Computers and Education Open 4 (2023) 100122

Contents lists available at ScienceDirect

Computers and Education Open

journal homepage: www.sciencedirect.com/journal/computers-and-education-open

A systematic review of integrating computational thinking in early childhood education

Jiahong Su [a,*], Weipeng Yang [b]

[a] Faculty of Education, The University of Hong Kong, Hong Kong SAR, China
[b] Department of Early Childhood Education, The Education University of Hong Kong, Hong Kong SAR, China

ARTICLE INFO

Keywords:
Teaching/learning strategies
Pedagogical issues
21st century abilities
Early years education
Teacher professional development

ABSTRACT

Computational thinking education has become an increasingly popular topic among practitioners and researchers. However, rare is known how to effectively teach and learn computational thinking in early childhood education. To address this knowledge gap, this systematic review examined 26 studies on the teaching and learning of computational thinking in early childhood education from 2010 to 2022. The content knowledge, tools, pedagogical design, assessment methods, and learning outcomes were analyzed. Results indicated that, with age-appropriate instructional design, children could develop early concepts and skills of computational thinking, as well as other related skills such as communication, collaboration, and problem solving. Across the studies, we found that most studies used quantitative research methods, with direct assessment and observation being the most. Several challenges were identified: (1) achieving a deeper learning of computational thinking; (2) a lack of valid and reliable computational thinking assessments for children with a wider age range; (3) selecting appropriate learning tools; and (4) designing age-appropriate activities for young learners. Although with these challenges, computational thinking education could bring new learning opportunities and enhance children's computational thinking skills, as well as other non-cognitive skills such as critical thinking, body-material interaction, and hand-eye coordination. This systematic review informs future endeavors in theorizing a digital learning framework that can integrate computational thinking into early childhood education.

Introduction

Recently, the importance of computational thinking (CT) in K–12 education has been highlighted (e.g., [1,2]). CT was first introduced by Papert [3], who defined it as procedural thinking and programming. Years after, Wing [4] further defined CT as one of the most important problem-solving skillsets that everyone could learn, instead of merely computer scientists. Particularly, in the educational context, CT refers to the processes that enable students to formulate problems and identify solutions that are presented in a form that could be conducted by information-processing and programmable agents [5]. Through interacting with the agents (e.g., robotics, objects in the Scratch program, and electronic toys), students can consider steps and use technical skills to manipulate the machines/agents to solve problems (e.g., [6–8]). However, there is a lack of systematic knowledge about the integration of CT in early childhood education (ECE)—a field that is significantly different from formal schooling.

Definitions of CT

CT is a 21st century skill that influences our everyday life and learning [9]. It is no longer merely considered as programming or computer skills that are required by computer scientists [4]. Researchers have defined it as a positive digital mindsets, attitudes and readiness towards understanding and using this digital literacy skill in our everyday life (e.g., [1,10]). CT allows us to obtain thinking ways that are similar to that of a computer scientist when facing problems such as simplifying, embedding, transforming, simulation, and system design [1,4]. In the stage of early childhood, children should not only develop their literacy skills such as reading, writing, and arithmetic, but also learn CT-related problem-solving skills such as logical thinking, sequencing abilities, abstraction, and algorithms [4]. Regarding

We have no known conflict of interest to disclose.This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

* Corresponding author at: The University of Hong Kong, Room 219, Runme Shaw Building, Pokfulam Road, Hong Kong.
E-mail address: maggiesu@connect.hku.hk (J. Su).

https://doi.org/10.1016/j.caeo.2023.100122
Received 22 August 2022; Received in revised form 29 December 2022; Accepted 4 January 2023
Available online 5 January 2023
2666-5573/© 2023 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

computational skills and computer science concepts, CT can be categorized by various problem-solving strategies such as sequencing, creative design, and content generation [11,12].

However, CT is not restricted to using computers to learn. Wing [13] further elaborated CT as a way of human thinking, a combination of mathematical and engineering thinking, problem-solving skills in our life that facilitate how people communicate and interact with others using CT tools. On top of cognitive skills and practices of computational and problem-solving skills, CT is further conceptualized as new perspectives in Brennan and Resnick [14]'s framework. In the model, students could gain perspectives that they interpret about the world around them and about themselves in terms of expressing, connecting and questioning.

Through this discussion, we can see that prior literature has indicated consensus on computational concepts (e.g. sequence, variables, and conditionals). However, the discussion on how to use CT across different contexts (e.g., mathematics, storytelling, and vocabulary learning) is limited. To explain important CT definitions and its related terms (e.g., concepts, pedagogues, tools), the terminologies of CT are presented in Table 1.

Table 1 describes the important terms about CT knowledge, concepts and skills that are mentioned at least twice in our selected studies, including sequencing, conditionals/ control structures, iterations/ loops, testing and debugging, pattern recognition, algorithms, modularity representation, and problem-solving.

