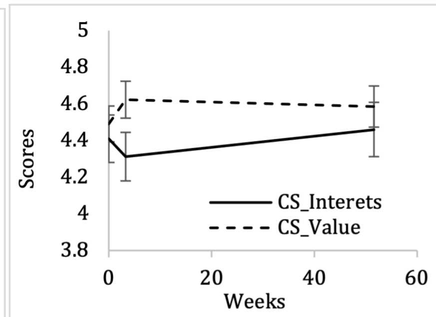

Figure 3: CS Attitudinal Subscores

3 RESULTS

3.1 Impact of Professional Development: Main Effects

3.1.1 CS Knowledge. Scores for both the computational thinking and computer science concepts assessments at all three timepoints were low (below 70%; see Figure 1). The low scores for the CS concepts test reflect the fact that it was designed to separate high performers from low performers, so instead of a C average being around 70%-80%, the average test scores were intended to be around 50%. Figure 1 shows that that the computational thinking scores were higher than those for the CS concepts test at all time points, which was confirmed by statistical tests: T1: t (200) = 9.81, p < .001, Hedges g = 1.43); T2: t (200) = 8.34, p < .001, g = 1.16; and T3: t (200) = 7.77, p < .001), g = 1.02).

Both knowledge sub scores followed similar trajectories across the three time point (Figure 1). As shown in Figure 4, there were significant gains in teachers' knowledge of both computer science concepts and computational thinking from participation in the summer PD. The computational thinking scores, however, showed a significant decrease from the end of the summer PD to the end of the school year (t (100) = -2,75, p <.05, g = -0.42). However, this final score was still significantly higher than baseline. (t (100) = 2.75, p < .05, g = 0.38). The similar decrease for the computer science concepts scores was not statistically significant.

3.1.2 Self-efficacy. As shown in Figure 2 the two self-efficacy subconstructs had similar trajectories, with significant T1-T2 gains (content: t (100) = 4.57, p < .001, g = 0.50; pedagogy: t (100) = 6.39, p < .001, g = 0.69). The teachers came into the professional development (T1) with significantly higher self-efficacy in pedagogy than content (t (186) = 7.83, p <.001, g = 0.68). This difference was maintained for the two subsequent time points (T2: t (186) = 7.84, p<.001, g = 1.05; T3: t (186) = 4.4, p <.001, g = 0.66).

3.1.3 CS Attitudes. In contrast to the results for CS knowledge and self-efficacy, the teachers' attitudes towards CS showed different pattern of results (Figure 3). Teachers had very high attitudinal ratings coming into the PD (which resulted in a ceiling effect that limited significant increases), and scores between teachers' interest in CS and their perception of CS value were not significantly different at T1 (t (186) = 1.28, p = .74). However, their perception of the value of CS was significantly higher than their interest in CS at T2 (t (186) = 4.94, p < .001, g =.61). This difference was accentuated by a decrease (nonsignificant) from T1 to T2.

3.2 Moderating Effects

Research question 2 dealt with identifying moderating effects of cohort and teacher demographics (gender, years of teaching CS, and geographic locale). Locale was defined as rural versus urban by using locale categories established by the U.S. Department of Education (https://nces.ed.gov/surveys/ruraled/definitions.asp). Years of teaching experience was divided into two groups:.5 standard deviations above the mean (more teaching experience) and .5 standard deviations below the mean (less experience).

3.2.1 Cohort. There were significant cohort moderating effects for knowledge of computational thinking and self-efficacy (both CS content and pedagogy skills). Figure 4 shows that although there was no T1 difference in computational thinking, cohort 2 increased at a significantly higher rate than cohort 1 from T1 to T2 (t (98) = 3.11, p < .05). Both cohorts showed decreases from T2 to T3; but the rate of decrease was not significantly different (t (98) = -1.79, p = .38).

Although there was no T1 cohort difference for self-efficacy (Figures 5 and 6), cohort 2 reported higher T2 scores (pedagogy: t (91) = 3.38, p < .01, g = 1.91); content/skills: t (91) = 2.63, p = .058). Cohort 2 gained self-efficacy at a higher rate from T1 to T2 (pedagogy: t (91) = 5.03, p <.001, g = 2.56; content/skills (t (91) = -1.36, p = .688.

