Building Early Elementary Teacher Confidence in Teaching Computer Science Through a Low-Cost, Scalable Research-Practitioner Collaboration

Justin Lee Clough, University of Southern California

Justin L. Clough is a PhD student at the University of Southern California studying Mechanical Engineering; his advisor is Assad A. Oberai. He received his Bachelors of Science from the Milwaukee School of Engineering and Masters of Engineering from Rensselaer Polytechnic Institute, both in Mechanical Engineering. He holds a DOD:SMART scholarship and works closely with the AFRL/RQHV teams at Wright-Patterson AFB.

Patricia Chaffey, University of Southern California

Patricia Chaffey has had a passion for studying and designing interaction between humans and technology since her undergraduate career at Mount Holyoke College, and continues to pursue this interest at the University of Southern California. Some of her notable work includes developing a robotic learning companion and designing a simulation to study how people interact with swarms of robots using a virtual agent as an intermediary. Patricia has received awards to support her travel to conferences and leadership workshops, which include, but are not limited to, the 2018 ELIS Expanding Horizons award, and the 2017 Computing Research Association – Women Grace Hopper Celebration Research Scholar award. Patricia has participated in a number of publications across the different labs she has been active in. When not in the lab, Patricia enjoys volunteering with BOTS (Building Opportunities with Teachers in Schools), where she works with elementary teachers and their students on robotics.

Gautam Salhotra, University of Southern California

Gautam Salhotra is a PhD student in the Computer Science department at the University of Southern California. He works with Dr. Gaurav Sukhatme at the Robotics and Embedded Systems Lab (RESL). His research focusses on machine learning, robotic manipulation, and techniques for robot task planning.

Colin G Cess, University of Southern California

Colin G. Cess is a PhD student in the Biomedical Engineering department at the University of Southern California. His advisor is Stacey D. Finley. He received his Bachelors of Science from the University at Buffalo in Pharmaceutical Sciences. His research focuses on interactions between tumor cells and immune cells.

Rey Pocius, University of Southern California

I am a PhD student at the University of Southern California, where I am working with Stefanos Nikolaidis in the ICAROS (Interactive and Collaborative Autonomous Robotic Systems) lab. My research spans human-robot interaction, haptics, and decision-making under uncertainty. My research is on shared autonomy for human-robot collaboration.

Dr. Katie Mills, University of Southern California

Building Early Elementary Teacher Confidence in Teaching Computer Science Through a Low-Cost, Scalable Research-Practitioner Collaboration

Abstract

In a world increasingly impacted by artificial intelligence and computer systems, there is an urgent need to target under-resourced districts where early elementary in-service teachers may not have had exposure to teaching computer science. These teachers benefit from support to develop computer science literacy in students, especially when robotics is used as physical computing in first- and second-grade classrooms. Studies show that students as young as four years old can build and program simple robots; furthermore, exposure to robotics improves both students' computational thought and creative ability. The Los Angeles Unified School District (LAUSD) Tech Force identified teacher intervention as its target; this prompted a research-practitioner collaboration between nine teachers in three elementary schools and a researcher group at a university in Los Angeles called Building Opportunities with Teachers in Schools (BOTS) to address this challenge. BOTS is a low-cost, scalable solution that focuses on improving the teachers' confidence in teaching computer science through robotics, in a partnership between teachers from the Latinx Boyle Heights area, university K-12 outreach professionals and Ph.D. student volunteers. University staff organized the tools and support useful to integrate robotics into teachers' in-school curricula in regular professional development workshops spanning multiple years. Using coding curricula from Code.org, Sphero SPRK+® robots as hardware, and several non-computer-based logical activities, the teachers have developed their own activities by adapting what they have learned in the BOTS Professional Development sessions to the needs of their students. Combining their teaching experience with the diverse technical knowledge of the university students and outreach staff allowed for a novel approach in increasing the technological literacy of elementary school students by targeting their teachers. BOTS provides educators with (i) the opportunity to continually expand their own self-efficacy in teaching robotics and (ii) long term support ensuring that the teachers remain self-assured in integrating new material. Results from the pilot year show that 100% of the participating teachers agree BOTS increased their confidence in teaching coding, and 75% agree that BOTS added value to their classroom instruction. Additionally, 100% reported that coding improved their students' problem-solving, communication, and creativity. All but one of the teachers have continued into the second year of the BOTS program.

Introduction

In a world increasingly impacted by computer systems and Artificial Intelligence (AI), all children need to gain familiarity with Computer Science (CS) so they can develop at minimum a basic understanding of the ways that AI, data analytics, and machine learning affect their lives. Familiarity or fluency in CS will help children retain agency and discernment while growing up with the increasingly complex computer systems of the "4th Industrial Revolution"[1]. Many schools and school districts recognize the need for introducing age-appropriate CS and robotics as young as in pre-kindergarten. Researchers confirm the value of this move, notably led by the pioneering work of Marina Umanschi Bers and her colleagues as well as a growing number of researchers in various countries. These scholars study ways to integrate CS in early education through various means; these include development of age-appropriate robotics kits, pre-service and/or in-service teacher training, afterschool robotics or CS programs provided by mentors who may or may not be the students' teachers, or in-class activities such as those provided by the national nonprofit, Code.org.

