Emerging Practices for Integrating Computer Science into Existing K-5 Subjects in the United States

Monica M. McGill
CSEdResearch.org
Peoria, IL, USA
monica@csedresearch.org

Laycee Thigpen
CSEdResearch.org
Lakeland, FL, USA
laycee@csedresearch.org

Alaina Mabie
CSEdResearch.org
Bradley University
Peoria, IL, USA
alaina@csedresearch.org

ABSTRACT

Research Problem. With increasing initiatives and policies to bring computer science (CS) learning opportunities to primary school students within the United States, there is a growing need to understand how to integrate CS into various other subjects. Research on emerging practices for integrating CS into other subjects (co-curricular), however, remains thin.

Research Questions. Our research question for this study was: *What are promising practices for integrating CS into other subject areas?*

Methodology. We conducted interviews with experts (n=9) in integrating CS into subject areas in the K-5 classroom using a semi-structured interview protocol.

Findings. Several promising practices emerged for designing curriculum, creating assessments, and preparing teachers to teach in a co-curricular manner. These include ways for teachers to vary instruction, integrating into core (and oft tested) language arts and mathematics, and simplifying assessments. Many of the findings are borne from the need to help new teachers become comfortable teaching a new subject integrated into their other subjects.

Implications. Our findings reveal a number of steps curriculum designers can take to design and implement CS integrated into other subject areas. While many are built on addressing the cognitive load that both teachers and students may face, there is deep recognition that teachers must gradually learn to become familiar with CS as well as how to integrate CS into other subjects that ensure true integration, rather than teaching two subjects in parallel.

CCS CONCEPTS

• **Social and professional topics → Computer education**; **Computer education programs**; **Computer science education**.

KEYWORDS

Computing education, learning, curriculum design, primary, K-5, elementary, integration, co-curricular, interdisciplinary, curriculum development, practices

WiPSCE '23, September 27–29, 2023, Cambridge, United Kingdom

ACM ISBN 979-8-4007-0851-0/23/09…$15.00
https://doi.org/10.1145/3605468.3609759

ACM Reference Format:
Monica M. McGill, Laycee Thigpen, and Alaina Mabie. 2023. Emerging Practices for Integrating Computer Science into Existing K-5 Subjects in the United States. In *The 18th WiPSCE Conference on Primary and Secondary Computing Education Research (WiPSCE '23), September 27–29, 2023, Cambridge, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3605468.3609759

1 INTRODUCTION

With increasing initiatives and policies to bring computer science (CS) learning opportunities to primary school students within the United States [9], there is a growing need to understand how to integrate CS into various other subjects. Further, there remain wide gaps in who has access to and is participating in CS education at the high school level [12]. Previous research highlights the fact that marginalized groups can feel that they don't belong in CS as early as 2nd and 3rd grades [7]. One solution to this is to bring CS education to lower grades so that belongingness (as well as knowledge) can be cultivated in the critical formative years. However, while middle school also has grown, the number of states that require CS in K-5 and the number of schools teaching CS in K-5 still remains low [9].

Some of the key barriers to offering CS to elementary school students that have previously been found include administrators are not supportive, teachers have not yet received training to teach CS, and resources for adoption remain low [8, 13, 20, 21]. A key barrier that has been mentioned in past studies is that teachers have no time to add an additional subject area to their day, particularly since they are immersed in teaching to their state standards which more heavily emphasize language arts and mathematics [8]. Another study found that other challenges specific to CT/CS in design include confusion of what an algorithm is, pupil resistance to design (as opposed to their eager desire to jump into coding), lack of time to teach design, lack of pupil experience with design, conflicting pedagogy choices and resources, and a lack of teacher knowledge about design [38].

All of these point to the larger need for any future curriculum to address a coalition of these factors. Integrating CS and computational thinking (CT) into subjects such as math and language arts has been viewed as a way to mitigate the barrier related to time [5]. It also recognizes that interdisciplinary education can benefit student learning and is often the core at K-5 learning [10], how integration occurs and how impactful it can be on student learning still remains unexplored.

In an effort to understand emerging promising practices in the field, we conducted a systematic literature review (results fully presented in [27]) and a qualitative study with experts in the field. Our research question for this study was: *What are promising practices for integrating CS into other subject areas?*

In this paper, we provide the results of our study as well as provide a set of recommendations to others considering emergent promising practices for creating co-curriculum for CS and other subject areas in the primary school classrooms. This study is important for curriculum designers who have or are interested in developing integrated CS curriculum.

2 BACKGROUND

In our systematic literature review described in [27], we conducted a search of articles focusing in on K-5 integrated CS across 2015-2022. Of the 900 articles in our initial search criteria, we eliminated all but 26 articles because they did not meet our criteria (e.g., not focused on K-5, not focused on integrated CS or computational thinking (CT), not focused on students as learners/participants). We limit our background to summarize this study, providing only a broad synthesis of our findings in this section.

2.1 Educator Perspectives

A few studies investigated educator perspectives on integrating CS into K-5 classrooms, including administrators and teachers. We touch briefly upon these in this section.

2.1.1 Administrator Perspectives. With respect to administrator perspectives, Howard conducted a qualitative study, interviewing five educational technology leaders about the challenges K-5 CS teachers face in the classroom and how integrating CS could be more effective and efficient [17]. Educational technology leaders were asked about their school and community as well as any personal successes or challenges when integrating CS. They were also asked what resources and materials their district used for lessons and projects while integrating CS and any CS efforts their state has made. The researchers identified three themes from the qualitative analysis: factors that influence the desire to teach CS in K-5, challenges in supporting K-5 teachers in CS integration, and ongoing professional needs of CS teachers. A shared challenge is related to uncertainty about assessments and basic technology deficiencies. The main factors that influence the desire to integrate CS in K-5 was student motivation and parent interest. State and community support are vital to increasing CS teachers interest.

2.1.2 Teacher Perspectives. Research examining K-5 teachers' perspective when integrating CS into the classroom indicates that a majority of teachers struggle defining and understanding CS and CT [11, 13, 35, 41]. Teachers also seemed to struggle separating CT from mathematics and CS from technology. This finding could be due to a lack of adequate and effective professional development, limited resources, and limited support staff [32]. Lastly, teachers have also struggled to justify the time spent on CS when it is not part of the standard state-mandated curriculum [32].

2.2 Subject Integration

To integrate CS into different subject areas, we highlight recent research into various areas of study.

2.2.1 Math Integration. It is no surprise that CS is often integrated into mathematics curricula due to the overlapping concepts such as abstraction, logic, variables, debugging, problem solving, and sequences. Niemelä et al. found that teachers preferred to integrate CS into geometry lessons over other mathematical subjects [28]. Several studies found that the integration of CS into mathematics helped keep students engaged [1, 31, 36].