3.2.2 Locale. There was a moderating locale effect for self-efficacy (Figures 7 and 8). Teachers from urban/city were more confident in CS than teachers from rural/town at T1 (pedagogy: t (91) = 4.1, p < .001, g = 1.05; concepts: t (91) = 3.02, p < .001, g = .86). However, rural/town teachers gained at a higher rate from T1 to T2 (pedagogy (t (91) = 3.83, p < .001. g = 1.71; concepts/skills: t (91) = 2.51, p = .08), thus reducing the difference in scores at T2 (non-significant difference). There were no significant effects in the change rate from T2 to T3, with results showing maintenance or slight increases.

3.2.3 Gender and Years of Teaching Computer Science. There were no significant moderating effects for gender or years of teaching experience; however, males had consistently higher baseline scores. In addition, teachers with less experience expressed greater interest and perception of CS value at T1 than those with more experience (interest: t (91) = 1.97, p =.27; value: t (91) = 2.69, p = .05).

4 DISCUSSION AND SIGNIFICANCE

Results of this study show that carefully structured PD that incorporates evidence-based elements (sufficient duration, teacher active learning and collaboration, modeling, practice, and feedback) can successfully impact teacher CS knowledge, self-efficacy, and perception of CS value. With the exception of CS interest, all teacher outcomes showed a significant increase due to the summer PD. The only outcome that was not positively impacted was CS interest.

The significantly higher knowledge scores for computational thinking versus computer science concepts can be expected given teachers' CS teaching experience and background knowledge. Computational thinking is a broad construct supporting many disciplines, and teachers may have been introduced to this construct in undergraduate STEM courses or PD. In contrast, the CS concepts test covered higher-level concepts that were new and difficult for the teachers. The summer CS content course, with a focus on learning a high-level programming language, was challenging to teachers. Their lack of initial knowledge (baseline average scores of 28%) contributed to the challenge of trying to learn coding. There was a strong desire on the part of teachers to see the relevance of what they were learning to how they could teach their students. For many, there was a disconnect, since Python/JavaScript would be too difficult for their students. Research on CS PD programs has found that text-based programming tools are mainly used at the high school level[26].

Despite these differences, the significant knowledge increases from baseline to post-PD provide evidence of the effectiveness of the summer PD in increasing teacher CS knowledge. This result is in line with other research showing that PD can prepare teachers to teach computational thinking and computer science concepts[14, 18, 44]. In contrast, the decrease in scores after the summer PD is in line with research that cognitive retention of new material is difficult even for highly motivated learners.

It is clear, however, that teachers need more than knowledge about computer science; they need skills and self-efficacy that they can successfully deliver CS instruction. Results showed significant increases in teacher self-efficacy in CS content and pedagogy skills after attending the PD, which aligns with previous CS self-efficacy research[16, 36]. Providing hands-on, concepts-based activities that can be utilized in the classroom appears to bolster teacher self-efficacy. The maintenance of CS knowledge and self-efficacy throughout the school year is hypothesized to be related to the ongoing professional development provided by the Saturday meetings. Focusing on classroom activities relevant to the concepts and grade levels teachers were teaching and providing avenues for teacher collaboration appears to solidify their CS knowledge and self-efficacy.

In contrast to the knowledge and self-efficacy results, the only decrease from pre- to post-PD in this study was teachers' interest in computer science, which we hypothesize was caused by teachers' reaction to the difficult and unfamiliar programming concepts presented in the CS content course. Teachers struggled with understanding a high level programming language, and this frustration likely decreased their personal interest in CS. They reported less motivation in solving Cs problems and less interest in additional learning in CS. However, this decrease was offset by an increase in interest from post PD to the end of the year. This result may be due to the Saturday meetings, which focused on classroom CS activities which teachers found more relevant than studying a programming language they did not teach at K-8.

Despite the negative impact on teacher CS personal interest, the PD increased teacher perception of the value of computer science in K-8 education. Teachers were committed to the value of CS for their students, even if they did not see CS as a discipline they personally wanted to pursue.

Overall, results show a continuing pattern of higher scores for CS pedagogy than content. Computational thinking scores were higher than CS concept scores; teacher pedagogical self-efficacy was higher than self-efficacy in content skills. In addition, interest in CS as a discipline decreased or remained static. These results suggest that teachers responded more positively to the pedagogy instruction rather than content.

