STEM Education Review

REVIEW ARTICLE

Integrating computational thinking into K-12 education: Bridging the gap between theories and practices

Zhichun Liu*

Faculty of Education, The University of Hong Kong, Hong Kong SAR, China

ABSTRACT

Computational thinking (CT) is now widely recognized as an important literacy skill for students in today's digital world. However, many existing initiatives are framed to present a narrow focus on the knowledge and skills related to how to use computing tools. In this article, I describe the gap between theoretical discussions on CT as a literacy for learning and practical implementations of various CT or CT-related curricula. Then, I discuss possible challenges in addressing the gap. Finally, the article will explore a conceptual framework for integrating CT learning with Science, Technology, Engineering, and Mathematics (STEM) education in a way that is accessible to all students. By highlighting these challenges and proposing solutions, this article aims to contribute towards building a more comprehensive and effective approach to bringing CT into K-12 education.

Key words: computational thinking, K-12, STEM education, teacher professional development

INTRODUCTION

In today's digital world, computational thinking (CT) has become an important literacy skill for students.[1] Although definitions of CT vary,[2,3] it is generally recognized as the thinking skills involved in problem-solving, which enables individuals to use technology to solve complex problems.[4] As technology continues to advance and become more integrated into our daily lives, CT skills may be important for students' future success in various professional settings.[5] As a result, many countries are promoting CT initiatives to bring CT into K-12 classrooms (*e.g.*, the Computer Science for All [CS4All], CoolThink@JC program, Scratch Education Collaborative). However, most current initiatives aimed at developing CT skills in students focus primarily on the adoption of computing tools such as unplugged devices, blocked-based programming, or educational robotics.[6] This narrow focus might neglect the broader theoretical discussions on CT as a literacy for learning.[1] Particularly, although teachers' enthusiasm for participating in CT professional learning is often high, there is a significant gap between the theoretical discussions and practical implementations of various CT or CT-related curricula.[7] For example, previous research has highlighted the lack of teacher training and support.[8,9] In addition, although there are CT-integrated Science, Technology, Engineering, and Mathematics (STEM) learning opportunities available, practitioners often have a limited understanding of CT, perceiving it within the context of coding rather than in a more comprehensive and integrated manner.[10]

This article aims to explore the gap and the challenges in bringing CT into K-12 STEM classrooms. By identifying the challenges, the article aims to propose a conceptual

***Corresponding Author:**
Zhichun Liu, Faculty of Education, The University of Hong Kong, Pok Fu Lam Road, Central and Western District, Hong Kong SAR, China. Email: lukasliu@hku.hk; https://orcid.org/0000-0002-6402-9174
Received: 22 September 2023; Revised: 19 December 2023; Accepted: 24 December 2023; Published: 29 December 2023
https://doi.org/10.54844/stemer.2023.0467

framework for engaging teachers in integrating CT learning with STEM education in a way that is accessible to all students. The proposed framework is intended to bridge the gap between theoretical discussions and practical implementations and provide a more comprehensive and effective approach to bringing CT into K-12 education.

CT IN K-12

In the past few decades, there has been an increasing interest in developing CT skills in younger learners.[5,11] Research has argued that CT, as a set of problem-solving skills, can help students develop many important skills in the 21st century, such as critical thinking, problem-solving, and creativity.[1] For example, Grover and Pea emphasized the importance of CT as a critical analytical skill that every child should possess.[12] Additionally, researchers have suggested that CT-related skills should be introduced to elementary and secondary school students early on to facilitate cognitive development.[11] Past research further supported the notion that developing CT in students can lead to interest in STEM subjects and careers.[13]

Despite the importance of CT education, there are still significant challenges associated with incorporating it into the K-12 curriculum. One of the most significant challenges is the lack of teacher training. Many teachers have not been trained in CT skills and may not feel comfortable teaching it. Integrating CT into the broader curriculum of STEM-related fields is challenging for many K-12 educators.[14] The disconnection between CT skills and K-12 STEM subjects may result in students being unable to transfer CT skills to real-world situations, particularly in solving authentic STEM challenges encountered in professional environments.[8] With the growing importance of STEM education worldwide, the integration between CT and STEM learning proposes important challenges to how research should move forward.

INTEGRATED VIEWS OF CT

Although the definitions and views of CT vary quite a bit,[2] different ways of defining CT emphasize its problem-solving nature.[4,12,15] As such, CT, naturally, is seen as a thinking tool that could be applied to diverse contexts and disciplines. Particularly, STEM areas traditionally integrate with computing tools to solve disciplinary or interdisciplinary problems (*e.g.*, computational physics or simulations in engineering). As a result, an intuitive goal of integrated CT learning is to enable K-12 students to use computing tools for specific scientific purposes. This integration approach centers around tool adoption and application. For example, students could learn how to collect, analyze, and visualize data to answer life science questions, solve engineering problems, or explore physics phenomena.[16]

The tool-oriented perspective for CT often attracts critics. For instance, Mannila *et al.* described the proponents of CT as arrogant computer scientists who want to make everybody a software developer just because they are.[17] However, this long-standing issue has been noted even back to the early development of CT.[18,19] Papert's conceptualization for CT asserts that computing tools such as programming will teach students to think and help their cognitive development.[20] As such, CT integration should not center around the tool per se, instead computing tools are the medium for exploring other disciplines.[12,21] Wing, in her seminal paper on CT definition, also describes CT as "a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use".[15] Following this view on CT and its universally applicable nature for interdisciplinary problems, CT integration emphasizes the cognitive development and effect of CT as a thinking skill and how the thinking skill can be applied to contexts outside of computer sciences.