Integrating CS into mathematics faces the usual challenges of teaching CS as a standalone subject, such as limited CS teacher training and limited lesson time. Results from previous research has found it challenging to integrate CS into mathematics without hindering the quality of the original mathematics curriculum [14, 22, 28]. One teacher believed that because mathematics has a reputation of being "hard", integrating CS into the curriculum would further deter students from participating in the class [28]. A few teachers believed that to be able to effectively and efficiently integrate CS into mathematics, the mathematics curriculum would need to be thoroughly updated [2]. This indicates that teachers may understand the challenge of meeting two sets of standards intertwined into more than one subject.

2.2.2 Science Integration. Within the literature, when CS is integrated into the K-5 science curriculum, it has often been integrated into biology. This could be due to the many cycles within biology such as cell cycles, life cycles, or reproductive cycles of plants. The cycles can be connected to CS concepts such as sequencing, debugging, abstraction, and iteration [25]. Many studies that integrated CS with biology have shown high student engagement and improvements in students' problem-solving, critical thinking, and CT skills [4, 25]. Luo et al. found that the participants were highly engaged (as measured by students' attention, excitement and time engaging with the tasks) in the lessons despite an initial low interest in CS.

CS integration into biology still suffers from limited lesson time and a lack in teacher confidence in their understanding of CS. Celepkolu et al. found that after teachers had attended a five day professional development workshop their knowledge of CS concepts improved, but they still lacked confidence in their understanding of CS [4]. Luo et al. were able to avoid these challenges by offering the CS course over the summer taught by a research team.

2.2.3 Engineering Integration. CS integration into K-5 engineering curriculum tended to be in the form of robotics and commonly included the use of educational tools such as the KIW robotics kit, the WeDo 2.0 kit, M-bot, or the Ozobot [6, 33, 37]. The use of educational tools has proven to keep students excited and engaged in the lessons [6, 33]. Chalmers found that students displayed high excitement, engagement, learning, coding and CT while using the WeDo 2.0 kits [6]. The students used trial and error while building the robot but once it was complete the students enjoyed altering the robot and exploring sequences.

Integrating CS into engineering and robotics also faces many of the same challenges seen throughout the literature such as limited lesson time, lack of teacher confidence in CS, and lack of resources [6, 21]. Khanlari found that teachers had insufficient supporting materials, such as the necessary hardware, software, and technical support which contributed to their lack of confidence in CS [20]. Another challenge is that teachers struggled justifying the time spent on integrating CS into engineering when it is not part of the state mandated curriculum [32].

2.2.4 Language Arts/Literacy Integration. CS integration into Language Arts is quite harmonious due to the overlapping concept of sequencing seen in storytelling and coding. Bers conducted a study using ScratchJr and taught 2nd graders how to use sequences, loops, repetition, and algorithmic thinking. Multiple studies had students follow directed tasks where they were given coding lessons that lead to an original narrative which required a plot and multiple characters [3, 39, 40]. The directed lessons were able to keep students engaged and excited about the lessons, and the students also showed signs of attentive listening [3, 40].

2.2.5 Integration into Multiple Subjects. CS integration is also commonly integrated into multiple subjects at once. Leonard et al. conducted a study that combined biology, dance, poetry, and CS into a multimodal lesson plan. This unique combination of subjects yielded high student engagement and excitement while also improving their CS and CT skills. Another similar study combined language arts, science, social studies, and CS into six modules [5]. One of the results from the studies highlighted challenges, including that the variety of integrated subjects can hinder the learning process and make it so the students are not able to improve student knowledge as may be typically expected [5, 23].

The literature shows that multi-subject integration of CS tended to yield high student engagement and improvement in student CS and CT skills [19, 23, 24, 30]. Educational tools and toys were also seen many times in the multi-subject integration literature such as KIBO, littleBits, and LEGO WeDo [19, 24] Lin conducted a study assess the how technology toys and tools can improve K-3 students CT skills [24]. It was found that the toys and tools were able to keep students engaged and improved students problem solving, reflection and collaboration skills.

Multi-subject integration of CS faces many of the same challenges as stand-alone CS, including limited lesson time, lack of teacher confidence and motivation, and lack of resources [5, 18, 30]. Israel et al. conducted a study at a school with students with no CS or CT programs and developed several integrated CS lessons, documenting challenges and barriers teachers faced along the way. They identified six main challenges which included access to technology, access to expert support in the classroom, computing access issues due to poverty and disability, limited instruction time, lack of students' computing experience, and classroom space. Century et al. found that students in a multi-subject integrated CS lesson did not have any significant differences academically or attitudinally compared to a control group.

3 METHODOLOGY

To answer our research question, *What are promising practices for integrating CS into other subject areas?*, we conducted interviews with experts in integrating CS into subject areas in the K-5 classroom (n=9) using a semi-structured interview protocol.

3.1 Participant Selection

We recruited participants that were researchers studying CS integration into various subject areas in primary schools. They were identified from studies they have published and shared. We also recruited participants in key districts in the United States that were notably engaged in the experimentation of this new type of curriculum in their classrooms, some of whom were engaged in Research Practice Partnerships (RPPs) with researchers. Participants were identified through their publications and presentations as well as other discussions our research team has had with these individuals over the last couple of years. A full list of the participants (with pseudonyms) is presented in Table 1.

3.2 Data Collection and Analysis

We conducted the interviews using a secure, private Zoom channel, recorded each with participant permission, and used a secure service for transcribing. The interviews lasted between 34 and 57 minutes (mean average of 43 minutes). We deidentified the interviewees prior to sending for transcription and secured the transcripts on secure, password-protected computers. Transcriptions were then uploaded on password protected, two-factor authenticated cloud software (Dedoose) for analyzing qualitative data.

We analyzed the data using deductive coding by first developing a set of codes *a priori* for each of the two core themes (general integration design and specific subject integration design). We relied on one researcher to code the entire set of interviews, with a second researcher providing quality checks against approximately 10% of the interviews to verify that the data was consistently and accurately coded across the set of interviews [29]. Relatively few discrepancies were discovered. As these were discovered, the two researchers discussed each discrepancy to determine how to resolve it and how the discrepancy may have impacted other codes. If a discrepancy resulted in the primary coder's interpretation of a code that may have impacted other codes (which occurred only twice), the primary coder went through all other text that was coded similarly to check for consistency. Once the interviews were all coded, we synthesized the data in the narrative and created new categories to group similar sets of data together that were not already grouped through the coding process. Categories were placed in one of the two respective themes (general or subject specific integration).