In addition to these overall effects of the summer courses and follow-up Saturday meetings the research also examined moderating effects related to cohort and demographics. There were significant moderating effects for cohort and locale, but not gender and CS teaching experience. Cohort moderating effects were found for computational thinking and self-efficacy. There was no computational thinking difference at baseline, but cohort 2 had a greater rate of increase as a result of the summer PD than cohort 1. This result may be related to switching from Python to JavaScript and the change in the lead instructor for the content course from a University Professor to K-12 computer science teacher for Cohort 2. It may also be due to direct effort in teaching content and pedagogy together, with content in the morning and pedagogy in afternoon. The self-efficacy cohort moderating effect also showed no baseline differences but steeper increases in Cohort 2's self-efficacy as a result of the PD. The result could again be related to the pedagogical changes in the course. In addition, since cohort 2 experienced the courses in a remote format, these results show that CS PD can be effective when delivered virtually.

There were also differential effects depending on teachers' location of urban versus rural. This project made a concerted effort to recruit rural teachers, since rural areas are less likely to have CS classes or clubs in their schools and parents are less comfortable with computers and technology[11]. Rural districts also face unique challenges in finding qualified teachers. Urban teachers scored higher than rural at baseline in computational thinking, self-efficacy and perception of CS value. This result is likely due to the greater access of urban teachers to CS resources and learning opportunities. However, results show that with effective PD this rural/urban gap can be closed. In addition, it appears that the Saturday meetings helped maintain or slightly increase urban and rural teachers' self-efficacy level.

Although there were no significant gender moderating effects, males had higher baseline scores, which mirrors results showing that boys are more likely to express interest in CS and learn programming to create software, apps, games, websites[10]. What is important, however, is that the baseline gender knowledge gap was reduced through carefully designed PD.

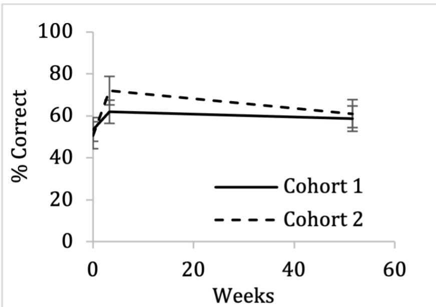

Figure 4: Computational Thinking

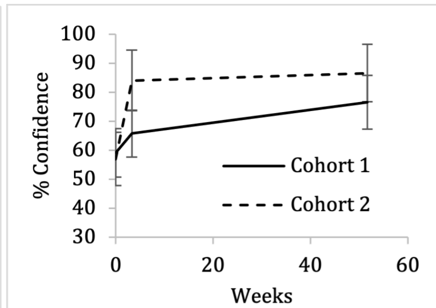

Figure 5: Content Self-efficacy

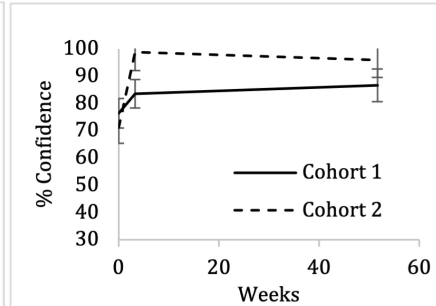

Figure 6: Pedagogy Self-efficacy

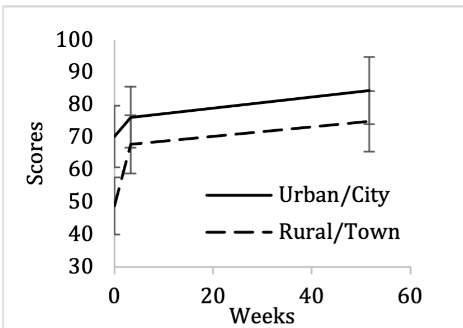

Figure 7: Content self-efficacy

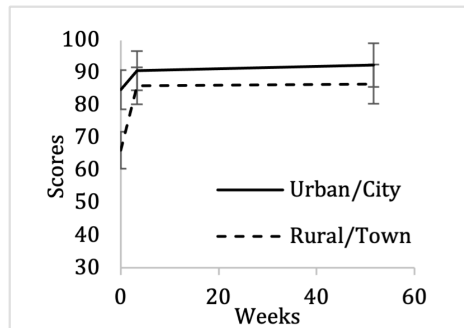

Figure 8: Pedagogy self-efficacy

There were also no significant moderating effects for experience in teaching CS. The baseline difference in favor of those with less experience and exposure to CS suggests that they were perhaps more excited and interested in CS as a new teaching venture, and thus ascribed more value and interest at baseline. The T2 and T3 results followed trends of the main effects, with decreases in T2 and maintenance or increases in scores at T3.