Importance of CT in ECE

CT has become an important concept in ECE and its significance grows in ECE with the emergence of age-appropriate technologies (e.g., [15,18]). Tang et al. [22] suggested that children could use CT skills to shape their learning and express their ideas (Papert, 1996). CT represents a "universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" ([4], p. 33). It is also an important skill for learning STEM [23].

Table 1
CT-related Terminologies.

Term	Definition	Sample studies
Sequencing	Sequencing ability is a cognitive ability that generates skills to arrange objects or actions in a correct order and procedural planning.	Relkin et al. [15]; Saxena et al. [16]
Conditionals/ control structures	Instruct the computer on the decision to make when given some conditions.	Bers et al. [17]; Pugnali et al. [18]
Iterations/ loops	Repeated processes in which the code segment is executed once	Bers et al. [11]; Pugnali et al. [18]
Testing and debugging	The process to find bugs and errors, and how learners correct the bugs found during testing.	Bers et al. [17]; Bers [6]
Pattern recognition	Creating rules, principles, and observed patterns in data.	Saxena et al. [16]
Algorithm Design	Creating ordered series of instructions to solve similar problems or to perform a task [10].	Clarke-Midura et al. [19]; Relkin et al. [7]
Modularity representation	A divide and conquer skill to separate the problems into smaller problems through sub-program/ modules.	Bers [6]; Relkin et al. [7]
KIBO	KIBO is an easy and fun way to bring robotics and coding to young learners and spark their interest in STEAM.	Pugnali et al. [18]; Relkin et al. [7]
Bee-Bots	Bee-Bot is a robot toy designed for young children for teaching sequencing and problem-solving.	Critten et al. [20]
ScratchJr	ScratchJr is a platform for young children (aged 5–7) to program their own interactive stories and games.	Papadakis et al. [21]; Papadakis & Kalogiannakis (2019)

Researchers have discussed the significance of CT learning at the early childhood stage, since kindergarteners could gain attitudes, mindsets, skills and knowledge about CT (e.g., [7,24]). With more and more age-appropriate CT instructional design for kindergarteners, empirical evidence has shown that children as young as three to six years old are able to build and program robots [11,17]. Some studies have explored how the design of different programmable agents can contribute to students' development of both CT cognitive abilities such as sequencing abilities, identification, pattern recognitions, and algorithms [7,21] and non-cognitive abilities such as collaboration, teamwork, communication, and creativity [20]. We can see that CT is not only grounded on concepts fundamental to computer science knowledge but it nurtures young children to become digital literates who can use CT tools to facilitate their learning and everyday life [22]. On top of CT-related digital skills, the smart devices also enable students to develop their fine-motor skills and hand-eye coordination [11], positive executive functioning, and learning behavior (e.g., self-regulation, persistence, and planning), so that they can successfully complete their planned tasks [25,26]. Moreover, teachers can also integrate academic content (e.g., concepts of engineering, storytelling, and mathematical ideas) in meaningful CT projects so that students can play to learn while learning in a creative way [11,22]. These studies mentioned different important abilities, skills and mindsets that kindergarteners need to learn at their young age, which CT can provide the learning opportunities for them to achieve.

Previous relevant reviews

Some researchers have conducted review studies on CT education, and the majority of them focused on later schooling, rather than on the early childhood stage [11]. For example, Grover and Pea [1] framed the current state of discourse on CT in K-12 education based on Wing [4]'s definition as a springboard to identify gaps in research and suggest future recommendations. Lye and Koh [27] presented the trends of empirical research on the development of CT from 2009 to 2013 and suggested possible research and instructional implications for K-12 education. In recent years, Shute et al. [28] reviewed the CT literature in K-16 settings and proposed a CT model. Likewise, Lockwood and Mooney [29] summarized CT research in secondary education in terms of the subjects used to teach CT, the tools adopted to teach and assess CT, and benefits and barriers of incorporating CT in secondary education. Zhang and Nouri [12], employing Brennan and Resnick's [14] framework, conducted a review of learning CT through Scratch in K-9 education. Hsu et al. (2021) conducted a meta-review of CT studies from 2006 to 2017, and identified the three most promising strategies (i.e., project-based learning, collaborative learning, and game-based learning), program design as the most common subject, and visual programming languages as the most common instruments to convey CT education. Tang et al. [22] reviewed the current CT assessments from kindergarten to higher education, in terms of context, construct, assessment type, and psychometric evidence. However, ECE hugely differs from primary and secondary education due to learners' characteristics. It remains less known how CT should be taught and learned in ECE.