When discussing the development of CT, it is important to consider not only cognitive abilities but also the social and cultural contexts in which CT is learned and applied.[1] Focusing solely on cognitive development may overlook the ways in which social and cultural factors shape CT development, further perpetuating inequities and marginalizing underrepresented populations, which is already a persistent problem in computer sciences.[22] To address this concern, an integrated CT approach needs to consider a range of perspectives and experiences. This might involve incorporating CT into STEM education, which could provide a more interdisciplinary view of learning and potentially engage a wider audience in CT learning.[23] This integrated approach encourages more accessible, indigenous, and critical perspectives to view CT learning that potentially engages a larger body of audience who may be traditionally marginalized.[24]

Researchers have suggested that a literacy perspective is necessary for the integration of CT learning. However, this does not mean that who is considered literate and what is considered legitimate should be defined under a static framework.[1] Instead, computational literacies should be understood in different contexts and scales, as the literacy concerning CT and its connection with STEM classrooms may vary. Therefore, Kaifai and Proctor propose a collaborative approach to building understandings of how to integrate CT.[1] As a result, the integration of CT may vary depending on cultural contextual objectives.

Although the researchers have emphasized the importance of integrating CT in STEM classrooms, the integration of CT in STEM classrooms is not a straightforward process and, as described, different theoretical perspectives exist on how to approach this integration. The different theoretical views on how CT should be integrated into STEM classrooms reflect the diverse challenges the research community faces. Consequently, these different perspectives have a significant impact on the practices and propose challenges as well.

EXPERIENCES AND CHALLENGES IN INTEGRATING CT IN STEM CLASSROOMS

Past CT research has reported various experiences when attempting to integrate CT into STEM classrooms.[8] Some of the practices have focused on students' CT learning. For instance, integrating CT into STEM classrooms has been shown to improve students' CT skills, including algorithmic thinking, decomposition, and pattern recognition.[25] On the other hand, some of the studies leverage CT learning as a way to improve STEM learning. For example, Hava and colleagues have posited that integrating CT has helped to make STEM learning more engaging and relevant to students, and they have reported the relationship between student STEM career interests and CT skills.[13] Besides the attitudinal impact, Farris *et al.* found that engaging fourth-grade children in computational modeling activity throughout the academic year helped students develop a deeper understanding of scientific concepts.[16]

Despite the potential benefits, integrating CT into STEM classrooms can be challenging. One common barrier is the lack of teacher training and support.[9] Many teachers report feeling ill-prepared to teach CT and lack the necessary knowledge and skills to integrate it effectively into their existing curricula because CT learning often involves using unfamiliar technologies.[7] Although many CT-integrated STEM learning opportunities are offered and participants may seem to be enthusiastic about CT learning, their understanding of CT is often within the context of coding, rather than from an integrated perspective.[10] Overall, while integrating CT into STEM classrooms has the potential to improve student learning and engagement, there are significant challenges that must be addressed.

GAPS BETWEEN THEORY AND PRACTICES

Over-investigation of tools

One of the most common and important gaps between the theory and practices for the integration of CT into K-12 STEM classrooms is the overemphasis on tool usage for both teachers and students. For instance, many professional development programs focus primarily on the teaching of tools, without sufficient emphasis on the underlying concepts of CT.[6] Similarly, many school CT initiatives only work with students to learn how to use various computing devices.[26] This approach could lead to the misinterpretation of CT as merely a set of skills that can be acquired and applied to specific tools, rather than a deeper understanding of CT as a problem-solving approach.

However, the underlying challenge is that although the theory asserts that the tool is not central to CT as thinking skills, it is crucial to recognize that the ability to use tools is an essential initial step in learning CT.[12,19] Tools such as blocked-based programming, syntax-based programming, or unplugged computing tools should only serve as a means of expressing CT concepts in a tangible and practical way. For example, Saxena *et al.* demonstrated that even young children at the age of 4–6, could learn basic pattern recognition and algorithmic design by engaging with computing tools.[27] By engaging with programming languages, students can learn to think algorithmically, break down complex problems into smaller parts, and develop logical reasoning skills. To achieve that, sufficient learning experiences with the tools per so is inevitable.