In general, working with robotics also allows children to learn-by-doing, which has been shown to better integrate the lesson in the child's mind[2,3,4]. STEM interventions inherently teach children basic skills such as analytical abilities, creative abilities, logical thought, and teamwork[5]; these skills are all necessary for academic achievement. Many studies have focused on the beneficial impact of brief exposures to a robotics curriculum, through either intensive week-long workshops or as an addition to their weekly coursework over a period of several weeks[6,7]. Kazakoff et al. found that a one-week workshop of programming robots enables students to improve story sequencing skills, showing that the skills learned from working with robots are able to spill over into language comprehension[8]. Sullivan and Bers found that longer, but less intensive, exposure over an eight-week period can greatly increase children's understanding of robotics and programming[7]. Many other workshops of this type have found similar results[2,9,10].

Yet in the nation's inner city schools, the introduction and integration of computer science can be slow and unequal compared to schools in more affluent portions of a district. In Los Angeles Unified School District (LAUSD), for instance, Jane Margolis and colleagues demonstrated in 2010 that an unnamed high school in east Los Angeles, near or in the Boyle Heights region, had sufficient computer technology but lacked CS courses that required critical thinking. Since Margolis et al.'s landmark study, LAUSD has made progress in addressing this disparity. In 2016, LAUSD's Instructional Technology (IT) Task Force recommended "[p]rofessional learning opportunities for all stakeholders [are] imperative to educate leaders on how to incorporate digital learning tools and how to adapt instruction to the opportunities afforded by digital tools"[11]. Since 2019, LAUSD has become the first district in the nation to adopt ISTE Student Standards and to join forces with California Emerging Technology Fund for a multi-million dollar partnership aimed at closing the digital divide[12]. The University of Southern California (USC) is a university that develops and implements K-12 STEM outreach programs in central and east Los Angeles, often in close partnership with these local sub-districts within LAUSD, the university's K-12 STEM Center has sought to foster solutions to the slow uptake of CS in Boyle Heights elementary schools. The mission of the Center is to inspire, inform, and impact underrepresented, under-served and disadvantaged K-12 students to develop STEM identity. The goal is to increase lifelong participation in STEM by providing community-oriented and research-based educational opportunities. The university has a number of faculty researchers in CS and robotics as well as students, at both the undergraduate and graduate levels, with experience in providing or participating in grade-school level robotics as a fun, hands-on way to learn basic concepts of CS. In keeping with the recommendations of LAUSD's IT Task Force to adapt instruction to utilize the unique pedagogical opportunities of digital tools, the K-12 STEM Center has partnered with LAUSD's Local District East and the Los Angeles Archdiocese (ADLA) to help teachers in three Boyle Heights elementary schools to bring CS and robotics into their weekly in-class lessons.

The key purpose of this intervention was to target in-service teachers who were well experienced in teaching inner city first- and second-graders but who had no/little experience teaching CS. The Center had previously provided an afterschool, constructivist program in different schools in central LAUSD taught by USC students to bring robotics directly to upper elementary students; while this near-peer mentoring has many beneficial outcomes, the elementary school teachers did not learn CS or robotics and thus remained dependent on external providers to teach this material. This time, the Center wanted to empower teachers so that the schools need no longer outsource STEM teaching to costly third-party vendors or to volunteers with intermittent commitment or funding.

Besides being unfamiliar with basic programming concepts, many teachers need help integrating these concepts and activities into the core curriculum. Hamner et al. describe Professional Development (PD) sessions aimed at providing teachers ways to incorporate engineering and CS concepts into non-technical courses[13]. When teachers blend technology with traditional subjects such as math and language arts, students find creative and novel ways to utilize engineering and computer science. Through teacher surveys, Hamner et al. found that PD sessions improved teacher confidence in implementing robotics and programming projects into their classrooms.

Cortina and Trahan describe a five-day workshop aimed at providing teachers ways to incorporate CS into their classrooms without having to make major changes to their curricula. For example, a math teacher could use a short computer program to display a geometric principle and then briefly talk about how the program works. As teachers already have large workloads and very little time, this allows them to still introduce computer science into their classrooms without compromising their main subjects. It also enables students to learn about how computer science can directly be applied to the subjects that they are studying[14].