Regarding CT education in early childhood, recent studies have started to discuss the types of robotics and programming tools used for CT instruction, characteristics of the activities, CT assessments, and the most influential researchers and countries in this area (e.g., [11,30]). Recently, more and more smart devices and electronic toys have been designed to provide young learners with playful learning opportunities in order to foster their computer science and CT skills [11,25,30]. This field has drawn upon a growing interest in integrating CT into ECE, and dedicated research efforts to CT teaching and assessment. These efforts include developing CT curriculum for young children (e.g., [19,26,31]), developing CT-driven teaching/learning tools (e.g., [7,32]), enhancing interactive and playful learning environments (e.g., [6]), as well as designing suitable assessments to examine children's CT skills (e.g., [17, 19]). These studies provide a significant body of literature that facilitates us to understand the nature and integration of CT instruction in ECE classrooms.

So far, only one review study has tried to document the CT education specifically for ECE [33]. This review selected 24 articles from Web of Science, Scopus, and ERIC databases. This study shows that age was an important factor in learning CT in early childhood. It identified that both plugged-in and unplugged applications improved children's CT skills through concrete experiences [33]. However, Bati's [33] study did not discuss the mapping of existing learning outcomes, assessment methods, as well as opportunities and challenges of CT education in the ECE settings. As such, this review aims to present a bigger picture of opportunities and challenges of CT education in early childhood. Of this interest, this review aims to understand the development and application of CT in ECE, including research methods, teaching strategies, learning outcomes, and challenges and opportunities. We analyzed the related CT in ECE literature from 2011 to 2022. Possible research directions, in terms of advancing teaching design and evaluation, are addressed as a reference for future research in this area.

Research objectives

This systematic review aims to assess, synthesize, and present current research on CT in ECE. The current review will offer a significant contribution to existing knowledge, because there are scarce review studies specifically focused on CT in ECE. ECE refers to the education and care of children from birth up to eight years of age. Although previous studies have brought CT into ECE classrooms and shown their promising effects (e.g., [7,11,16,31]), very little has been known about the challenges and opportunities of CT for ECE. Therefore, there is a need to timely analyze existing work focusing on the early CT development in order to explain the challenges and opportunities of CT in ECE. In order to address this issue, the current review asks the key question: How has CT been taught and learnt in ECE?

In this review, our objectives are: (1) to evaluate the instructional design, CT tools, pedagogical approaches, research methods, and research findings ascribed to the existing literature of studies on CT in ECE; and (2) to explore future research directions in terms of advancing teaching design and evaluation for early CT curriculum. The findings can help direct future research in CT tools, instructional design, learning outcomes, and assessment methods for CT in early childhood research, and meanwhile provide a useful guide for the design, implementation, and evaluation of CT in early childhood education research. Specifically, three research questions (RQs) guided this review:

RQ1: How were the CT activities designed and implemented for young children, as related to the instructional design and CT tools?

RQ2: What were the learning outcomes of CT curricula in ECE settings?

RQ3: What assessment methods were used to study the teaching and learning of CT in ECE settings?

Methods

This review was conducted to rigorously analyze, evaluate, and synthesize studies pertaining to the answer review questions. We followed the Preferred Reporting Items for Systematic Review (PRISMA) guidelines [34]. A review protocol was developed, describing the literature search process, eligibility criteria, data extraction, and data analysis procedures.

Literature search process

The electronic databases used for the literature search included (1) Web of Science and (2) Scopus. Referring to related early CT education search strings used in other studies (e.g., [30]), we used the following search string in this review: "computational thinking" OR "robot" OR "coding" OR "robotics" OR "programming" AND "early childhood" OR "young child*" OR "preschool*" OR "kindergarten*" OR "pre-k*" OR "childcare" OR "child care" OR "day care". To facilitate database search, this study examined peer-reviewed articles published until May 2022 when the literature search was conducted. All articles were accessed in May 2022. The data used for the analysis included titles, keywords, and main texts.

Eligibility criteria

As shown in Fig. 1, using the keyword search descriptors, 3160 articles were identified, 249 from Web of Science and 2911 from Scopus. The following articles were excluded based on their title and abstract: (1) studies irrelevant to the research topic ($n = 3033$). For example, first, the study is not related to CT. Second, the study only focuses on instructional design. Third, it is a literature review, discussion, and/or position papers; (2) duplicate studies ($n = 60$); (3) studies whose participants were not 3–8 years old ($n = 3$); (4) studies whose focus was not CT ($n = 6$); (5) studies that did not discuss curriculum/learning program/learning activities ($n = 13$); and (5) articles that were not journal articles ($n = 17$). Our inclusion criteria required all articles to be in English. As a result, 26 articles were thoroughly reviewed in the current study.

Data extraction

Data extraction was performed on an Excel Sheet which records several important information of the selected articles, including (1) research designs, (2) participants, (3) knowledge, (4) tools, (5) intervention time, (6) assessed, (7) location, (8) findings.