Overall, the moderating results showed that the PD was successful with experienced and less experienced teachers, with teachers from both rural and urban locales, and with both males and females. Of note is that these effects were obtained over two separate years of summer professional development sessions – one using face-to-face delivery and one using remote delivery via Zoom. The PD was successful in in-person as well as virtual delivery environments, showing that, with careful planning, CS PD can be delivered remotely. Maintenance of the self-efficacy and attitudinal scores was fostered by the Saturday meetings, which focused on CS classroom activities and resources at the various grade levels. The one exception to this trend was the cognitive knowledge outcomes, which was not the emphasis of the Saturday meetings, and which research has shown tends to decline with time without periodical review or rehearsal[1].

This study provides insight regarding earlier research which concluded that traditional college-level practices should not automatically be used in K-8 environments and that the field has not yet achieved a solid body of K-8 CSEd research[4]. Our research supports this conclusion in several ways; first, it was critically important to have experienced CS K-12 teachers, as well as University CS faculty, supporting the PD. The cadre of undergraduate CS students was also helpful for teachers who were struggling with the CS content and needed one-on-one help.

Results also suggest that the three instruments used in this project that were developed and validated with college-level audiences (both knowledge tests and the personal interest items) may not be appropriate for K-8 teachers. The alpha levels for the knowledge assessments used in this study were low (.60 and .51), and the low scores across all three time points may have contributed to teacher anxiety, negatively impacting their attitude towards CS. In addition, while the personal interest questions in the CS attitude scale may be appropriate for CS undergraduates, their usefulness in K-8 appears to be affected by level of the CS content covered in the PD.

Nevertheless, since CS represents a new subject in public schools and requires new pedagogical approaches, research studies that focus on elements of effective PD, explore differential effects depending on context and teacher demographics, and test new and adapted instruments for use in K-8 are critical. This study contributes to the current research base on K-8 CS by (a) determining specific effects of summer PD and follow-up Saturday meetings, (b) isolating critical moderating effects, and (c) development and utilization of new and adapted instrumentation.

REFERENCES

[1] Harry P Bahrick. 1984. Semantic memory content in permastore: fifty years of memory for Spanish learned in school. *Journal of Experimental Psychology: General* 113, 1 (1984), 1.

[2] Albert Bandura. 1977. Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review* 84, 2 (1977), 191. https://doi.org/10.1037/0033-

295X.84.2.191

[3] Albert Bandura. 1997. *Self-efficacy: The exercise of control*. W H Freeman, New York, NY.

[4] Paulo Blikstein and Sepi Hejazi Moghadam. 2018. Pre-college computer science education: A survey of the field. (2018). https://services.google.com/fh/files/misc/pre-college-computer-science-education-report.pdf

[5] Code.org, CSTA, and ECEP Alliance. 2020. *2020 state of computer science education: Illuminating disparities*. https://advocacy.code.org/2020_state_of_cs.pdf

[6] Quintin Cutts, Judy Robertson, Peter Donaldson, and Laurie O'Donnell. 2017. An evaluation of a professional learning network for computer science teachers. *Computer Science Education* 27, 1 (2017), 30–53. https://doi.org/10.1080/08993408.2017.1315958

[7] Linda Darling-Hammond, Ruth Chung Wei, Alethea Andree, Nikole Richardson, and Stelios Orphanos. 2009. Professional learning in the learning profession. *Washington, DC: National Staff Development Council* 12 (2009). http://www.learningforward.org/docs/pdf/nsdcstudy2009.pdf

[8] Brian Dorn and Allison Elliott Tew. 2015. Empirical validation and application of the computing attitudes survey. *Computer Science Education* 25, 1 (2015), 1–36. https://doi.org/10.1080/08993408.2015.1014142

[9] Michael S Garet, Andrew C Porter, Laura Desimone, Beatrice F Birman, and Kwang Suk Yoon. 2001. What makes professional development effective? Results from a national sample of teachers. *American Educational Research Journal* 38, 4 (2001), 915–945. https://doi.org/10.3102/00028312038004915

[10] Google and Gallup. 2016. Computer Science Learning: Closing the Gap: Girls. https://services.google.com/fh/files/misc/computer-science-learning-closing-the-gap-girls-brief.pdf