As a result, the practice of CT needs to start, but also often ends with tools even if the tools should not be the sole focus of instruction. In the literature on teacher professional development in CT, a considerable amount of learning experiences is found to increase teachers' confidence and the adoption of computing tools.[6] However, even for professional development opportunities that focus on moving teachers beyond the scope of tool adoption, many teachers need to spend a significant amount of time at the beginning of professional learning to familiarize themselves with the tools before starting to develop an understanding of CT. For instance, Kayumova *et al.* explored the 16 teachers' CT professional development engagement and reluctance. They have identified that at the beginning of professional learning, teachers are most challenged by new tools, such as learning to code and setting up devices. Almost all teachers had to experience technical and logistics barriers because they felt they needed to first figure out how to use the tools and develop the ability to solve technical problems students might encounter in the classroom.[28] This pattern echoes the research in promoting teachers' technology-integrated practices: Teachers are often not confident with the tools to begin with, which tremendously prevents them from exploring further the deeper implications of CT.[29]

The integration of CT into K-12 STEM classrooms requires a balance between teaching the use of tools and developing a deeper understanding of the underlying CT concepts. Professional development programs should strive to emphasize the conceptual understanding of CT, rather than the mere use of tools. By doing so, teachers can better facilitate the integration of CT into their classrooms and provide students with a solid foundation in CT skills.

Trivialization and essentializing of CT

As described previously, CT is not a clearly defined subject, and the integrated view of CT may have different levels from a theoretical perspective. However, such theoretical complication is not accessible to practitioners, which often results in problems of trivialization and essentialization (see discussions around culturally responsive pedagogy[30]). While these mindsets may represent an intermediate stage of CT learning, they create a significant gap between theory and practice.

Trivialization occurs when CT is reduced to a set of technical skills and concepts related to how to use computing tools, without considering its broader implications and applications. For instance, some teachers may focus solely on teaching coding syntax and manipulation of computing tools without exploring the creative and problem-solving aspects of CT. This approach may lead students to view CT as a mere tool for building apps or games, rather than a way of thinking and reasoning about complex problems. Only experiencing CT learning within one single context (*e.g.*, games) may significantly limit students' ability to transfer the learned skills to other tasks.[31]

Essentialization, on the other hand, refers to the oversimplification and stereotyping of CT, assuming that CT skills, essentially, are just a simple combination of cognitive skills such as algorithmic thinking, abstraction, and pattern recognition.[2] For example, participants in professional development may demonstrate engagement patterns of paraphrasing and referring to keywords of the CT concepts and it is often an oversimplification of the content.[32] This essentialization is sometimes also encouraged by the instructors. For instance, in one of the professional development efforts I am involved in, teachers often were confused with the idea of abstraction in CT and asked for clarification. The facilitator could not provide a straightforward answer and had to essentialize abstraction as identifying the most important thing. As a result, the teachers interpreted understanding the problem components, which is problem decomposition, as abstraction as well. Moreover, teachers implemented the misconception with all student levels assuming that CT should be taught in the same way across all subjects and grade levels. This approach may lead to a lack of attention to the contexts of learning, such as students' backgrounds, interests, and learning needs, and may reinforce stereotypes and biases related to gender, race, and socioeconomic status.[1]

It is important to recognize that trivialization and essentialization are not necessarily permanent mindsets, but rather an intermediate stage of CT learning. Longitudinal evidence from both teachers' CT learning and students' CT learning revealed that they can move beyond these limiting perspectives and develop a more nuanced and comprehensive understanding of CT.[9,33] However, if we do not address these issues and move beyond this intermediate stage of CT learning, there are potential harms. Trivialization may lead to a lack of creativity and innovation in CT applications, limiting students' ability to solve complex problems in different contexts. Essentialization may perpetuate stereotypes and biases, leading to inequitable access and outcomes in CT education. Therefore, it is crucial to provide teachers and students with the necessary tools and resources to develop a deeper and more meaningful understanding of CT, one that goes beyond technical skills and concepts and embraces the broader implications and applications of CT thinking.

One-size-fits-all solution to CT learning

Another significant issue in the integration of CT into K-12 STEM classrooms is the gap between the theory that CT may look different in each context and the practice of designing a one-size-fits-all tool or curriculum. CT is not a fixed set of skills or concepts that can be easily taught or learned. Rather, it is a flexible problem-solving approach that can be applied in diverse contexts and for different purposes.[1] However, in many cases, both teachers and students ask for simplified curricula or tools that can be easily taught or learned. Pre-packaged and pre-scripted curricula are also easy to promote.[34] All these reasons can lead to a misinterpretation of CT as a set of fixed skills that can be acquired and applied in a particular context.

The situated framing of learning implies that CT may look different in each context, depending on the context, scope, and purpose because of the uniqueness of each community of practice.[35] For example, CT in students' STEM learning would look different from CT in a computer science class, as the shared goals of two different communities may differ. Weintrop *et al.* developed a taxonomy for CT practices in STEM learning and identified the commonly shared practices between CT and STEM learning.[23] Although the taxonomy is decontextualized, it provides the teachers a great tool to recognize CT practices within their context and implement CT in their classrooms in a flexible way depending on the needs. The integration of CT into K-12 STEM classrooms requires a flexible and adaptable approach to CT instruction, tailored to the specific context and audience. CT is not a fixed set of skills or concepts that can be easily taught or learned, but a flexible problem-solving approach that can be applied in diverse contexts and for different purposes. Therefore, designing CT curricula that are tailored to the specific context and audience is essential to effectively bring CT into STEM classrooms.[36]

However, the practice of designing CT curricula often ends up being a one-size-fits-all approach, which may start with recognizing the contextualized meaning of CT learning within local communities but then stop evolving once the resources are developed. Many curricula and tools are designed to be easily taught or learned, with a focus on specific skills or concepts, rather than the broader problem-solving approach of CT.[9,37] This gap between theory and practice in CT curricula can lead to a persistence challenge in effectively bringing CT into STEM classrooms.