Data analysis procedures

To enhance validity and reliability, the literature was reviewed carefully to extract, code, and categorize systematically using content analysis procedures [35]. All included studies were coded and reviewed by two researchers. Disagreements were resolved by discussion among researchers to ensure over 80% inter-rater reliability. The coding framework of teaching and learning CT in the ECE studies in terms of instructional design, assessment methods, and learning outcomes (Table 2).

Findings

Although CT is an essential topic and has been examined around the globe, it has been inadequately investigated in the ECE context – 26 studies were identified in the literature from 2011 to 2022. We assumed that this number of research articles is sufficient to provide an exploratory view of early CT education.

Overview of the selected studies

Year of publications

Twenty-six articles that were focused on early CT were thoroughly reviewed (2010, 1 article, 2011, 1 article; 2013, 1 article; 2014, 2 articles; 2016, 3 articles; 2017, 3 articles; 2019, 4 articles; 2020, 4 articles; 2021, 5 articles; 2022, 2 articles). As shown in Fig. 2, summaries are developed based on the articles related to the author, year, location of study, research designs, sample, knowledge, tools, intervention, assessment, and findings of the included studies. Details of the included

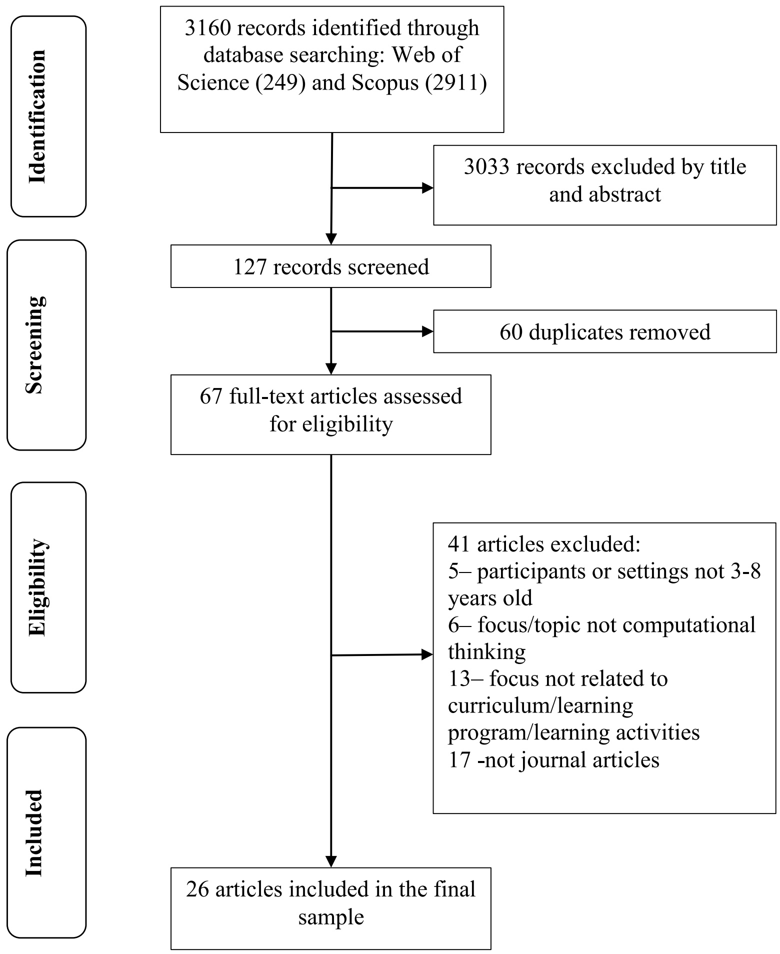

Fig. 1. PRISMA Diagram of Included Articles in the Systematic Review.

Table 2
Coding Framework.

Themes	Sub-themes	Samples
CT tools	KIBO	Relkin et al. [7]; Bers et al. [17]; Bers [6]
	Tangiblek	Bers et al. [11]; Bers [36];
	Bee-Bots	Critten et al. [20]
Instructional design	Positive Technological Development (PTD) Framework	Bers et al. [17]; Bers et al. [11]; Bers [6]
	Activity-based learning strategies	Cho and Lee [37]
Learning Outcomes	CT and coding skill	Relkin et al. [7]; Bers et al. [17]
	CT and programming concepts	Papadakis et al. [21]; Bers et al. [11]
	Communication and collaboration skills	Critten et al. [20]
	Hand and arm movements	Welch et al. [38]
Assessment Methods	Knowledge assessment	Papadakis et al. [21]
	CT skills assessment	Saxena et al. [16]
	Observation	Bers et al. [17]; Saxena et al. [16]

studies are presented in Appendix 1.