3.3 Data Saturation

Data saturation as a concept is widely used across several qualitative research methods [16]. Data saturation can occur in studies with as few as five participants for homogeneous study populations [15]. While there are various methods for measuring data saturation, we chose six initial participants for this study, and added two additional using a reflective pattern. If new codes and themes were being introduced after each new participant was added, we added another participant. To verify data saturation, we added one more participant, bringing the total to nine. No new themes or codes were added with the ninth participant, and we chose not to add any additional participants to our study.

3.4 Positionality Statements

One of the researchers in this study has been studying CS and CT in K-12 for several years, with a particular emphasis on reaching all students through equitable practices. They believe that integration is a promising practice for ensuring that all students will learn CS in meaningful and practical ways. They brought this perspective into

Table 1: Participant roles, experiences and knowledge about integrated CS.

Participant	Role and Experience	Specific Integrated CS Knowledge
Andrea	Associate Professor in School of Education; Research heavily focused on integration of CS into K-5 (particularly teacher PD)	Focused on integrating literacy and CS in K-5; also focusing on universal design for learning principles
Benjamin	Involved with K-12 CS framework development, K-12 CS standards, state standards; Former teacher and administrator	Focused on standards and content as well as teacher training; created assessments aligned with the curriculum at the grades 3 to 5 levels.
Elizabeth	STEM and CS Supervisor at very large county school district (11 years); former 5th grade teacher (6 years)	Focused on integrated, problem-based learning modules in elementary school classrooms
Mary	Associate Professor with extensive focused on Learning Sciences	Focused on integrating computing in math through all grades, K through 5; also focusing on early literacy, K through 2; extensive knowledge of learning theory as applied to computational thinking and CS
Patty	Mathematics education researcher studying how K-5 math relates to CS	Originally started with general interest of integrating technology into K-5 math curriculum, morphing into studying how to integrate Scratch programming into math in a mutually beneficial way. Worked several years on projects focused on integrating CT via unplugged activities for 3rd-5th grades.
Roxanne	30+ years experience conducting research and evaluation; projects focused heavily in supporting districts integrating CS into subject areas	Integrated CS into the literacy block for grades 3-5. Integrating CS and culturally responsive pedagogy with universal design for learning for 4th grade
Sallie	Associate Professor of CS; specializes in 3rd-8th grade CS education (15 years)	Integrating fractions and CS in 3rd and 4th grades; working on a 4th, 5th grade curriculum integrating CS into Literacy components and support for English language learners.
Terry	Assistant Professor in Education Department with focus on CS; PhD in Learning Sciences and holds a B.S. in CS	Began by adding standalone instruction in K-8 across district; Heavily involved in Math+CS integration, including alignment to standards
Wesley	Full professor and researcher focused on CT and coding in K-5; teaches instructional design and development.	Began focusing on teacher training with engineering and computing activities; external evaluator for a program that trains elementary teachers who teach coding.

this study and ensured that emphasis is also given to challenges and difficulties found to achieve success in teaching CS in all grades.

Another researcher is a former early childhood educator and has conducted qualitative research focusing on teacher education and experience. She is a professional development provider focused on creating equitable environments in educational settings. Her identity as a female, Black American contributed to this study by ensuring diverse voices of teachers were included.

The third researcher was involved with finding relevant articles associated with integrating CS into other subject areas. They are new to CS education research and have been working under the guidance of the other researchers.

4 RESULTANT CATEGORIES

Participants noted several areas that indicated emergent promising practices based on their experiences designing, implementing and studying practices in the K-5 classroom.

4.1 Emergent Promising Practices for General Integration Design

In this section, we focus on emergent promising practices for general integration, including curriculum, assessment and teacher readiness. Figure 2) provides a high-level summary.

4.1.1 Promising Practices for Curriculum Design. **Adopt Universal Design for Learning Principles.** Interviewees shared several characteristics of high quality integration. Andrea noted that they paid attention to "the layer of universal design for learning principles…making sure there's lots of different entry points and options for students throughout the whole lesson" as well as "lots of scaffolding." She developed the CS Plus Universal Design for Learning Planning Cycle which is "this whole way to think about designing instruction where you're integrating computer science with literacy and also universal design for learning principles to make it accessible for all students." Additionally she "designed professional development on how to integrate computational thinking and aspects of computer science into literacy instruction..[which]

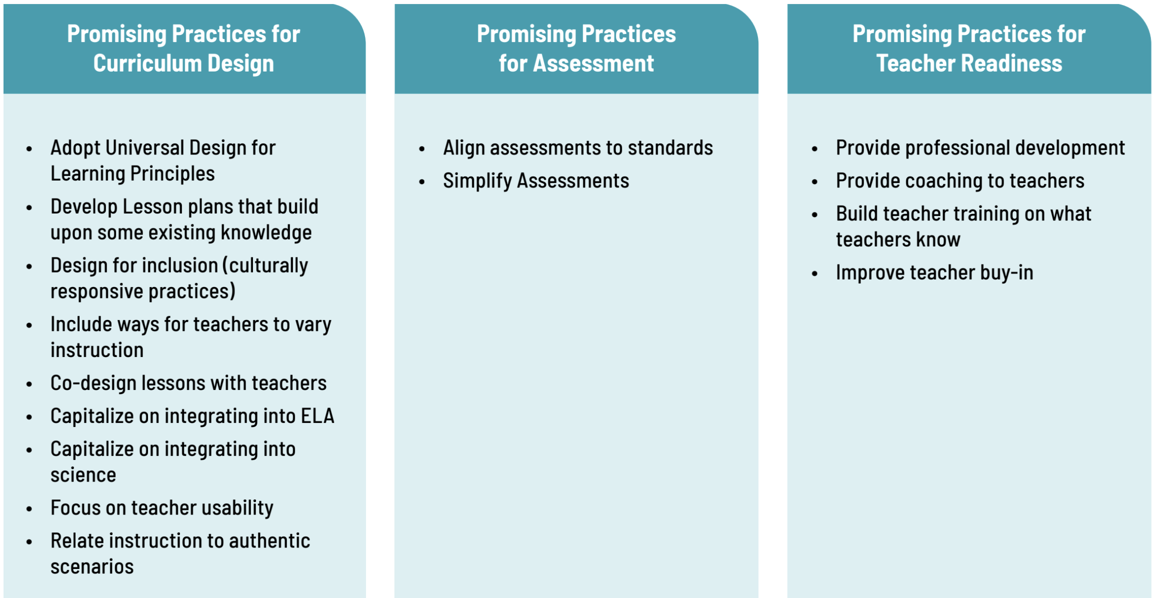

Figure 1: Summary of general findings. More details about how to deliver these practices are provided in Section 5.

focused on inclusion, so thinking about how you include students with disabilities in this." Others also noted the importance of fully integrated lessons–that subjects should not be taught without the careful entwining of both the subject matter plus CS.