While benefiting from education research such as that by the aforementioned principal investigators, the Center sought to blend scholarly findings with the deep experience of teachers who have proven success in the targeted inner city schools. In 2018, the K-12 STEM Center began exploring the CS need in early elementary students in Boyle Heights and ways to create a low-cost, scalable intervention grounded in constructivist pedagogy. Principals in two LAUSD elementary schools gave input as did the principal of a nearby Catholic school, all three serving similar low income, largely immigrant, Latinx communities in relatively close proximity. Bringing together teachers in the Los Angeles Archdiocese and LAUSD seemed beneficial, as the Center sought to address the way the Latinx community was organized rather than the institutional divides between public and parochial school systems. The K-12 STEM Center was inspired in this project by the collaborative nature of Research-Practitioner Partnerships (RPP) which are “partnerships composed of research institutions and state or local education agencies that have identified an education issue or problem of high priority for the education agency that has important implications for improving student education outcomes. These partnerships are to carry out initial research and develop a plan for future research on that education issue,” as defined by the National Center for Educational Research[15].

The goal was to value and foreground the teaching skills of in-service teachers in the partner schools with the insights available to us as a Center working with CS researchers as well as with multiple Boyle Heights schools within both LAUSD and the Los Angeles Archdiocese. The Center was also able to use the experience of university students who started learning CS in elementary school through the physical computing of robotics. Although inspired by the RPP model, this partnership might best be described as a Community of Practice in which equal value is placed on in-service teachers, Ph.D. students in CS and engineering majors, and students with a history of robotics learning experiences in elementary schools, all supported by the staff, infrastructure, and grants of the K-12 STEM Center. External independent assessment has been provided by STEM Program Evaluation, Assessment, and Research (SPEAR) consultants with substantial experience in evaluating education interventions such as BOTS.

Called Building Opportunities with Teachers in Schools (BOTS), the collaborators aimed to design a low-cost, scalable solution that focuses on improving the teachers’ confidence in teaching computer science through robotics. Presented here are the preliminary results of the pilot begun in 2018 as a Community of Practice between nine teachers in three elementary schools in the Latinx Boyle Heights area, K-12 Center outreach professionals, six Ph.D. student volunteers and two undergraduate students with additional occasional volunteers provided by the student organization *Robogals*. The K-12 STEM Center organized the tools and support used to integrate robotics into teachers’ in-school curricula in monthly Professional Development workshops. In parallel to the aforementioned goal was the question: Can BOTS boost in-service, inner city teacher ability and self-confidence to teach coding and introduce robots as authentic, real-world digital learning opportunities?

Methods

A long-term objective of BOTS is to develop a low-cost, sustainable series of PD sessions that could scale to any school that wanted to download the learning goals and activities from the STEM Center’s website. Thus, for this pilot program, one goal was to select curricula and equipment that would minimize costs. Two main tools were selected: Code.org, for introducing basic CS concepts, and Sphero SPRK+® robots, for providing hands-on experience with robotics.

Code.org is a free resource that can be used to teach foundational CS concepts through easy-to-understand block programming. It is designed to empower in-service teachers who have little to no experience in CS, promoting the concept that the teacher is the “lead learner” among the students where everyone must learn-by-doing. Besides computer-based activities, Code.org also provides many “unplugged” activities that do not require a computer to learn key computational thinking concepts. This allows teachers to stay with the BOTS lessons on days when they may not have access to tablets, computers, or other equipment shared between classrooms. A beneficial aspect of Code.org is its online dashboard, which allows teachers to monitor their students' progress throughout the lessons.

The Sphero SPRK+® robots were selected due to their durability, ability to engage the students, and cost, which was on average $100 per unit at the time of the launch of the BOTS program. They are able to be programmed via block coding or JavaScript, thus allowing the same robots to be used as students progress to the more advanced JavaScript programming language in upper grades. Using block coding, the students transferred skills learned on screen in the Code.org activities to navigate mazes with the robots. A sample maze is shown near the bottom of Figure 1. With this transition to the physical world, students had to apply the engineering design process, first understanding the problem, imagining a solution, implementing the solution, and finally analyzing the results. The teachers first experienced at the PD sessions, and then emphasized to their students, that planning, trial and error are key to CS and robotics.