Countries/regions

Most of the studies ($n = 13$) reviewed were conducted in the United States. Five studies were implemented in four European countries, namely Spain, Portugal, and the United Kingdom, respectively. Seven studies were implemented in Asian countries/regions, including Hong Kong ($n = 1$), Korea ($n = 2$), Mainland China ($n = 3$), and Cyprus ($n = 1$). One study was implemented in South America, such as Uruguay. This distribution shows that ECE educators across North America, Asia and Europe have started their CT curricula for ECE levels; however, it has still received less attention in a global context.

Research methods

Most studies were found to use a quantitative research method, followed by the mixed-methods method (see Table 3). Out of the 26 selected studies, fifteen studies applied quantitative data collection methods, such as pre-and post-assessments through gameplay, CT knowledge, CT skills assessments, and surveys. For example, three studies used *TechCheck* assessments to examine children's CT skills [7, 15,25]. For example, Relkin et al. [7] used pre- and post-knowledge assessments to examine children's CT knowledge (i.e., algorithms, modularity, control structures, representation, hardware/software, and debugging) through *TechCheck* assessments.

Six studies used a mixed-methods approach to collect data through various procedures, such as observations, interviews, diary journals, questionnaires, pre-test/post-test, and assessments. Saxena et al. [16] used a mixed-methods research design to obtain data from performance assessments, classroom observations, and teacher interviews in a study of CT education for children ages 4 to 6. Their study revealed that children could acquire a variety of CT skills, including pattern recognition, sequencing, and algorithm design, through a mix of plugged and unplugged activities [16].

Six studies used a qualitative approach, and data was collected using observations, field notes, online form, and video analysis. For example, Welch et al. [38] used video analysis to analyze children's coding tasks

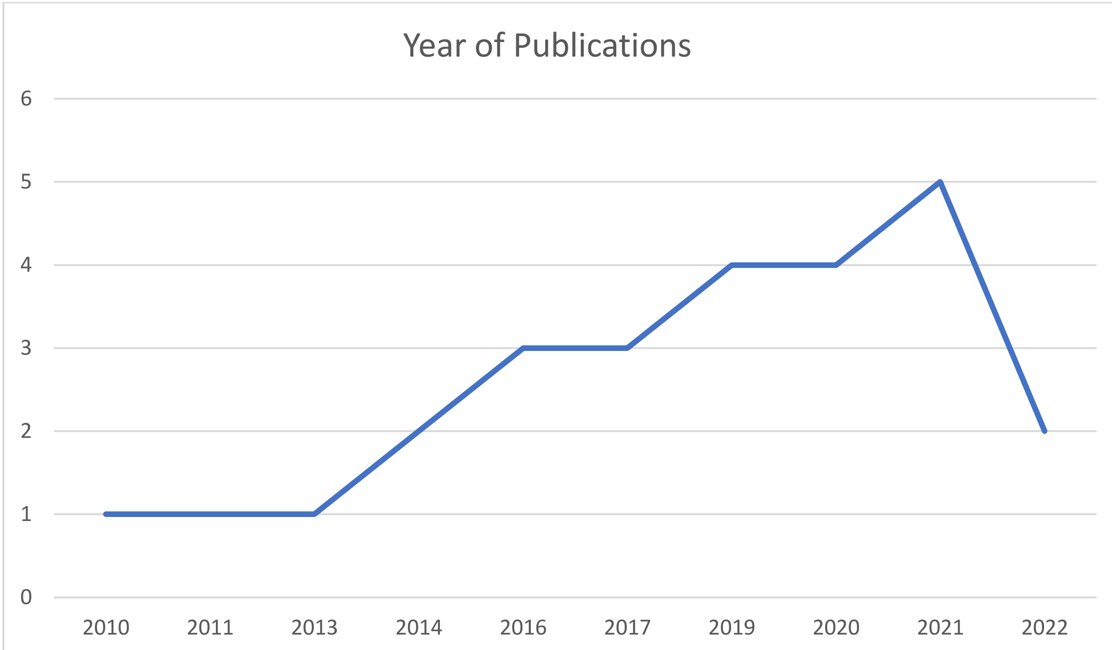

Fig. 2. Research publications in the area of CT in early childhood education.

including "analysing students' actions, gestures, and verbal responses with the robot and with each referencing the robot and its materials" (p.7). Results show that children used hand and arm movements (e.g., gestures) and vocal descriptions to describe a created conception of a dynamic linear unit, the coding toy had an impact on the children's expressions (the artifact).

Data collection methods

As shown in Table 3, in terms of the overall data collection techniques used, knowledge and skills assessments (13 articles) are the most usually used, followed by observations (5 articles).