Develop lesson plans that build upon some existing knowledge. Some participants noted high quality integration follows a CS trajectory and must include some content that, according to Sallie, "students are already familiar with," according to Andrea and "not introducing too much in a single lesson." This not only helped students, but also teachers. Through Patty's experience working with teachers she learned integration is easier when "they [teachers] start to see a connection anywhere." Patty also noted that teachers didn't want to wait for the best fit, but rather they were more of the mindset that 'Let's just pick one, engage with it' to try it out and see how it progresses.

Design for inclusion (culturally responsive practices). Andrea added that the teachers "teach very different populations of kids." She noted that her team "wanted to design something that could be done in a mainstream classroom that students without disabilities could engage in and was also accessible for students with disabilities." Roxanne noted that Culturally Responsive Pedagogy and Universal Design for Learning need to be considered in CS for students to "come out of the experience better than when they went in." Patty agreed that learning experiences should be culturally responsive: "the resources [should] be adaptable, so that kids can make some choices that make the project relevant to them."

Include ways for teachers to vary instruction. Roxanne noted, teachers need "some flexibility for teaching the lesson." An example of flexibility was Andrea and her team gave teachers "simple checklists at the end of each lesson" instead of rubrics, that she deemed "time-consuming," to assess their students' learning. She said, "The checklist, we found, [was] something that teachers felt like they could do more reasonably... We're still working on it, but that's what we've gotten." Sallie has experienced teachers who have tried the example codes and felt "none of those three are going to resonate with my class." These teachers have asked to make their own examples instead.

Co-design lessons with teachers. Mary discussed a CT unit that "introduced these computational thinking concepts and then asked [students] reflection questions of what does this remind you of in your discipline. . . to help connect it to those things. We got crickets because that's a big leap to make apparently, I now know. Instead, what we have switched to is these more activity based, so we co-design activities with the faculty member that the teachers can use in their actual classrooms." She also reflected that she works with a faculty member who is responsible for a grade or a discipline, and asks them to identify a problem that they currently have in their curriculum. She then works to find a solution for it by integrating CS.

Focus on teacher usability. One of Roxanne's projects was to integrate CS in the literacy block. Roxanne and her team were able to collaborate with "expert curriculum writers. . . to revise the modules to. . . up the quality of those modules as far as teacher usability." The goal of her most recent project is "to take one module, a fourth-grade module, and infuse culturally responsive pedagogy and universal design for learning" as an integral component.

Capitalize on integrating into Literacy. A key reason to integrate CS into literacy was noted by Benjamin, who stated that "there are far more minutes allocated to literacy in the day than any other subject. It feels like the easiest thing to align to, as opposed to say, we're taking over math instruction for the day with computer science. I'd say the driver for integration was mostly pragmatic in terms of getting into the school day for how it could fit in a scalable and equitable way." Andrea had different reasons for integrating CS into literacy, noting that "just having something that's already very familiar gives you a way to connect to something new, essentially in some ways it gives you a parallel. Another reason is, of course, coding is a literacy skill because it's a language. I feel like it belongs in the literacy category. . . A third reason is because I think anytime in the real world when you're going to program something, you are doing it for some purpose."

This turned out to be a great choice for Andrea. She further stated that, "In our first study, we heard directly from literally almost every teacher in our study that connecting it to literacy helped them feel comfortable teaching it, because it was so unfamiliar to them. It was something that for many of them was intimidating just because they had a lot to learn or they'd never taught it before. Not that they didn't feel capable, it's more that they're like it's just a lot of new information, I've never taught it before."

Capitalize on integrating into science. Roxanne and her team conducted a quasi-experimental study and "used a component-based approach to doing research" were they looked "at the whole intervention and its associations with outcomes...we did find that the groups who were using the [science] modules [taught] more CS." Additionally they "also found that certain components of the modules had association with particular outcomes, including attitudinal outcomes. For example, one of those components was interdisciplinary instruction. We view interdisciplinary instruction as one what we call an interactional component, meaning the behaviors and interactions we expect a teacher to have when engaging with students."

Relate instruction to authentic scenarios. Patty stated, "Actually when we were developing trajectories, what I thought were some of the most interesting conversations we had is...to think about, okay, so what are the different kinds of loops? You can have a counter stop the loop, or you can have a condition stop the loop. What do you need to think about if you're trying to make a computer do that? Is something repeating, yes or no? If it's stopping, why is it stopping? I was really interested in having that upfront, whereas I think the more CS-minded people were like, no, you do counted loops, and then you do conditional loops. I was like, I'm more interested in making sure kids think about how these relate to the real world."

4.1.2 Promising Practices for Assessment. **Align assessments to standards.** Not only are content and pedagogical approaches still being explored, assessment of learning among students is as well. According to Benjamin, "teachers are really unsure about assessment approaches." Benjamin shared that he worked with other CS experts "to create assessments aligned with the curriculum at the grades 3 to 5 levels. That was so helpful for clarifying what our CS outcomes were and having measures of what we mean by success."

Simplify assessments. While teachers continue to familiarize themselves with integrating CS into subject areas, simplified assessments can reduce the cognitive load teachers experience. Andrea described the success her team had with helping teachers assess their lessons. Originally, they had planned for teachers to use rubrics but found them to be an unreasonable expectation. They found that using a simple checklist at the end of each lesson to include what the teacher really wanted students to try was "something that teachers felt like they could do more reasonably."

4.1.3 Promising Practices for Teacher Readiness. **Provide professional development.** Multiple participants discussed the important role of teacher training for effective integrating practices. Andrea spoke to the level of difficulty many teachers have when creating integrated lessons, noting that "integration is very hard for teachers because it requires a lot of time and explicit planning." For teacher training to be most impactful, Benjamin suggested "thinking critically about how to implement it" instead of solely preparing them to implement them exactly as prescribed.

Sallie noted that PD was key to overcoming the intimidation that teachers feel about integrating CS into other subject areas, since many of the teachers "were very scared before they showed up." Overall Sallie's focus was to get teachers "to understand the curriculum [and]. . . to do it so that they're comfortable." During the PD, teachers go over the curriculum and view it from students' perspectives. Patty also provides interactive time with the curriculum.