Short-term intervention

The final gap I will discuss is that many programs about CT integration are short-term initiatives while impactful system change needs sustainable engagement. The lack of sustainable models for building a learning community for students and teachers is a significant gap in integrating CT into K-12 education. While various initiatives have been implemented to integrate CT into K-12 education, they have been short-term interventions with limited lasting impact.[6] The issue is not unique to CT integration,[38] but it is particularly important in this case, as CT also interconnects with many other important disciplines.

One-shot learning experiences for teachers and students are common in K-12 STEM classrooms. This approach involves a single session or workshop in which teachers or students are introduced to CT concepts and tools. While these one-shot learning experiences can be beneficial in raising awareness of CT and introducing basic concepts, they are often not effective in developing a deep and meaningful understanding of CT.

At the teachers' end, many professional development are organized in the form of one-day workshops or summer institutes.[6] Although these short professional development opportunities are, arguably, the most practical and straightforward way of introducing new ideas to teachers, past studies on teacher professional development have already illustrated that this one-shot model is not effective in bringing system changes.[38] Desimone outlined the levels of impact on teachers' professional development and urged the research community to move away from immediate evaluations of professional development outcomes.[39] In many cases, teachers may not have enough time to fully understand the implications of integrating CT into their teaching practices, resulting in surface-level implementation. Additionally, teachers may not have the necessary support to continue integrating CT into their teaching practices after the short-term intervention ends.[40] Without sustained engagement and support, teachers may not be able to fully integrate CT into their teaching practices, resulting in a limited impact on student learning.

The same problem also reflects on the students' end: Students often experience the CT integration learning experience as a temporary introduction to the tools and some big ideas. In very few cases, we see students' longitudinal engagement with CT learning and STEM integration.[33] As a result, oftentimes, the short-term effect of CT learning or students' enthusiasm can be confounded with the novelty effect. And once the intervention from the research team is removed, the computing tools will stay on the shelf and gather dust.[41] While they may be enthusiastic about learning CT at first, this initial excitement may not be sustained without ongoing engagement and support. In some cases, students may not have the opportunity to continue practicing CT skills after the short-term intervention ends. As a result, the impact of CT learning on student learning may be limited.

The lack of a sustainable model for building a learning community for both teachers and students in integrating CT with STEM highlights the need for a long-term approach to CT integration. This approach should aim to create a culture of CT and STEM integration in schools, which will help to sustain CT practices in STEM classrooms. Teachers and students need ongoing engagement and support to fully integrate CT into their teaching and learning practices. There is a need for a long-term approach that creates a culture of CT and STEM integration in schools, which will help to sustain CT practices in STEM classrooms. Without such an approach, the impact of CT integration on student learning will remain limited.

This gap between commonly practiced one-shot learning opportunities and the prolonged engagement of CT learning can lead to a persistent challenge in effectively bringing CT into K-12 STEM classrooms. Teachers and students may not develop a deep and meaningful understanding of CT, which can limit their ability to use CT skills to tackle real-world problems and contexts.[6]

BRIDGING THE GAP: THEORETICAL AND PRACTICAL RECOMMENDATIONS

The gaps between CT integration theories and practical implementations present persistent challenges that need to be addressed in order to effectively bring CT into K-

12 STEM education. While the enthusiasm for CT research and practice is high, successfully integrating CT with STEM requires bridging these gaps between theory and practice. In this section, I will discuss a conceptual model for bridging the aforementioned gap (Figure 1). The model aims to provide a framework for analyzing and addressing the persistent gaps in integrating CT into STEM classrooms. It highlights the need for multilevel interventions that bridge conceptual, practical, and community aspects.

Interconnections in the gaps and the conceptual model

Before discussing the recommendations, it is worth noting that the gaps discussed in this article are also interconnected. When the focus is heavily placed on learning to use CT tools like block-based programming languages, the deeper conceptual understanding of CT tends to get overlooked. Teachers and students start viewing CT as simplistic skills and procedures for manipulating these tools, rather than as a complex problem-solving process, which could lead to trivialization and essentialization.[6,29] When CT is reduced to step-by-step skills (*i.e.*, trivialization and essentialization), it becomes easy to design generic curricula, tools, and strategies that would be uniformly applied across different contexts.[1] However, this contradicts the situated nature of CT, which necessitates adapting CT integration based on the specific goals, cultures, and practices of each learning community. In turn, with a belief that a generic curriculum or tool can effectively teach CT to all students, there is less motivation to sustain long-term, customized integration tailored to local contexts.[36] Short teacher workshops or student summer camps seem sufficient as ways to deliver standardized CT content to everyone. Just like the complicated nature of the gaps, the potential solutions to bridge the gap would also be multileveled.