CT tools and instructional design

In ECE, students could learn CT skills with age-appropriate tools and instructional design. Six of the 26 studies used KIBO as a platform for early childhood research CT. ScratchJr is the second most popular tool for early CT research, with five studies using it as a CT tool. Other coding platforms reported in the study include Bee-Bots ($n = 3$), TangibleK ($n = 3$), Daisy the Dinosaur ($n = 1$), Kodable ($n = 1$), Coding bots ($n = 1$), Aphid's Toys ($n = 1$), Matatalab ($n = 1$), CHERP ($n = 1$) and Cubetto ($n = 1$). See Table 4 for more details.

To investigate the key factors for successful CT activity design (See Table 5) in ECE, we examined a set of pedagogical elements in terms of instructional design, theory, and learning tools that were used in the curriculum to scaffold children's CT understandings. First, five studies designed curricula to enhance children's learning of CT concepts (e.g., sequencing, repeats, and conditionals) using CT tools (e.g., [16,17,20, 24,25]). For example, Pugnali et al. [18] designed a course that engaged students in exploring CT concepts (e.g., sequencing, repeats, and conditionals) through related tools (e.g., KIBO, ScratchJr) in the USA. Results showed that young children could learn foundational CT skills from suitable curricula. Second, five studies have applied the Positive Technological Development (PTD) framework to instructional design (e.g., [6,11,17,18,36,49]). The framework includes three main components, namely assets, behaviors, and classroom practices [17], and the six Cs (i. e., communication, collaboration, community building, content creation, creativity, and choice of conduct) to empower individuals (Bers et al. 2018). This framework aims to promote positive development by utilizing appropriate tools such as tangible robotics [36]. For example, the TangibleK robotics program addresses another six Cs (i.e., caring, connection, contribution, competence, confidence, and character) in the PTD framework that enables young learners to work with technologies [36]. Third, Papadakis et al. [21] used a constructivist approach to design developmentally appropriate learning activities for preschoolers to learn CT skills such as sorting objects by size, shape, and color, and completing a series of actions logically. Fourth, some researchers used artifact-centric activity theory (ACAT) for children's learning of the dynamic linear unit concept [38]. The ACAT framework explains how students interact with a coding robot toy (the artifact) mediates a student's conceptualization of a dynamic linear unit within the context of a teacher-led small group activity group (the object). Fifth, some researchers suggest three strategies (i.e., questioning, modeling, and motivation/ encouragement) to enhance children's participation in the activities in order to improve their CT competencies (i.e., problem decomposition, abstraction, algorithm and procedures, pattern recognition, debugging/troubleshooting) [47]. Sixth, several researchers have designed projects to improve children's CT (e.g., [7,11,21,37,]), as shown in Table 6. For example, Relkin et al. [7] designed a project to write creative compositions about what would happen at their own Wild Rumpus Party, conduct group discussions, and engage students in programming the KIBO to perform Wild Rumpus Party activities which could promote children's CT.

Learning outcomes in the CT studies

This section discusses the learning outcomes of CT learning in kindergarten education. We categorized the learning outcomes into three domains in terms of cognitive and non-cognitive outcomes. Cognitive learning outcomes include skills and knowledge, whereas non-cognitive domains include other skills such as collaboration, communication, hand and eye movement [52].

As shown in Table 7, a number of studies showed that effective CT instructional design could enhance children's early CT and coding skill acquisition [7,17] and improved the mastery of CT and programming concepts [11,16,21,24,]. To begin with, students in the coding as another language (CAL) curriculum group perform better than students in the no-CAL group without coding to learn modularity, algorithms,

Table 3
Research Methods and Data Collection Reported in the Included Studies.

Study	Research methods	Data Collection
Bers [36]	Mixed Methods	Student's portfolios, video journals, and assessment (Knowledge)
Kazakoff et al. [39]	Quantitative	Pre-and post-tests (sequencing skills)
Bers et al. [11]	Quantitative	Assessments (debugging, correspondence, sequencing, and control flow)
Wang et al. [40]	Qualitative	Interviews, observation notes, photographs, and videotape
Elkin et al. [41]	Quantitative	Solve-It assessment
Papadakis et al. [21]	Quantitative	Assessments (basic programming concepts)
Portelance et al. [42]	Qualitative	Open-ended field notes
Cho & Lee [37]	Quantitative	Computing Survey
Pugnali et al. [18]	Mixed Methods	Assessments (Solve-IT) and observations
Sung et al. [43]	Quantitative	Paper-based pre-, post-, and delayed tests (Programming skills)
Bers et al. [17]	Mixed Methods	Observations, interviews, diary journal, and pre- workshop and post-workshop questionnaires (teacher proficiency)
García-Valcárcel-Muñoz-Repiso& Caballero-González. [44]	Quantitative	Pre- and Post-tests (Sequences, action-instruction correspondence, and debugging)
Nam et al. [45]	Quantitative	Pre-and post-tests (sequencing and problem-solving skills)
Pila et al. [24]	Mixed Methods	Child interviews; Pre-and post-assessments (gameplay assessments)
Angeli and Valanides [46]	Quantitative	Assessments (color test and spatial relations test)
Rehmat et al. [47]	Qualitative	Video analysis
Relkin et al. [15]	Quantitative	*TechCheck* Assessment
Saxena et al. [16]	Mixed Methods	Performance assessments (CT learning), lesson observations, and teacher interviews
Clarke-Midura et al. [19]	Quantitative	CT Assessments
Critten et al. [20]	Qualitative	Observations and field notes
Gerosa et al. [48]	Quantitative	CT Assessment
Monteiro et al. [49]	Qualitative	Online form and field observations
Relkin et al. [7]	Quantitative	*TechCheck* assessments
Wang et al. [50]	Quantitative	Assessment (Coding ability)
Welch et al. [38]	Qualitative	Video analysis
Yang et al. (2022)	Quantitative	Assessments *TechCheck* to assess children's CT skills Picture sequencing task Head-Toes-Knees-Shoulders test Demographic Surveys (age and gender)