Mary provides teacher training and models the activity they want the teachers to complete. She elaborated, "Whereas when we just threw them into the deep end, they felt very uncomfortable making their own code for themselves, but if you gave them code and asked them to copy it, they were much more comfortable with it. Not copy-paste but actually have to redo it themselves." Mary also had teachers use the activity in student teaching.

Provide coaching to teachers. Terry mentioned that "the technology coach as a mediator" proved to be beneficial in supporting teachers and being available for troubleshooting. Wesley stated in an ideal world every school would have a "computer specialist." He added, "What I've seen work the best in the different schools. . . is when there is a really good relationship between the specialist teacher who teaches the coding skills to the kids and the classroom teachers who works with the coding teacher to then have projects coordinated with that teacher." Andrea and her team provided scaffolding of support for the teachers because they were "all at very different places in their journey." Some teachers needed and heavily relied on instructional videos, other teachers only used the lesson plans, and some teachers did not need either support and were able to make "adjustments based on what's happening" during their instructional time. Mary noted, she would "co-teach [with a faculty member]. . . and introduce the programming activity ... Then the faculty member will "layer on that disciplinary perspective, especially related to the teaching methods that they're learning."

Roxanne supported teachers by having "respect for the teacher's experience... [making] sure that the teachers know that they're educational research partners [and] they're not just somebody who's been forced to teach this lesson."

Build teacher training on what teachers know. Mary shared her experience, "When we started, we just had the topic of patterns, and so I created a program that used a loop. I created something, and I showed them how I created that program so they saw an existing program and how to do it, and then I sent that code to them. . . and they could substitute different pieces. . . Then it was a dead end because I didn't know how they would create a lesson out of it for their students. This year, instead, what we did is the faculty members sent me a pattern that they see in the standards, and I recreated that pattern in Pencil Code so that it matched exactly what they were looking at and so that it was connected to the standards, their lesson plan, all that stuff. To me, I don't really care what the pattern is, so it was easy to create that pattern. Then it's easier for them to see how they would apply it to the classroom."

Improve teacher buy-in. Patty noted having certain teachers that were motivated to try CS for various reasons such as, '"I like it because it's technology," and there were some that were, "I like it because it's going to support my math scores, my math teaching, my math learning." There weren't that many that were just like, "I like it because I think CS is important."' Mary shared that teachers were "way more motivated when it's something that they already know about and care about." Wesley echoed that teachers were motivated to try coding when they "realized that actually coding is really useful" and that it could be used to improve their teaching.

4.2 Emergent Promising Practices for Specific Subject Integration Design

In this section, we provide a summary of the findings for emergent promising practices for integrating CS into specific subject areas, focused on general subjects, literacy, math, and science. Figure 2) provides a high-level summary for this section.

4.2.1 Integration into Literacy. **Use games and other tools.** Participants took various approaches for integrating CS into literacy. Elizabeth shared that "[Students] made a game for ELA on matching vocabulary words, using a Micro:bit and had the Micro:bit tell them if they're right or wrong in scratch." Mary added, "In the literacy one, I don't know if we're going to get there just because I don't know if it's appropriate unless we end up doing a digital storytelling, which is what most people do."

Reflect upon how language is used in English and in computing. Mary is also working on a project that focuses on early literacy (K-2). Her team is having conversations "about language and how you understand the different language associated with computing and how that's important for general literacy."

4.2.2 Integration into Science. **Leverage cause and effect in science with conditional logic.** Roxanne shared, "In the CS, we thought that the ideas of conditional logic went well with the science content of invasive species, the notions of cause and effect…" She added, "there's lots of different invasive species in the area. That was already part of the curriculum. Integrating it with CS has made it more accessible to some students who didn't understand the impact prior, which has been a really big selling point with our curriculum team here at the district."

4.2.3 Integration into Math. **Go heavy on computational thinking.** Patty "thinks there's a lot of interesting ways that integrating computer science could benefit the mathematical learning of students." Additionally she "found that the teachers found the CT practices–debugging, decomposition, abstraction–to really open up their ways of thinking about how to engage kids in problem-solving. It gave kids the vocabulary to talk about their problem-solving a lot more. I think that gave a little more traction on making elementary math especially more problem-based instead of just engaging in exercises that repeat the same things over and over again."

Use virtual manipulatives. Sallie discussed her team's success with developing a CS integrated with math curriculum whose goal "was to provide a simple enough CS experience that provided opportunities to play with the math and engage with the math." The activities ranged from "projects that were basically virtual manipulatives,..projects where they were illustrating a number sentence,... project where it was displaying the same thing in different modes like a number line plus a pie chart plus a line." The various projects introduced only a small amount of coding each time.

Focus on fractions. Sallie noted that in another project, "We chose fractions because it's something students struggle with and there's correlations between understanding fractions and being able to do algebra.... Fractions are a very important subject in 3rd and 4th grade" and are often taught "at the end of the year and then some teachers don't get to it. We were doing it within everyday mathematics, which does a spiral curriculum. They start near the beginning of the year with early fractions readiness."

Enhance learning with other tools. Terry noted he has used Sphero for a math project where students could "represent mathematical ideas programmatically in a block space language…represent them through Sphero in deciding direction use as a digital protractor as a way to define direction." He added, "Students did a lot of embodiment of the robot, and embodiment of mathematical ideas in terms of deciding angles and direction, and durations, and speeds. My students also looked at an embodied perspective around ways that mathematical and computational thinking ideas are expressed through gesture and motion mediated by the Sphero itself." Terry also shared his experience with "curricular mechanism called Cornerstones" which is a "long-term problem-based learning opportunities that exist at every grade and are aligned to different subjects." "We crafted a research study essentially asking a question of what it looks like to integrate computational thinking and robotics in a fourth grade math classroom. With particular characteristics of working in Title 1 schools where historically they have not had a lot of formal computer science present. The project was about designing these cornerstones. Then we also explored ways to align them to math standards, ways to align them to the Eureka Math curriculum, which is what we use in elementary school."

4.2.4 Integration into Social Studies. **Incorporate cultural holidays into CS.** One participant noted that she teaches standalone CS, but her comment reflected that she integrated social studies into CS. Sallie said, "When it's standalone, I do all culturally relevant stuff. The theme is always like, oh, cultural holiday or some cool youth culture thing. You have to pick a theme for the CS."

5 DISCUSSION

Participants noted several emergent promising practices based on their experiences designing, implementing and studying practices in the K-5 classroom. We further summarize the findings in this section, framing the findings as recommendations. We also refer back to previous research as appropriate. In this section, we focus on emergent promising practices for general integration design, including curriculum, assessment and teacher readiness.