[11] Google and Gallup. 2017. *Computer Science Learning: Closing the Gap: Rural and Small-Town Districts*. https://goo.gl/hYxqCr

[12] Shuchi Grover, Roy Pea, and Stephen Cooper. 2015. Designing for deeper learning in a blended computer science course for middle school students. *Computer science education* 25, 2 (2015), 199–237. https://doi.org/10.1080/08993408.2015.1033142

[13] Halil Ibrahim Haseski and Ilic Ulas. 2019. An investigation of the data collection instruments developed to measure computational thinking. *Informatics in Education* 18, 2 (2019), 297–319. https://doi.org/10.15388/infedu.2019.14

[14] Emily Hestness, Diane Jass Ketelhut, J Randy McGinnis, and Jandelyn Plane. 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 (2018), 411–435. https://www.learntechlib.org/primary/p/181431/

[15] White House. 2016. FACT SHEET/ A Year of Action Supporting CS for All. https://obamawhitehouse.archives.gov/the-press-office/2016/12/05/fact-sheet-year-action-supporting-computer-science-all

[16] Erdogan Kaya, Ezgi Yesilyurt, Anna Newley, and Hasan Deniz. 2019. Examining the impact of a computational thinking intervention on pre-service elementary science teachers' computational thinking teaching efficacy beliefs, interest and confidence. *Journal of Computers in Mathematics and Science Teaching* 38, 4 (2019), 385–392.

[17] Jiyoung Kim, Joshua Childs, Anne Leftwich, Kendra Montejos Edwards, Carol L Fletcher, and Katie Hendrickson. 2021. Landscape of Computer Science Teacher Qualification Pathway. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 1299–1299. https://dl.acm.org/doi/pdf/10.1145/3408877.3439635

[18] Siu-Cheung Kong, Ming Lai, and Daner Sun. 2020. Teacher development in computational thinking: Design and learning outcomes of programming concepts, practices and pedagogy. *Computers & Education* 151 (2020), 103872. https://doi.org/10.1016/j.compedu.2020.103872

[19] Sonia Koshy, Alexis Martin, Laura Hinton, Allison Scott, Bryan Twarek, and Kalisha Davis. 2021. The computer science teacher landscape: Results of a nationwide teacher survey. *Retrieved (May 23, 2021) from https://csteachers. org/page/cs-teacher-landscape* (2021). https://www.kaporcenter.org/the-computer-science-teacher-landscape-results-of-a-nationwide-teacher-survey

[20] Susan Loucks-Horsley, Katherine E Stiles, Susan Mundry, Nancy Love, and Peter W Hewson. 2009. *Designing professional development for teachers of science and mathematics*. Corwin press, Thousand Oaks, CA.

[21] Allison Master, Sapna Cheryan, and Andrew N Meltzoff. 2016. Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science. *Journal of educational psychology* 108, 3 (2016), 424. https://psycnet.apa.org/manuscript/2015-37516-001.pdf

[22] Lowell Matthews Jr and Pat Yongpradit. 2019. Addressing America's Growing Demand for Information Technology and Computer Science: The Case for Change in K-12 Education. *Foundation for Excellence in Education* (2019). https://www.excelined.org/wp-content/uploads/2019/06/ExcelinEdCode.AddressingAmericasGrowingDemandforITandCS.TheCaseforChangeinK12Ed.June2019.pdf

[23] Microsoft. 2018. *It's Computer Science Education Week, and we're here to help you make it great!* https://blogs.microsoft.com/latinx/2018/12/05/its-computer-science-education-week-and-were-here-to-help-you-make-it-great/

[24] Lee Dee Miller, Leen-Kiat Soh, and Markeya S Peteranetz. 2019. Investigating the Impact of Group Size on Non-Programming Exercises in CS Education Courses. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. Minneapolis,MN,USA, 22–28. https://doi.org/10.1145/3287324.3287400

[25] NCCWIT. 2019. Bridging the Encouragement Gap in Computing. https://ncwit.org/resource/practicingencouragement/

[26] Lijun Ni, Gillian Bausch, and Rebecca Benjamin. 2021. Computer science teacher professional development and professional learning communities: a review of the research literature. *Computer Science Education* (2021), 1–32. https://doi.org/10.1080/08993408.2021.1993666L

[27] Lijun Ni, Mark Guzdial, Allison Elliott Tew, Briana Morrison, and Ria Galanos. 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*. 553–558. https://doi.org/10.1080/08993408.2021.1993666