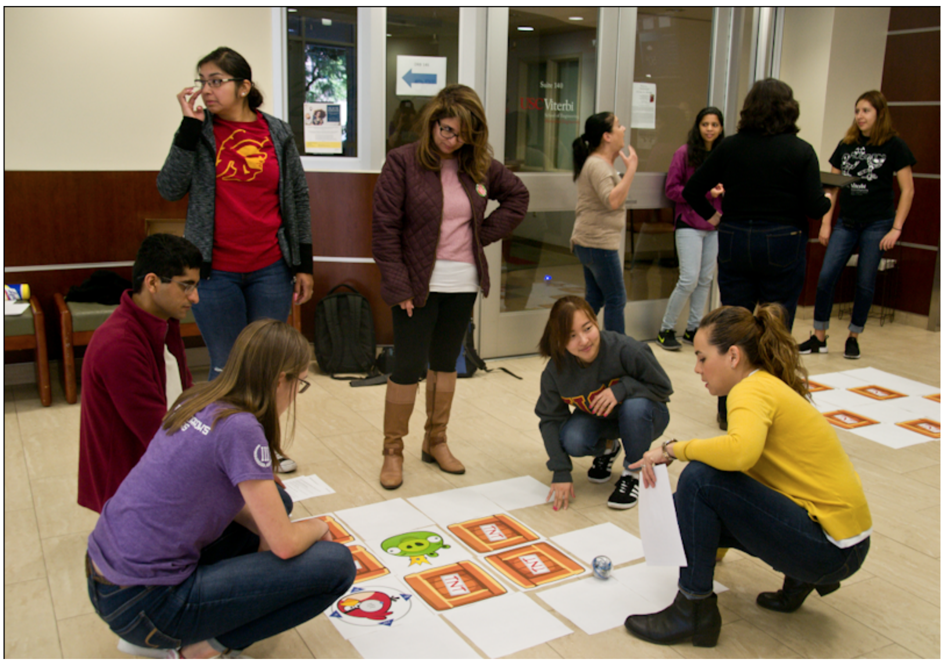

Figure 1: BOTS teachers, Ph.D. student and undergraduate volunteers plus Center student staff use print-outs adapted from the online Code.org mazes to create similar mazes on the floor so they can learn to program the Sphero SPRK+® robots by adding speed, duration, and direction (as angles) to their block coding.

While the Sphero robots retained the block programming learned in the Code.org activities, they presented new challenges. Besides just direction of movement, students have to include the speed of the robot and the duration of travel. Direction was also coded in term of angles relative to the robot's starting position, as opposed to simple directions such as "left," "right," and "forward." Although angles are not generally taught before fourth grade, the BOTS teachers' first- and second-grade students readily learned angles in their eagerness to program the robots. Besides being exposed to advanced mathematical material, the teachers were able to use this to bridge into other academic areas. To help their students learn to identify angles, teachers used sentence word gaps (shown in Figure 2), a technique commonly used in teaching language arts.

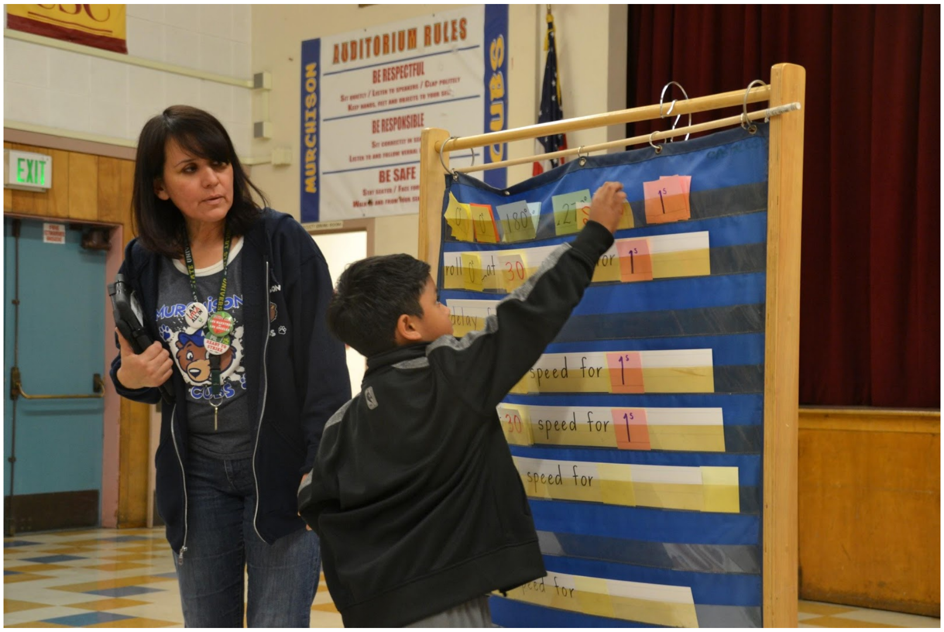

Figure 2: A first-grade teacher at an elementary school, using the sentence word gaps to teach early elementary students the angles they need to program the direction of the robots.

BOTS consisted of a series of progressive PD sessions held for three hours on a Saturday morning almost monthly throughout the academic year. Each session built upon the previous and gave the teachers the opportunity to reflect and receive feedback on their classroom implementation of the previous session's activities. Teachers were paid a training stipend for each session. Funding for this stipend came from the USC good neighbors campaign. Six Ph.D. students as well as several undergraduate students volunteered their time to assist in helping the teachers with the practical parts of the sessions. All other student and professional staff were employed by the K-12 STEM Center. Independent external assessment was provided by SPEAR consultants. This consisted of pre- and post-workshop surveys to determine if the PD session accomplished its learning goals for content knowledge as well as to measure changes in teacher confidence and self-efficacy.