5.1 Curriculum Design

With respect to how to integrate CS into other subjects, previous research has shown that barriers exist to integrating CS into other subject areas, including curricular co-design and general elementary school CS education processes [8]. These include limited instructional time and budget for resources [8].

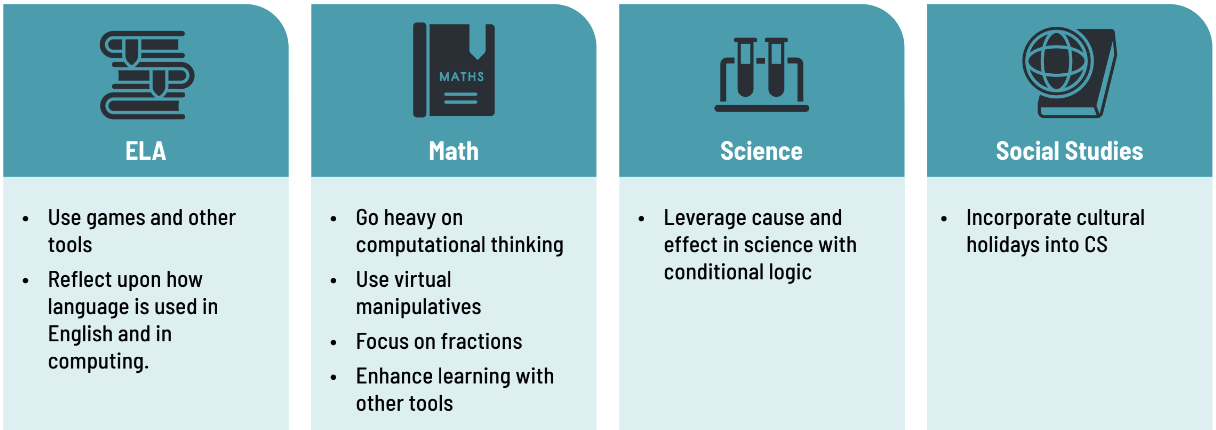

Figure 2: Summary of subject integration findings. More details for each are provided in Section 5.

5.1.1 Capitalize on integrating into literacy. Language arts was mentioned several times by participants, with emphasis on using integrated CS as a self-expression. Draw also upon the similarities between languages like English and programming languages, including, as one person noted, "language and how you understand the different languages associated with computing and how that's important for general literacy." This recommendation is based on the evidence provided by several participants that literacy is the core subject in K-5 and many schools, including those that are Title I, are spending the bulk of their time teaching literacy in K-5.

5.1.2 Adopt Universal Design for Learning Principles. The Universal Design for Learning is a framework for high-quality, inclusive instruction. Participants noted the role it plays in constructing integrated CS in ways that meet the needs of learners. This recommendation is based on Universal Design for Learning being a key driver for creating equitable outcomes in curriculum design and achieving accessibility for all students.

5.1.3 Design for inclusion (culturally responsive practices). Broaden diverse perspectives throughout the curriculum and ensure that all students are considered and included in the design. This recommendation highlights the need for designing and including culturally relevant practices in the curriculum.

5.1.4 Include ways for teachers to vary instruction. Honor different approaches to integration, particularly paying attention to the fact that many teachers are new to CS. Offering teachers the ability to choose a variegated pathway that they feel comfortable with and that meets the needs of their students can give teachers more agency and success. This recommendation highlights the need for teachers to feel comfortable with the curriculum and adapt it in ways that are meaningful for their students.

5.1.5 Develop lesson plans that build upon some existing knowledge. This speaks to the cognitive load that students experience when learning two new subjects and the learning tools that often accompany them. Cognitive load for not only a new subject but also a subject integrated into another is high for both teachers and students, which is known to be a barrier to learning. Breaking this down by building upon existing knowledge as well as careful scaffolding can help mitigate this issue. Careful scaffolding will enable more learning by reducing that cognitive load.

5.1.6 Co-design lessons with teachers. Including teachers' voices in the design of lessons for integrating CS into other subject areas will enable greater success. This will also enable greater usability of the lessons by teachers. This recommendation showcases the need for teachers to be a major and consistent part of the design process to ensure that the curriculum fits the needs of the teachers and the students.

5.1.7 Relate instruction to authentic scenarios. Add connections to integrated activities that are authentic in relation to students' worlds. With this, provide complementary videos that make these connections similar to those that Code.org has created in the past. This recommendation stresses the need for the CS activities to be authentic and relevant to the students' worlds.

5.2 Assessment

5.2.1 Align assessments to standards. Ensure that the assessments are aligned to the standards for the subject being taught and for CS. This recommendation highlights the need for assessments to be aligned to standards.

5.2.2 Simplify Assessments. While teachers are still working to understand how to teach and assess integrated CS, keep the assessments simple (e.g., a simple checklist of whether the items have been met, potentially with yes/no options). This recommendation emphasizes the need for assessments to be simple, to be teacher friendly, and to assess the integrated subjects.

5.3 Teacher Readiness

Preparing teachers to teach CS and CT is a critical and well-known issue as CS and CT enter K-12 classrooms. The need is no different for integrating CS/CT into specific subject areas [8]. As teachers taught integrated CS with subject areas, they were not confident in their understanding of CS [4, 5, 18, 25, 30]. Previous research indicated that teachers' confidence in teaching CS integrated with mathematics had significantly increased through training [1, 22, 41].

5.3.1 Provide professional development. Within PD provided on integrating CS into other subject areas, it was noted that building teacher training on what teachers know in a scaffolded approach will reduce the cognitive load and will increase the levels of comfort teachers feel with the materials. This recommendation stresses the critical role that professional development provides to teachers learning how to teach CS integrated lessons.

5.3.2 Provide coaching and resources to teachers. In addition to general PD to teachers, providing coaching (and resources) as they start to train and implement their new practices for integrating CS is critical to their success. This recommendation highlights the needed support of coaching and resources to help teachers remove barriers and grow more knowledgeable and skillful in understanding and teaching CS integrated subjects.

5.3.3 Improve teacher buy-in. Teacher buy-in is critical to the success of integrating CS into elementary schools. Without teacher buy-in, advancing this integration in a way that leads to academic achievement is critical. This recommendation is based on the need for teacher buy-in in order for successful CS implementation in elementary school.

5.4 Subject Matter Specific

We were surprised to learn that participants didn't really discuss engineering or robotics in their interviews, particularly given the number of studies that exist in the literature. Although engineering is not included in this list, we caution the reader to consider integrating into engineering as well as other subjects (i.e., fine arts, physical education, social studies). Additional research into these subject areas is warranted and expected over the next few years. In this section, we limit our discussion to literacy, mathematics and science, which were the focus subjects of our participants.