To address the challenges brought situated in the interconnected gaps, the proposed conceptual model (Figure 1) draws on existing theories related to the design of effective professional learning, with a focus on practical implementation and collaborative participation.[39,42] These theories provide insights into how to effectively bring about practice change and involve various stakeholders as a community, recognizing the long-term impact of such changes.[43,44] Therefore, the model presents a multi-leveled and interconnected way to address the challenges. It discusses potential approaches at three levels: the conceptual level, the practice level, and the community level. These levels recognize that addressing the gaps requires a comprehensive and systematic solution that encompasses different aspects of CT integration in STEM classrooms. It is important to note that the recommendations provided within each level are not exhaustive or prescriptive. They are meant to inspire further exploration and serve as a starting point for researchers and practitioners to consider a systematic approach to bridging the gaps.

Building technology self-efficacy

While an over-emphasis on tools should be avoided, building basic technology self-efficacy is an important first step, especially for teachers and students new to CT. At the initial stages of engaging with new tools like block-based programming platforms, a sense of confidence in one's ability to use the technology is needed before deeper CT skills can be developed.[45] Teachers need to feel comfortable navigating and troubleshooting the tools before they can shift focus to the conceptual teaching of CT.[46]

Similarly, students need a level of fluency with the technology in order to then apply it for creative problem solving versus getting stuck on syntax errors.[47] Building self-efficacy can facilitate initial engagement with novel technologies when integrating CT. It is important to provide learners hands-on learning opportunities that enable them to explore CT concepts using computing tools. For instance, past research has demonstrated that partial pair programming could help students develop their CT and self-efficacy.[48]

However, the end goal should be to use these tools as vehicles for CT skill development, not mastery of the tools themselves.[18] Once basic fluency is achieved, the educational focus should move towards conceptual understanding and connecting CT across disciplines. Scaffolded, sustained engagement can gradually transition learners from tool fluency to deeper competencies.

Developing a nuanced understanding of CT

Given the diversity in definitions and endpoints of CT learning, it is important to engage learners in discussing and defining CT within their own contexts. This avoids essentialization or trivialization of CT as just a set of predefined practices. Teachers should be encouraged to explore pedagogical approaches that emphasize conceptual understanding of CT, rather than mere tool use. For example, Ketelhut *et al.* found that integrating CT into the curriculum through project-based learning helped students develop a deeper, more nuanced understanding of CT concepts and improved their pedagogical content knowledge around how to integrate CT into STEM learning.[9]

Rather than treating CT as fixed skills to be acquired, teachers can engage students in defining what CT means in their learning community and empower them to apply

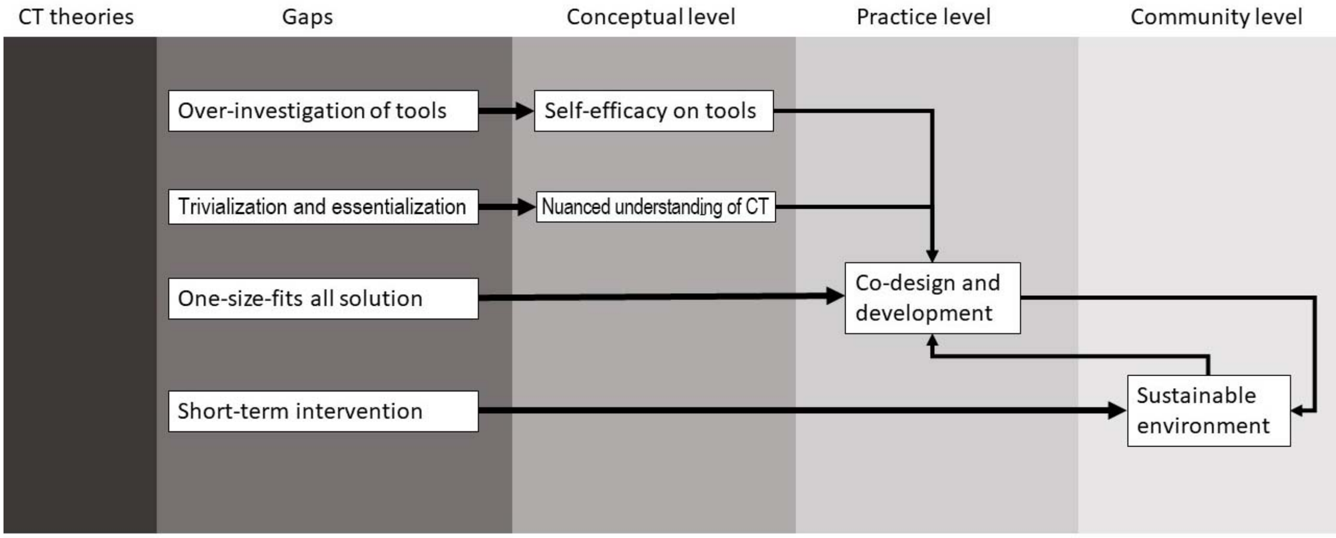

Figure 1. A conceptual framework for bridging the gaps between theories and practices. CT, computational thinking.

CT as a way of thinking to solve authentic, situated problems.[1] Learners can be guided to move beyond viewing CT as simplified cognitive skills and instead explore various views and conceptualizations of CT. With continued, contextualized engagement, the nuances of CT and its possibilities within different subjects can emerge over time. The goal is for both teachers and students to see CT as an evolving set of context-dependent practices, not a generic checklist of competencies.