Results

Each session had attendees fill out pre-session and post-session surveys, where questions pertaining to confidence and knowledge in the material were asked. Likert-scale style questions relating to knowledge were presented as a statement, with attendees selecting the statement most closely aligned with their knowledge on a 5-point scale. Scales ranged from "Not knowledgeable at all" to "Extremely knowledgeable" and "Strongly disagree" to "Strongly agree". In the following subsections, a breakdown of the teacher responses at each workshop is detailed and calculations showing differences presented. For all sessions, the average values before and after the session as presented as well as percent difference.

Session 1

An overall growth can be seen across various coding related skills. In general, growth in confidence was highest amongst the activities relating directly to the workshop tasks (e.g., connecting concepts of direction, speed, and time to where the physical robot will move, and then actively coding those tasks), with additional levels of growth achieved on the topics of algorithms and persistence in problem solving. As shown in Table 1, teachers left their workshop with higher confidence in their knowledge of coding concepts.

Table 1: Average scores before and after, and percent difference on knowledge of topics from PD session 1.

Prompt	Pre	Post	Difference
Block-based programming actions (such as clicking, drag and drop)	2.67	3.78	41.7%
Teaching algorithms as it relates to coding	1.78	3.22	81.3%
Teaching 3-D dimensions of robotic movement (direction, movement (speed), and duration (time))	1.44	3.33	130.8%
Teaching and coding Sphero robots to travel a maze	1.33	3.22	141.7%
Teaching persistence and problem solving as it relates to robots and coding	1.78	3.44	93.8%
Teaching using Code.org's "On-line Puzzles"	2.22	3.56	60.0%

The data in Table 2 relates to session 1 where teachers were asked to briefly describe their levels of agreement with statements relating to what level of impact coding and computational thinking had on other important skills. While teachers were receptive to the idea of programming being important as a tool, they did not strongly feel that it was an essential one in the STEM fields.

Table 2: Average scores before and after, and percent differnce, on impact of coding from PD session 1.

Statement	Pre	Post	Difference
Coding improves persistence	3.33	4.00	20.0%
Coding improves problem solving	3.44	3.89	12.9%
Coding improves communication	3.22	3.89	20.7%
Coding improves creativity	3.22	3.89	20.7%
Coding and programming improve collaboration	3.22	3.89	20.7%
Programmers are in high demand	3.56	3.89	9.4%
Coding is an essential skill for STEM and other future career opportunities	3.44	3.89	12.9%

Session 2

In the second workshop session, teachers were asked to share their levels of understanding on some of the skills and strategies for building a functioning program. In general, there was positive change across all sectors. Notably, the area where teachers had the lowest average initial understanding (and subsequently, the highest level of growth) related to the practice of pseudocoding. Data collected and processed is shown in Table 3.

Table 3: Average scores on understanding before and after PD session 2 along with percent difference.

Prompt	Pre	Post	Growth
I understand pseudocoding is a necessary step for discussing and planning how to code the sequence of commands BEFORE coding them	3.00	4.33	44.4%
In the unplugged activity, I can use arrows to code maze solutions	3.44	4.00	16.1%
In the unplugged activity, I can use Code.org block paper cut-outs to code maze solutions	3.78	4.44	17.6%
In the plugged activity, I can use Sphero block paper cut-outs to code maze solutions	3.89	4.33	11.4%
I am able to code the Sphero to run through two mazes with help	3.33	4.22	26.7%
I am able to code the Sphero to run through a third maze without help from a university mentor	3.56	3.89	9.4%

Teachers were then asked to rate their confidence in teaching these topics to their students. Although there was growth across all categories, the teachers were on average less confident initially in the topics of introducing movement dimensions (e.g., speed, duration) and teaching students to code through the mazes. This initial hesitancy may stem from either the teachers' initial uncertainty with the topics at hand, or may originate in the teachers' initial assessment of what they felt their students could understand at their current ages. Despite the initial uncertainty, their confidence had increased by the end of the workshop. These data are presented in Table 4.

Table 4: Processed data from PD session 2 on teachers' confidence; average scores before and after session along with percent difference.