5.4.1 Literacy. Integration of CS into literacy has been documented with promising results [3, 39, 40]. Two previous studies used storytelling integrated with CS concepts that enabled students to program their stories using appropriate age-based tools [39, 40]. Students showed high engagement as measured by task completion, excitement with the activities and listened attentively to the instructions [3, 34, 40]. Within our study, the following emergent promising practices emerged for literacy:

- **Lean heavily into components of computational thinking.** CT can be integrated into the literacy curriculum. For example, sequencing, a component of CT, is used within storytelling and can be taught within that context.
- **Reflect upon how language is used in English and in computing.** There are commonalities between language syntax and semantics for both English and in computing. Reinforcing these commonalities could provide another connection for teachers and students to build upon.
- **Support narrative storytelling.** Storytelling involves self-expression and agency, which has been shown to be an effective way to engage students in their learning.
- **Use games and other tools.** Even in literacy, games or game-like tools and frameworks can be a powerful tool for learning. One participant specifically noted this as a tool they use in integrating CS in literacy.

5.4.2 Mathematics. Previous research has shown that integrating CS into mathematics has led to positive outcomes, including an increase in student engagement [36] and student motivation to learn mathematics [1]. With respect to content knowledge, increases were found in student knowledge [36], students' sequential thinking [26], and students' ability to decompose number problems [26]. Within our study, we found the following emergent promising practices:

- **Go heavy on computational thinking.** We recommend heavily integrating CT into mathematics. One participant noted that teachers found the CT practices of debugging, decomposition, and abstraction opens ways of thinking about how to engage students in problem-solving, including giving kids vocabulary for problem-solving.
- **Use virtual manipulatives.** Virtual manipulatives, particularly those that students may be familiar with in the real world and those that provide authentic context. An example might be a project where it is displaying the same data in different modes like a number line, a pie chart, and a line.
- **Focus on fractions.** Participants have had positive experiences integrating CS into the instruction of fractions, which is a key mathematical concept often taught in 3rd grade.

5.4.3 Science. When CS is integrated with science, research has shown that student engagement is high and that it increases students' problem-solving skills, critical thinking, CT, and content knowledge of both subjects [4, 25]. Participants shared the following emergent promising practices for integrating CS with science:

- **Leverage cause and effect in science with conditional logic.** One participant noted that cause and effect processes in science lend themselves to conditions and flow in CS.

5.5 Limitations

There are several limitations to this study. When compared to other subject areas, integrating CS into other subject areas is relatively new. It is difficult to identify emerging promising practices quantitatively at this point and qualitative research affords the opportunity to dig deeper into the particular questions related to how and why which practices are being used. Given this, the findings should be taken into context of the newness of the field.

Another limitation is that the study took place with researchers and curriculum developers within the United States. Extending this to additional participants outside of the U.S. could contribute to a more comprehensive set of emerging promising practices.

6 CONCLUSION

With increasing initiatives and policies to bring computer science (CS) learning opportunities to primary school students within the United States [9], there is a growing need to understand how to overcome barriers to bringing CS into the K-5 school classroom. The evidence from our study indicates that there are several emergent promising practices for integrating CS and CT with other subject areas. This evidence is general in nature and is also specific to certain subject areas. Given this nascent field, we expect the list of promising practices to both solidify and to grow.

ACKNOWLEDGMENTS

This material is based upon work supported by Code.org. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Code.org.

We acknowledge and thank Brenda Huerta for her assistance with the literature review.