In addition, learners could be encouraged to engage in reflective practice, which involves reflecting on their own experiences and beliefs about CT and how they can integrate CT into their teaching practices in a way that is meaningful and effective for their students. Researchers have demonstrated that reflective experiences could be an important way for teachers to develop nuanced CT understanding over time.[9,10,40] This way, teachers can develop a more contextualized understanding of CT and avoid essentializing or trivializing CT concepts and practices.

Co-design and development approach

One of the ways to achieve a contextualized understanding and implementation of CT learning is to engage teachers through co-design and development activities. Teachers are not provided with pre-packaged curricula and the learning community works together collaboratively to design CT curricula and select tools that are appropriate for the local contexts, needs, and cultures. The co-design process emphasizes participatory practices where educators and learners are actively involved in creating customized CT learning experiences, rather than being passive recipients of pre-packaged, generic instructional materials.[49]

Co-design activities that involve teachers in the design of CT curricula and activities can help to foster a deeper and more contextualized understanding of CT. Teachers could be encouraged to collaborate with CT experts, other educators, and students to design CT learning activities that are relevant, engaging, and accessible to all learners. This way, teachers can develop a better understanding of the diverse needs and interests of their students and design CT learning experiences that are tailored to the specific context and audience. For instance, researchers have developed co-design workshops with teachers. After the co-design experience, participants were able to integrate multiple computing tools that engage students in various CT-STEM practices.[50]

Moving towards a sustainable learning model

Finally, CT learning needs to become a prolonged process both for teachers and students. The learning experience should be sustained by the entire community of practice, not necessarily just the teachers and the students, but also the administrators, information technology (IT) or pedagogical supports, and parents.[40] Community engagement and commitment will be able to sustain the practice of CT learning. To that end, the gap also calls for a sustainable learning model that takes a community perspective. A sustainable learning model can help all the members of the learning community develop their CT knowledge and skills over time, integrate CT into their learning practices, and collaborate with other educators and stakeholders to improve CT learning.

The past literature on teacher professional development has highlighted the importance of engaging various stakeholders in the learning community. For instance, Simmonds *et al.* highlighted how support from administrators could be crucial in building a long-term CT program.[51] In addition to the existing stakeholders, the CT learning community could engage learners outside of the existing community and move the existing members forward by transforming their roles.[52] In this way, the entire community can collaborate to create a CT-rich learning environment that could be sustained for a long time.[52]

Finally, as discussed previously, the recommendations to address the gap are going to be multileveled because of the complicated nature of the problem. It is worth noting that the recommendations do not necessarily follow a linear pattern. Although the conceptual level recommendations (*i.e.*, building self-efficacy and nuanced understanding of CT) provide the foundations for the practice level recommendation (*i.e.*, co-design and co-development). The practice-level recommendation also could be an important step to take at the community level. However, the community level changes would also impact the co-design and co-developed practices. Through these practical changes, self-efficacy and nuanced understanding of CT may also emerge.

IMPLICATIONS FOR PRACTICES FUTURE RESEARCH

The integration of CT into K-12 STEM classrooms has the potential to improve student learning outcomes, promote critical thinking and problem-solving skills, and prepare students for the demands of the 21st-century workforce.[1] However, there are several gaps between what the theory promotes and what the practices look like in the current effort in CT integration. Closing these gaps would be important for moving CT education forward.

As discussed previously, all stakeholders could have an important role to play in promoting CT learning and CT integration in the learning community. One important step is to collaborate with other educators, researchers, and stakeholders to develop and implement effective CT integration curricula and teaching practices. This could involve participating in professional development programs, engaging in collaborative lesson planning, and sharing best practices with other educators. By fostering a culture of collaboration and innovation, the entire community can work together to create a CT-rich learning environment that supports student learning and success.

However, the research on how to address the gaps between the theory and practices is limited. Many efforts in CT learning often lack a balance between tool usage and conceptual understanding, are insufficient in building confidence in using tools and overall technology efficacy, and may essentialize or trivialize CT concepts and practices.[6] Very few studies take a longitudinal approach or a community approach to examine how CT can be integrated into STEM classrooms.[9,33] Further research is needed to identify effective strategies for bridging these gaps and creating sustainable learning models that support the ongoing learning and growth of the community.

CONCLUSION

The integration of CT into K-12 STEM classrooms holds great promise for improving student learning outcomes and preparing students for the demands of the 21st-century workforce. However, addressing the gaps in current professional development programs for CT integration is essential for realizing this potential. By working together to develop effective CT curricula and teaching practices, involving the entire community of practice in CT education, and fostering a culture of continuous improvement and innovation, educators and stakeholders can help to create a CT-rich learning environment that supports student learning and success.

Overall, this article seeks to contribute to the ongoing discussion on how to effectively incorporate CT into the K-12 curriculum. The article highlights the importance of CT as a literacy skill and proposes a framework that can help students develop these skills in a meaningful way. By doing so, the article aims to help educators and policymakers create a more inclusive and effective approach to CT education in K-12 schools.