Statement	Pre	Post	Growth
Teaching student dyads (using pair programming) how to use pseudocode as a necessary step of discussing and planning how to code the sequence of commands BEFORE they code the activity	3.22	4.33	34.5%
Teaching student dyads in an unplugged activity how to use arrows to code maze solutions	3.44	4.00	16.1%
Teaching student dyads in an unplugged activity how to Code.org block cut-outs to code maze solutions	3.33	4.00	20.0%
Teaching student dyads the Sphero blocks, with the added dimension of speed and duration as well as angles for direction compared to Code.org's blocks	2.78	4.00	44.0%
Teaching student dyads to use Sphero block code on the Sphero community app, transitioning from the unplugged activity with paper cut-out blocks to the Sphero code	3.11	4.11	32.1%
Teaching student dyads to code the Sphero to run through the two mazes	2.67	3.78	41.7%

Session 3

In the third workshop, teachers were asked to discuss what changes they had noticed in their students' computational thinking skills as a result of the BOTS curriculum. Teachers reported that their students were more confidant with their abilities, and were willing to take risks and try new strategies in pursuit of solving the coding mazes. One teacher noted that "Students are very comfortable using code.org apps and transitioned to Sphero easily," while another mentioned that "My students are able to collaborate with each other and work on their own. They are no longer relying on me to give them the answers." Teachers were also asked to evaluate their confidence in introducing the new material to their students; the aggragated data from their responses are presented in Table 5.

Table 5: Average before and after scores for PD session 3 along with percent difference regarding confidence teaching select topics.

Please rate your confidence in teaching the following skills to your students before and after attending today's workshop	Pre	Post	Growth
I can explain the differences and bridging activities between Sphero coding and Code.org coding	3.50	4.83	38.1%
I can perform and teach block-based coding in the directional/angular, Code.org, and Sphero domains	3.17	4.50	42.1%
I understand and can utilize scaffolding techniques of directional/angular block codes used by Code.org coding to inform and enhance Sphero coding	3.33	4.33	30.0%
I can access Sphero coding templates and solutions via the University profile on edu.sphero.com	3.33	3.83	15.0%
I can define and feel comfortable teaching the words "loop" and "iteration" or "repeat"	3.83	4.83	26.1%
I can identify when and where to use a loop structure instead of using manual repetition	3.50	5.00	42.9%
I can break down a long sequence of instructions into the largest repeatable sequence	3.67	4.67	27.3%
I can successfully integrate BOTS into my classroom curriculum	3.67	4.33	18.2%

Session 4

In the fourth workshop, teacher were asked to revisit questions from the first workshop to examine long term growth. In questions relating to how coding improved student persistence, problem solving, and communication, all teachers either agreed or strongly agreed on the improvements from learning to code. The specific break down of the collected data can be seen in Figure 3; the prompts corresponding to this figure are listed below:

1. Coding improves my students' persistence
2. Coding improves my students' problem solving
3. Coding improves my students' communication
4. Coding improves my students' creativity
5. Coding improves my students' collaboration skills
6. Coding is an essential skill for STEM and other future career opportunities for my students
7. BOTS added value to my classroom instruction
8. BOTS took an appropriate amount of time to implement
9. BOTS was easy to implement into my classroom curriculum

10. Participating in BOTS increased my confidence in teaching coding
11. I see a connection between computer science and robotics in my curriculum
12. I will continue to include coding skills in my future curriculum