REFERENCES

- [1] Gashawa Ahmed, Jalal Nouri, LeChen Zhang, and Eva Norén. 2020. Didactic methods of integrating programming in mathematics in primary school: Findings from a swedish national project. In *Proceedings of the 51st ACM technical symposium on computer science education*. 261–267.
- [2] Květoslav Bártek. 2017. CROSS-CURRICULAR LINKS BETWEEN INFORMATICS AND MATHEMATICS IN CONTEMPORARY CZECH SCHOOL REALITY. In *ICERI2017 Proceedings*. IATED, 4286–4289.
- [3] Marina Umaschi Bers. 2019. Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education* 6, 4 (2019), 499–528.
- [4] Mehmet Celepkolu, Erin O'Halloran, and Kristy Elizabeth Boyer. 2020. Upper elementary and middle grade teachers' perceptions, concerns, and goals for integrating CS into classrooms. In *Proceedings of the 51st ACM technical symposium on computer science education*. 965–970.
- [5] Jeanne Century, Kaitlyn A Ferris, and Huifang Zuo. 2020. Finding time for computer science in the elementary school day: a quasi-experimental study of a transdisciplinary problem-based learning approach. *International Journal of STEM Education* 7, 1 (2020), 1–16.
- [6] Christina Chalmers. 2018. Robotics and computational thinking in primary school. *International Journal of Child-Computer Interaction* 17 (2018), 93–100.
- [7] Sapna Cheryan, Allison Master, and Andrew N Meltzoff. 2015. Cultural stereotypes as gatekeepers: Increasing girls' interest in computer science and engineering by diversifying stereotypes. *Frontiers in psychology* (2015), 49.
- [8] Victor R Lee Jody Clarke-Midura and Jessica Shumway Mimi Recker. 2022. "Design for Co-Design" in a Computer Science Curriculum Research-Practice Partnership. In *Proceedings of the 16th International Conference of the Learning Sciences-ICLS 2022*.
- [9] Code.org, CSTA, & ECEP Alliance. 2022. 2022 State of Computer Science Education.
- [10] Selma Deneme and Selen Ada. 2012. On applying the interdisciplinary approach in primary schools. *Procedia-Social and Behavioral Sciences* 46 (2012), 885–889.
- [11] Georgios Fessakis and Stavroula Prantsoudi. 2019. Computer Science Teachers' Perceptions, Beliefs and Attitudes on Computational Thinking in Greece. *Informatics in Education* 18, 2 (2019), 227–258.
- [12] Carol L Fletcher, Bryan Cox, Leigh Ann DeLyser, and Amy J Ko. 2021. Applying CAPE to Assess Equitable CS Education. In *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE, 1–2.
- [13] Megean Garvin, Heather Killen, Jan Plane, and David Weintrop. 2019. Primary school teachers' conceptions of computational thinking. In *Proceedings of the 50th ACM technical symposium on computer science education*. 899–905.
- [14] Mariam Haspekian. 2017. Computer science in mathematics new curricula at primary school: New tools, new teaching practices. In *Proceedings of the 13th International Conference on Technology in Mathematics Teaching (ICTMT 13)*. 23–31.
- [15] Monique Hennink and Bonnie N. Kaiser. 2022. Sample sizes for saturation in qualitative research: A systematic review of empirical tests. *Social Science & Medicine* 292 (2022), 114523. https://doi.org/10.1016/j.socscimed.2021.114523
- [16] Monique M Hennink, Bonnie N Kaiser, and Vincent C Marconi. 2017. Code saturation versus meaning saturation: how many interviews are enough? *Qualitative health research* 27, 4 (2017), 591–608.
- [17] Nicol R Howard. 2019. EdTech Leaders' Beliefs: How are K-5 teachers supported with the integration of computer science in K-5 classrooms? *Technology, Knowledge and Learning* 24, 2 (2019), 203–217.
- [18] Maya Israel, Jamie N Pearson, Tanya Tapia, Quentin M Wherfel, and George Reese. 2015. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education* 82 (2015), 263–279.
- [19] Elena Jurado, David Fonseca, Jorge Coderch, and Xavi Canaleta. 2020. Social steam learning at an early age with robotic platforms: A case study in four schools in Spain. *Sensors* 20, 13 (2020), 3698.
- [20] Ahmad Khanlari. 2016. Teachers' perceptions of the benefits and the challenges of integrating educational robots into primary/elementary curricula. *European Journal of Engineering Education* 41, 3 (2016), 320–330.
- [21] Ahmad Khanlari and Fatemeh Mansourkiaie. 2015. Using robotics for STEM education in primary/elementary schools: Teachers' perceptions. In *2015 10th International Conference on Computer Science & Education (ICCSE)*. IEEE, 3–7.
- [22] Theodore J Kopcha, Cheryl Y Wilson, and Dayae Yang. 2022. Improving teacher use of educational robotics to teach computer science in K-5 mathematics. In *Computational Thinking in PreK-5: Empirical Evidence for Integration and Future Directions*. 47–54.
- [23] Alison E Leonard, Nikeetha Dsouza, Sabarish V Babu, Shaundra B Daily, Sophie Jörg, Cynthia Waddell, Dhaval Parmar, Kara Gundersen, Jordan Gestring, and Kevin Boggs. 2015. Embodying and programming a constellation of multimodal literacy practices: Computational thinking, creative movement, biology, & virtual environment interactions. *Journal of Language and Literacy Education* 11, 2 (2015), 64–93.
- [24] Veronica J Lin. 2015. Computational thinking and technology toys. (2015).
- [25] Feiya Luo, Pavlo D Antonenko, and E Christine Davis. 2020. Exploring the evolution of two girls' conceptions and practices in computational thinking in science. *Computers & Education* 146 (2020), 103759.
- [26] Feiya Luo, Maya Israel, and Brian Gane. 2022. Elementary computational thinking instruction and assessment: A learning trajectory perspective. *ACM Transactions on Computing Education (TOCE)* 22, 2 (2022), 1–26.
- [27] Alaina Mabie, Monica M. McGill, and Brenda Huerta. 2023. A Systematic Literature Review Examining the Impacts of Integrating Computer Science in K-5 Settings. In *In Proceedings of the American Society of Engineering Education*.
- [28] Pia Niemelä, Tiina Partanen, Maarit Harsu, Leo Leppänen, and Petri Ihantola. 2017. Computational thinking as an emergent learning trajectory of mathematics. In *Proceedings of the 17th Koli Calling international conference on computing education research*. 70–79.
- [29] Patrick O'Connor. 2002. Determining the impact of health library services on patient care: a review of the literature. *Health Information & Libraries Journal* 19, 1 (2002), 1–13.
- [30] Stephanie DC Playton, Rebecca Hite, and Paula K Leach. 2021. A Case Study of a Researcher-Practitioner Partnership in Teaching STEM+ C to Rural Elementary Students. *The Electronic Journal for Research in Science & Mathematics Education* 25, 2 (2021), 30–46.
- [31] Kathryn M Rich, Diana Franklin, Carla Strickland, Andy Isaacs, and Donna Eatinger. 2022. A learning trajectory for variables based in computational thinking literature: Using levels of thinking to develop instruction. *Computer Science Education* 32, 2 (2022), 213–234.
- [32] Peter Rich, Olga Belikov, Emily Yoshikawa, and McKay Perkins. 2018. Enablers and inhibitors to integrating computing and engineering lessons in elementary education. *Journal of Technology and Teacher Education* 26, 3 (2018), 437–469.
- [33] José Manuel Sáez López, Rogelio Buceta Otero, and Sebastián De Lara García-Cervigón. 2021. Introducing robotics and block programming in elementary education. *RIED. Revista Iberoamericana de Educación a Distancia* 24, 1 (2021), 95–113.
- [34] Ammar H Safar, Ali A Al-Jafar, and Zainab H Al-Yousefi. 2016. The effectiveness of using augmented reality apps in teaching the English alphabet to kindergarten children: A case study in the State of Kuwait. *EURASIA Journal of Mathematics, Science and Technology Education* 13, 2 (2016), 417–440.
- [35] Phil Sands, Aman Yadav, and Jon Good. 2018. Computational thinking in K-12: In-service teacher perceptions of computational thinking. *Computational thinking in the STEM disciplines: Foundations and research highlights* (2018), 151–164.
- [36] Carla Strickland, Kathryn M Rich, Donna Eatinger, Todd Lash, Andy Isaacs, Maya Israel, and Diana Franklin. 2021. Action Fractions: The design and pilot of an integrated math+ CS elementary curriculum based on learning trajectories. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 1149–1155.
- [37] Amanda Sullivan and Marina Umaschi Bers. 2016. 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 (2016), 3–20.
- [38] Jane Lisa Waite, Paul Curzon, William Marsh, Sue Sentance, and Alex Hadwen-Bennett. 2018. Abstraction in action: K-5 teachers' uses of levels of abstraction, particularly the design level, in teaching programming. *International Journal of Computer Science Education in Schools* 2, 1 (2018), 14–40.
- [39] Robert Whyte, Shaaron Ainsworth, and Jane Medwell. 2019. Designing for Integrated K-5 Computing and Literacy through Story-making Activities. In *Proceedings of the 2019 ACM Conference on International Computing Education Research*. 167–175.
- [40] Robert Whyte, Shaaron Ainsworth, and Jane Medwell. 2020. Designing Multimodal Composition Activities for Integrated K-5 Programming and Storytelling. (2020).
- [41] Aman Yadav, Christina Krist, Jon Good, and Elisa Nadire Caeli. 2018. Computational thinking in elementary classrooms: Measuring teacher understanding of computational ideas for teaching science. *Computer Science Education* 28, 4 (2018), 371–400.