CT education is becoming increasingly important for success in the 21st-century workforce. It can help students develop critical thinking skills, problem-solving abilities, and creativity, which are all essential for success in a technology-driven world. However, there are significant challenges associated with incorporating CT into K-12 curriculum. These challenges include the lack of teacher training, the lack of access to technology, and the need for more comprehensive CT curricula. Addressing these challenges will require a collaborative effort between educators, policymakers, and the broader community. By working together, we can help ensure that all students have the opportunity to develop the CT skills they need to succeed in the 21st century.

DECLARATION

Author contributions

Liu Z: Conceptualization, Data curation, Methodology, Investigation, Resources, Writing—Original draft, Writing—Review and Editing, Visualization, Supervision, Project administration, and Funding acquisition. The author have read and approved the final version.

Source of funding

The work described in this paper was fully funded by a grant from the University Research Committee (No. 2201101075).

Conflict of interest

The author has no conflicts of interest to declare.

Data sharing

Data used to support the findings of this study are available from the corresponding author upon request.

REFERENCES

1. Kafai YB, Proctor C. A revaluation of computational thinking in K–12 education: Moving toward computational literacies. *Educ Res*. 2022;51(2):146–151.
2. Shute VJ, Sun C, Asbell-Clarke J. Demystifying computational thinking. *Educ Res Rev*. 2017;22:142–158.
3. Lodi M, Martini S. Computational thinking, between Papert and Wing. *Sci Educ*. 2021;30(4):883–908.
4. ISTE, CSTA. Operational definition of computational thinking for K-12 Education. Accessed December 28, 2023. https://cdn.iste.org/www-root/Computational_Thinking_Operational_Definition_ISTE.pdf
5. Zhang L, Nouri J. A systematic review of learning computational thinking through Scratch in K-9. *Comput Educ*. 2019;141:103607.
6. Liu ZC, Gearty Z, Richard E, Orrill C, *et al*. Computational thinking into k-12 classrooms: Experiences and challenges from professional learning experiences. In: Chinn C, Tan E, Chan C, Kali Y, eds. *Proceedings of the 16th International Conference of the Learning Sciences-ICLS 2022*. International Society of the Learning Sciences; 2022;2022:2100–2101.
7. Hickmott D, Prieto-Rodriguez E. To assess or not to assess: tensions negotiated in six years of teaching teachers about computational thinking. *Inform Educ*. 2018;17(2):229–244.
8. Wang C, Shen J, Chao J. Integrating computational thinking in STEM education: A literature review. *Int J Sci Math Educ*. 2022;20(8):1949–1972.
9. Ketelhut DJ, Mills K, Hestness E, Cabrera L, Plane J, McGinnis JR. Teacher change following a professional development experience in integrating computational thinking into elementary science. *J Sci Educ Technol*. 2020;29:174–188.
10. Dong Y, Catete V, Jocius R, *et al*. PRADA: A practical model for integrating computational thinking in K-12 education. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery; 2019: 906–912.
11. Ching YH, Hsu YC, Baldwin S. Developing computational thinking with educational technologies for young learners. *TechTrends*. 2018;62:563–573.
12. Grover S, Pea R. Computational thinking in K–12: A review of the state of the field. *Educ Res*. 2013;42(1):38–43.
13. Hava K, Koyunlu Ünlü Z. Investigation of the relationship between middle school students' computational thinking skills and their STEM career interest and attitudes toward inquiry. *J Sci Educ Technol*. 2021;30(4):484–495.
14. Israel M, Pearson JN, Tapia T, Wherfel QM, Reese G. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Comput Educ*. 2015;82:263–279.
15. Wing JM. Computational thinking. *Commun ACM*. 2006;49(3):33–35.
16. Farris AV, Dickes AC, Sengupta P. Development of disciplined interpretation using computational modeling in the elementary science classroom. In: *Proceedings 2016 ICLS*. ISLS; 2016.
17. Mannila L, Dagiene V, Demo B, *et al*. Computational thinking in K-9 education. In: *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference*. Association for Computing Machinery; 2014: 1–29.
18. Papert S. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books; 1980.
19. Papert S. *The Connected Family: Bridging the Digital Generation Gap*. Taylor Trade Publishing; 1996.
20. Lewis CM. Good (and bad) reasons to teach all students computer science. In: Fee SB, Holland-Minkley AM, Lombardi TE, eds. *New Directions for Computing Education: Embedding Computing across Disciplines*. Springer; 2017.
21. diSessa AA. *Changing Minds*. The MIT Press; 2000.
22. Vrieler T, Nylén A, Cajander A. Computer science club for girls and boys–a survey study on gender differences. *Comput Sci Educ*. 2021;31(4):431–461.
23. Weintrop D, Beheshti E, Horn M, *et al*. Defining computational thinking for mathematics and science classrooms. *J Sci Educ Technol*. 2016;25(1):127–147.
24. Kaifai YB, Fields DA, Lui DA, *et al*. Stitching the loop with electronic textiles: Promoting equity in high school students' competencies and perceptions of computer science. In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery; 2019: 1176–1182.
25. Arastoopour Irgens G, Dabholkar S, Bain C, *et al*. Modeling and measuring high school students' computational thinking practices in science. *J Sci Educ Technol*. 2020;29:137–161.
26. Bers MU, Flannery L, Kazakoff ER, Sullivan A. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Comput Educ*. 2014;72:145–157.
27. Saxena A, Lo CK, Hew KF, Wong GKW. Designing unplugged and plugged activities to cultivate computational thinking: An exploratory study in early childhood education. *Asia Pac Educ Res*. 2020;29(1):55–66.
28. Kayumova S, Asif AD, Richard E, *et al*. *Exploring elementary teachers' eagerness and reluctance to integrating computational thinking*. AERA; 2023.
29. Kao CP, Wu YT, Chang YY, Chien HM, Mou TY. Understanding web-based professional development in education: The role of attitudes and self-efficacy in predicting teachers' technology-teaching integration. *Asia Pac Educ Res*. 2020;29:405–415.
30. Sleeter CE. An agenda to strengthen culturally responsive pedagogy. *Engl Teach: Pract Crit*. 2011;10(2):7–23.
31. Liu Z, Jeong AC. Connecting learning and playing: The effects of in-game cognitive supports on the development and transfer of computational thinking skills. *Educ Technol Res and Dev*. 2022;70:1867–1891.
32. Zhao Y, Liu Z, Orrill CH, Kayumova S, Balasubramanian R. Designing professional learning workshop for shaping teachers' learning pedagogical content knowledge in computational thinking. In: Blikstein P, Van Aalst J, Kizito R, Brennan K, eds. *Proceedings of the 17th International Conference of the Learning Sciences - ICLS 2023. International Society of the Learning Sciences*; 2023: 2071–2072.
33. Kong SC, Wang YQ. Monitoring cognitive development through the assessment of computational thinking practices: A longitudinal