and representation aspects [7]. Moreover, children aged 4–6 can successfully demonstrate CT skills, such as pattern recognition, sorting, and algorithm design [16]. Lastly, post-assessments show enhanced knowledge of Daisy commands (i.e., move, grow, and jump) and Kodable gameplay (i.e., to move a character through a maze by placing arrows in the correct order) after a 1-week program. These examples indicate that children could learn basic CT concepts (e.g., debugging, procedures) [24].

In addition to the CT knowledge and skills discussed above, some studies [38] found that children can increase their non-cognitive abilities and skills, such as collaborating and communicating with other learners in a digital environment, as well as hand and eye, and body movement during coding activities [20]. For example, a study conducted by Critten et al. [20] found that CT curriculum could improve children's communication and collaboration skills after they participate in unplugged activities (i.e., 'Bathing the baby', 'Dressing for a party' activities). This study also indicates that children younger than 34 months could gain communication and collaboration skills and interact with other participants in computational activities through

Table 4
CT Tools Used in the Included Studies.

Study	Tools	Description
Bers [36]	TangibleK	Learning robotics and programming.
Kazakoff et al. [39]	CHERP	Computer programming
Bers et al. [11]	TangibleK	Learning robotics and programming.
Wang et al. [40]	Tangiable	Computational thinking
Elkin et al. [41]	KIBO	Programming
Papadakis et al. [21]	ScratchJr	Young children learn how to write code and encode their learning (ScratchJr.org, 2015).
Portelance et al. [42]	ScratchJr	Programming blocks
Cho & Lee [37]	Aphid's toys	Children played rock-paper-scissors and allowed the ladybugs to hold fast to eat the aphids toy.
Pugnali et al. [18]	ScratchJr; KIBO	KIBO: to investigate the effect of a tangible programming interface on children's understanding of computational thinking skills ([18], p. 176). ScratchJr: programming language to children create interactive stories, colleges, and games (Strawhacker et al., 2015)
Sung et al. [43]	ScratchJr	Learning number line, counting, number ordering, addition, subtraction, and magnitude comparison
Bers et al. [17]	KIBO	KIBO robot with sensors, light output, and turntable platform ([17], p.133).
García-Valcárcel-Muñoz-Repiso& Caballero-González. [44]	TangibleK	Learning CT knowledge (e.g., sequences, action-instruction correspondence, and debugging).
Pila et al. [24]	Daisy the Dinosaur and Kodable.	Daisy the Dinosaur: teaching young children the foundation coding ("Daisy and Dinosaur", 2016); Kodable: provides a set of curriculum designed to teach young children coding ("Kodable Curriculum", 2016).
Angeli and Valanides [46]	Bee-Bot	Help children's problem-solving using Bee-Bot
Relkin et al. [15]	TACTIC-KIBO	Assess CT skills (e.g., sequencing challenges, shortest path puzzles, missing symbol series, object decomposition, obstacle mazes, symbol shape puzzles, identifying technological concepts, and symmetry problems).
Saxena et al. [16]	Bee-Bot	Programmable robot with a mat and several functions of the device (i.e., backward/ forward and rotation to the left/ right buttons).
Clarke-Midura et al. [19]	Coding robots	Children interact with coding toys.
Critten et al. [20]	Bee-Bots	Learn how to program and code to control Bee-Bots.
Gerosa et al. [48]	RoboTito	CT skills
Monteiro et al. [49]	ScratchJr, two robots	Programmed with tangible blocks and a built-in keyboard.
Relkin et al. [7]	KIBO	To teach young children programming and literacy concepts, such as algorithms, modularity, hardware/software, control structures, debugging, representation, and design process.
Wang et al. [50]	Card-based game	Coding ability
Welch et al. [38]	Cubetto	This toy uses a programming board (12 codes).
Yang et al. (2022)	Matatalab coding set	Basic Level Hands-on Coding Robot Set for children 4–9 years old [51].