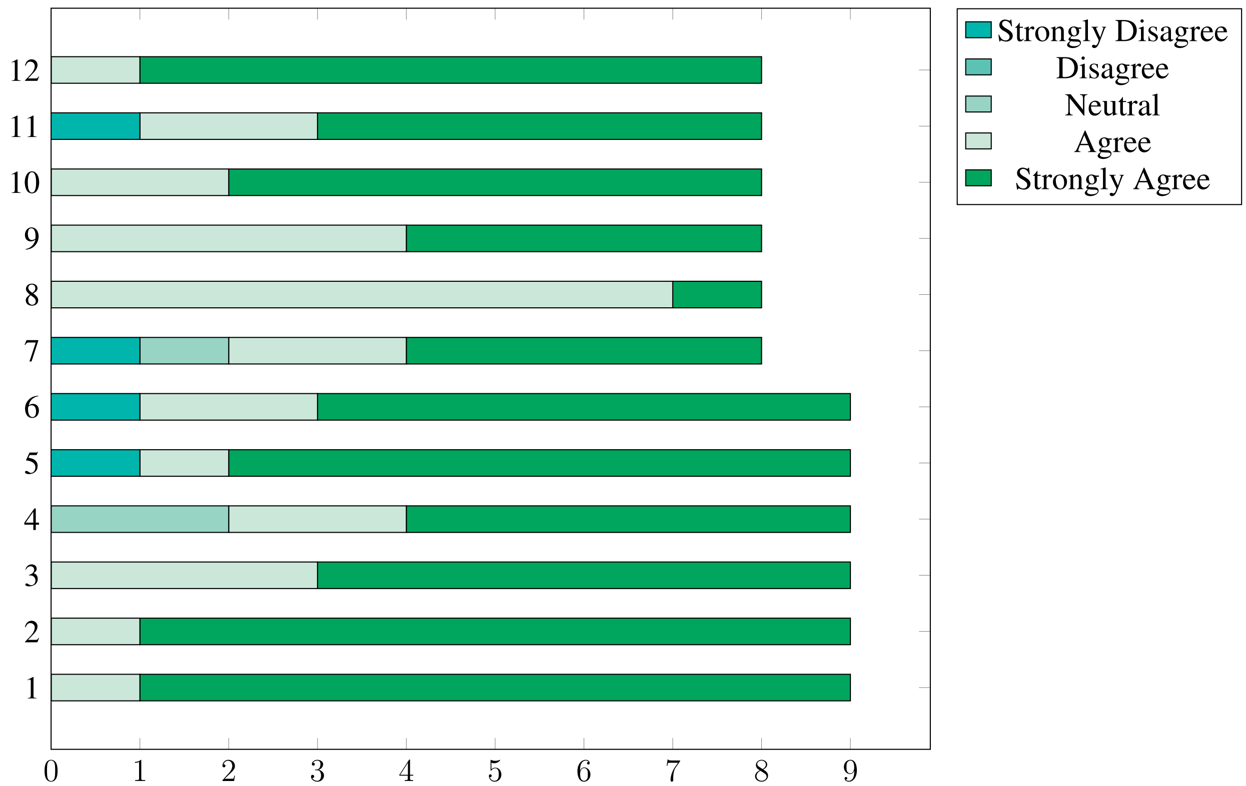

Figure 3: Responses to prompts from last survey of PD session 4.

Contradictorily, in another question related to how coding improved collaboration skills, one teacher strongly disagreed, while all other teachers either strongly agreed or agreed. The outlier continued into the next question, which asked teachers to rate their agreement with the question that proposed coding to be an essential skill for STEM and other future careers. All other teachers either agreed or strongly agreed that coding was an essential skill in STEM. Teachers were also asked to rate whether or not BOTS added value to their classroom instruction. Most agreed or strongly agreed, but a quarter of our attendees were ambivalent or in disagreement with the statement. Given that all teachers reported an interest in continuing with the BOTS program and with including coding in their curriculum, the disagreement may stem from external factors. One of the major goals of the program related to the accessibility of the curriculum, for teachers and students alike. On these points, teachers either agreed or strongly agreed that BOTS was both reasonably quick, and easy, to implement in their classrooms.

After running the BOTS program for half a year, it was found that the participants have an overall positive view of the BOTS program. Participants could see the importance of coding and programming in their classrooms and found that the program helped to increase their confidence in teaching this type of material to their students. The majority of teachers saw the value of having BOTS in their classroom and believed that it contributed to students' confidence, communication skills, persistence in learning, critical thinking skills, and collaboration abilities.

Discussion

The purpose of creating the Building Opportunities With Teachers in Schools program was to determine if a monthly community of practice between early elementary teachers, K-12 STEM Center staff and Ph.D. student volunteers could boost in-service, inner city teacher ability and self-confidence to teach coding and introduce robots as authentic, real-world digital learning opportunities. Based on the teacher responses to surveys administered pre- and post-PD sessions by an independent external evaluator, the results show this to be true.

Overall, the first- and second-grade teachers who participated in the BOTS program held a positive perception of these activities, as measured in the surveys from the four PD sessions. These surveys show that teachers felt more confident in implementing the material in the classroom and demonstrated improved self-efficacy navigating through Code.org and introducing related activities with the robots. As discussed here, the majority of teachers stated that BOTS activities led to increases in student collaboration, communication, and persistence.

Subsequently, the teachers completed the first academic year of PD by attending an additional four PD sessions during the following spring semester, and all but one teacher volunteered again to participate in the current academic year. One additional school within the Archdiocese was added to this second cohort of BOTS educators. Early results from this second year of PD confirm the benefits of this approach to enabling teachers to bring CS to early elementary students through long-term, low dose teacher training using low-cost, scalable materials that teachers could adapt to their in-class curriculum.

This program gives teachers and young students an exposure to robotics and computer science in a hands-on way that they would not experience otherwise until much later int their education, if at all. This early dip into the waters of applied computer science lessen the unfamiliarity that the students may experience when met with computer science again; this may be in the form of artificial intelligence, data science, or natural language processing, just to name a few. The international community can also benefit from this study. The courses and lesson plans from Code.org are presented in 60 different languages. Additionally, more time and emphasis can be given to unplugged activities if hardware access is limited.