intervention on primary school students. *Comput Hum Behav*. 2023;145:107749.

34. Murray R, Shea M, Shea B, Harlin R. Issues in education: Avoiding the one-size-fits-all curriculum: Textsets, inquiry, and differentiating instruction. *Child Educ*. 2004;81(1):33–35.
35. Sfard A. On two metaphors for learning and the dangers of choosing just one. *Educ Res*. 1998;27(2):4–13.
36. Lee I, Malyn-Smith J. Computational thinking integration patterns along the framework defining computational thinking from a disciplinary perspective. *J Sci Educ Technol*. 2020;29(1):9–18.
37. Lehmkuhl-Dakhwe KV. An instructional framework, model lessons, and professional learning program for science standards-aligned computing in 4th-12th grade classrooms. In: *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE; 2018: 1–5.
38. Darling-Hammond L. Teacher education around the world: What can we learn from international practice? *Eur J Teach Educ*. 2017;40(3):291–309.
39. Desimone LM. Improving impact studies of teachers' professional development: Toward better conceptualizations and measures. *Educ Res*. 2009;38(3):181–199.
40. Choate B, Dubosarsky M, Chen K. An innovative professional development model for teaching robotics to novice educators. In: *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE; 2018:1–5.
41. Tsai CW, Shen PD, Tsai MC, Chen WY. Exploring the effects of web-mediated computational thinking on developing students' computing skills in a ubiquitous learning environment. *Interact Learn Environ*. 2017;25(6):762–777.
42. Clarke D, Hollingsworth H. Elaborating a model of teacher professional growth. *Teach Teach Educ*. 2002;18(8):947–967.
43. Gibbons LK, Lewis RM, Nieman H, Resnick AF. Conceptualizing the work of facilitating practice-embedded teacher learning. *Teach Teach Educ*. 2021;101:103304.
44. Lefstein A, Louie N, Segal A, Becher A. Taking stock of research on teacher collaborative discourse: Theory and method in a nascent field. *Teach Teach Educ*. 2020;88:102954.
45. Yildiz Durak H, Atman Uslu N, Canbazoğlu Bilici S, Güler B. Examining the predictors of TPACK for integrated STEM: Science teaching self-efficacy, computational thinking, and design thinking. *Educ Inf Technol*. 2023;28(7):7927–7954.
46. Jaipal-Jamani K, Angeli C. Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking. *J Sci Educ Technol*. 2017;26:175–192.
47. Durak HY, Yilmaz FGK, Yilmaz R. Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. *Contemp Educ Technol*. 2019;10(2):173–197.
48. Wei X, Lin L, Meng N, Tan W, Kong SC. The effectiveness of partial pair programming on elementary school students' computational thinking skills and self-efficacy. *Comput Educ*. 2021;160:104023.
49. Lee I, Grover S, Martin F, Pillai S, Malyn-Smith J. Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *J Sci Educ Technol*. 2020;29(1):1–8.
50. Wu S, Peel A, Bain C, Anton G, Horn M, Wilensky U. Workshops and co-design can help teachers integrate computational thinking into their k-12 stem classes. In: *Proceedings of International Conference on Computational Thinking Education*. 2020: 63–68.
51. Simmonds J, Gutierrez FJ, Casanova C, Sotomayor C, Hitschfeld N. A teacher workshop for introducing computational thinking in rural and vulnerable environments. In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education. Association for Computing Machinery; 2019: 1143–1149.
52. Hestness E, Ketelhut JD, McGinnis JR, Plane J. Professional knowledge building within an elementary teacher professional development experience on computational thinking in science education. *J Technol Teach Educ*. 2018;26(3):411–435.