[28] V Darleen Opfer and David Pedder. 2011. Conceptualizing teacher professional learning. *Review of Educational Research* 81, 3 (2011), 376–407. https://doi.org/10.3102/0034654311413609

[29] Markeya S Peteranetz and Anthony D Albano. 2020. Development and evaluation of the Nebraska Assessment of Computing Knowledge. *Frontiers in Computer Science* 2 (2020), 11. https://doi.org/10.3389/fcomp.2020.00011

[30] Markeya S Peteranetz, Abraham E Flanigan, Duane F Shell, and Leen-Kiat Soh. 2018. Helping engineering students learn in introductory computer science (CS1) using computational creativity exercises (CCEs). *IEEE Transactions on Education* 61, 3 (2018), 195–203.

[31] Markeya S Peteranetz, Patrick M Morrow, and Leen-Kiat Soh. 2020. Development and validation of the computational thinking concepts and skills test. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. Portland,OR,USA, 926–932. https://doi.org/10.1145/3328778.3366813

[32] Robert C Pianta. 2005. A new elementary school for American children. *SRCD Social Policy Report* 19 (2005), 4–5.

[33] Paul R Pintrich, David AF Smith, Teresa Garcia, and Wilbert J McKeachie. 1993. Reliability and predictive validity of the Motivated Strategies for Learning Questionnaire (MSLQ). *Educational and Psychological Measurement* 53, 3 (1993), 801–813. https://doi.org/10.1177/0013164493053003024

[34] Stephen W Raudenbush and Anthony S Bryk. 2002. *Hierarchical linear models: Applications and data analysis methods*. Vol. 2. sage, London,England.

[35] Kathryn Rich, Carla Strickland, and Diana Franklin. 2017. A literature review through the lens of computer science learning goals theorized and explored in research. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. Seattle, WA, 495–500. https://doi.org/10.1145/3017680.3017772

[36] Peter J Rich, Stacie L Mason, and Jared O'Leary. 2021. Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. *Computers & Education* 168 (2021), 104–196. https://doi.org/10.1016/j.compedu.2021.104196

[37] Jean Ryoo, Joanna Goode, and Jane Margolis. 2015. It takes a village: Supporting inquiry-and equity-oriented computer science pedagogy through a professional learning community. *Computer Science Education* 25, 4 (2015), 351–370. https://doi.org/10.1080/08993408.2015.1130952

[38] Robert Schwarzhaupt, Joseph Wilson, Fanny Lee, and Melissa Raspberry. 2021. Teachers' engagement and self-efficacy in a PK–12 computer science teacher virtual community of practice. *Journal of Computer Science Integration* (2021). https://doi.org/10.26716/jcsi.2021.10.8.34

[39] U.N. Educational Scientific and Cultural Organization. 2017. Cracking the code: Girls' and women's education in science, technology, engineering and mathematics (STEM). https://unesdoc.unesco.org/ark:/48223/pf0000253479

[40] Zbyněk Šidák. 1967. Rectangular confidence regions for the means of multivariate normal distributions. *J. Amer. Statist. Assoc.* 62, 318 (1967), 626–633. https://doi.org/10.1080/01621459.1967.10482935

[41] Jim Stanton, Lynn Goldsmith, Richards W Adrion, Sarah Dunton, Katie A Hendrickson, Alan Peterfreund, Pat Yongpradit, Rebecca Zarch, and Jennifer D Zinth. 2017. Landscape Report: State-Level Policies Supporting Equitable K-12 Computer Science Education. https://www.edc.org/sites/default/files/uploads/State-States-Landscape-Report.pdf

[42] Jennifer Wang, Hai Hong, Jason Ravitz, and Sepehr Hejazi Moghadam. 2016. Landscape of K-12 computer science education in the US: Perceptions, access, and barriers. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. Memphis, TN, USA, 645–650. https://doi.org/10.1145/2839509.2844628

[43] Ruth Chung Wei, Linda Darling-Hammond, and Frank Adamson. 2010. Professional development in the United States: Trends and challenges. (2010). https://edpolicy.stanford.edu/sites/default/files/publications/professional-development-united-states-trends-and-challenges.pdf

[44] Aman Yadav, Sarah Gretter, Jon Good, and Tamika McLean. 2017. Computational thinking in teacher education. In *Emerging research, practice, and policy on computational thinking*. Springer, 205–220.