During these past months, collaboration with STEAM professionals at the Los Angeles Unified School District and the Los Angeles Archdiocese has deepened. All partners are working toward using the BOTS approach to expand CS skill and confidence in select schools in each district. In the Fall of 2019, the BOTS team and LAUSD partnered to provide a week-long salary point class for 35 teachers; on Day 1, the Code.org PD in Computer Science Fundamentals was provided, Days 2-3 introduced teachers to the Sphero SPRK+® robots and physical mazes adapted from the Code.org lessons, and Days 3-4 showed teachers how their students could progress from block coding to building and programming VEX IQ robots as they moved into higher grades. Currently, the Center staff and volunteer Ph.D. students are preparing to expand these pilot studies in the coming academic year to ask the question: Can the teacher intervention and the community of practice provided by BOTS provide a sustainable solution to the problem of unequal access to age-appropriate 21st century computer science and computational thinking at elementary schools serving Latinx immigrant families in LAUSD and ADLA schools? Discussions are underway to bring the BOTS approach to new schools to determine the portability of this partnership approach between USC's K-12 STEM Center staff, Ph.D. student mentors and teachers at four inner city elementary schools along with the STEAM Coordinators at LAUSD and ADLA. Plans are to shift BOTS from a cohesive community of practice into an actual Research Practitioner Partnership, to obtain district and university permission to not only evaluate teacher progress in CS skills and confidence but also to measure their students' improvement in self-efficacy, persistence, and computational thinking.

Naturally, there are limitations to this study that need to be addressed moving forward with this work. Some possible future work and areas of improvement are as follows. First, it would be interesting to survey the students of these teachers who have participated in the BOTS program. Secondly, following these students as well as student whose teachers did not participate in BOTS throughout the remainder of their K-12 education could provide insight into how this program affects students in the long term. This could entail proficiencies in both STEM and non-STEM courses, as well as college enrollment or other forms of higher education. Lastly and most directly, more quantitative measures of educator performance and understanding could be implemented. For example, it could be useful to measure the time it takes a teacher to complete equivalent mazes with the robot both before and after an appropriate PD session. Given the need for low-cost interventions in inner city elementary schools, the positive results of this study show the promise of the BOTS program. Plans to address these limitations are underway as are steps to test the scalability of the program to new schools.

Acknowledgements

The authors thank the undergraduate student workers who aided in preparing for and facilitating the PD workshops. Special thanks is given to Hannah Hayes and Tyler LaBonte as well as Sandra LoRe, Ph.D. and Greg LoRe of SPEAR. Thanks is given to the USC Good Neighbors Campaign, the Specialty Family Foundation, and the K-12 STEM Center who provided support, financial and otherwise, for this project.

References

[1] Bernard Marr. 8 things every school must do to prepare for the 4th industrial revolution. Forbes Magazine, May 2019.

[2] Amanda Strawhacker and Marina U Bers. "i want my robot to look for food": Comparing kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces. International Journal of Technology and Design Education, 25(3):293–319, 2015.

[3] Marina Umaschi Bers, Louise Flannery, Elizabeth R Kazakoff, and Amanda Sullivan. Computational thinking

and tinkering: Exploration of an early childhood robotics curriculum. Computers & Education, 72:145–157, 2014.

[4] Seymour Papert. Mindstorms: Children, computers, and powerful ideas. Basic Books, Inc., 1980.

[5] Ismail Marulcu and Mike Barnett. Fifth graders' learning about simple machines through engineering design-based instruction using lego™ materials. Research in Science Education, 43(5):1825–1850, 2013.

[6] Mohammad Ehsanul Karim, Séverin Lemaignan, and Francesco Mondada. A review: Can robots reshape k-12 stem education? In 2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO), pages 1–8. IEEE, 2015.

[7] Amanda Sullivan and Marina Umaschi Bers. Robotics in the early childhood classroom: learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. International Journal of Technology and Design Education, 26(1):3–20, 2016.

[8] Elizabeth R Kazakoff, Amanda Sullivan, and Marina U Bers. The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. Early Childhood Education Journal, 41 (4):245–255, 2013.

[9] Emanuela Castro, Francesca Cecchi, Massimiliano Valente, Elisa Buselli, Pericle Salvini, and Paolo Dario. Can educational robotics introduce young children to robotics and how can we measure it? Journal of Computer Assisted Learning, 34(6):970–977, 2018.

[10] Gwen Nugent, Bradley Barker, Neal Grandgenett, and Viacheslav I Adamchuk. Impact of robotics and geospatial technology interventions on youth stem learning and attitudes. Journal of Research on Technology in Education, 42(4):391–408, 2010.

[11] Los Angeles Unified School District. Instructional technology initiative task force recommendations. Los Angeles Unified School District Division of Instruction, June 2016.

[12] Los Angeles Unified School District. Instructional technology initiative 2019-20 professional development catalog. Los Angeles Unified School District Division of Instruction, September 2019.

[13] Emily Hamner, Jennifer Cross, Lauren Zito, Debra Bernstein, and Karen Mutch-Jones. Training teachers to integrate engineering into non-technical middle school curriculum. In 2016 IEEE Frontiers in Education Conference (FIE), pages 1–9. IEEE, 2016.

[14] Thomas J Cortina and Keith Trahan. Increasing computing in high school through stem teacher workshops. International Society for Technology in Education (ISTE), 2013.

[15] Sarah Brasiel and Allen Ruby. Researcher-practitioner partnership in education research. Institute of Education Sciences National Center for Education Research, 2019.