Table 5
Instructional Design Reported in the Included Studies.

Study	Instructional design
Bers [36]	Session 1: Sturdy Building (the engineering design process) Session 2: What Is a Robot? (robots have special parts to follow instruction) Session 3: Hokey-Pokey: sequence of commands (the sequence or order of commands matters) Session 4: Again and Again until I Say When (loops and number parameters) Session 5: Through the Tunnel (sensors and loops) Session 6: The Robot Decides (sensors and branches)
Kazakoff et al. [39]	Engineering design process (i.e., built and programmed robotic vehicles to carry, push, and sort recyclable materials)
Bers et al. [11]	Lesson 1: The Engineering Design Process; Lesson 2: Robotics; Lesson 3: Choosing and Sequencing Programming Instructions; Lesson 4: Looping Programs (Control Flow Instructions 1); Lesson 5: Sensors; Lesson 6: Branching Programs (Control Flow Instructions 2)
Elkin et al. [41]	Session 1: Introduction to engineering and robotics; Session 2: Introduction to what is a program; Session 3: Introduction to sensing and sensors; Session 4: Sensing and introduction to repeats; Session 5: Repeats loops with numbers; Session 6: Final projects
Papadakis et al. [21]	Module 1: An introduction to ScratchJr; Module 2: Animations; Module 3 Stories; Module 4: Games; Module 5: Project time
Portelance et al. [42]	Programming blocks (Yellow Trigger blocks, blue Motion blocks, purple Looks blocks, green Sound blocks, orange Control flow blocks, and red End blocks)
Cho & Lee [37]	Lesson 1: Play the Rock-paper-scissors and let the ladybugs hold fast to eat the aphids toy; Lesson 2: Children tell their friends where to eat by saying 'right,' 'left,' and 'forward.'; Lesson 3: Write down signs of how a ladybug catches aphids and move a ladybug NXT robot as they write symbols; Lesson 4: Moves the Ladybugs NXT robot; Lesson5: Programme the NXT robot ladybugs
Pugnali et al. [18]	Lesson 1: Sequencing; Lesson 2: Repeats; Lesson 3: Conditionals; Lesson 4: Final project (all skills)
Sung et al. [43]	Number line, counting, number ordering, addition, subtraction, and magnitude comparison through ScratchJr
Bers et al. [17]	Fundamental computational thinking and coding skills For example: Sequencing (ordering a sequence of steps to perform actions), repeats (performing the same sequence a number of times), conditionals (decisions related to events or actions), and debugging (finding and fixing errors in the code).
García-Valcárcel-Muñoz-Repiso& Caballero-González. [44]	Sequences, action-instruction correspondence, and debugging
Nam et al. [45]	Activity 1: Mastering basic functionalities (begin, forward); Activity 2: Mastering basic functionalities (begin, forward, backward); Activity 3: Mastering basic functionalities (forward, backward, turn right); Activity 4: Mastering basic functionalities (forward, backward, turn right, and left); Activity 5: Returning a baby bird to the nest; Activity 6: Going to meet Bong; Activity 7: Finding a doughnut; Activity 8: Riding a bus; Activity 9: Making a sandwich; Activity 10: Taking a trip to China; Activity 11: Finding letters; Activity 12: Travelling to see dances around the world
Pila et al. [24]	The concepts of sequencing, conditions, and loops using two tablet- based apps (e.g., Daisy the Dinosaur and Kodable)
Saxena et al. [16]	LEGO pattern; Story telling; Sequencing stories; Vocabulary building songs; Direction game with cards; Tic-Tac-Toe
Relkin et al. [7]	Programming and literacy concepts 1 Sequencing/order, logical organization; 2 Breaking up larger task into smaller parts, instructions; 3 Recognizing patterns and repetition, cause and effect; 4 Symbolic representation, models; 5 Smart objects are not magical, objects are human engineered 6 Problem solving, perseverance, editing/revision 7 Identifying problems, problem solving, perseverance
Clarke-Midura et al. [19]	Interact with coding robots
Critten et al. [20]	Computational skills and ultimately, and concepts of programming and coding
Gerosa et al. [48]	Spatial concepts, Sequences, sequential movements, debugging, and sensors.
Monteiro et al. [49]	Computational thinking (plugged and unplugged), coding, and robotics
Yang et al. (2022)	Robot programming group (MatataBot) Directional command functions; Forward and backward command blocks; Turn-left and turn-right command blocks; MatataBot's parameter, drawing, and directional command function

Table 6
Learning Activities Used to Enhance Children's CT.

Study	Learning activities	Advantages
Bers et al. [11]	Project creation: Snakes that slither, recycling trucks that collect refuse, and sewing needles that travel back and forth through fabric, etc	Powerful ideas
Wang et al. [40]	Game activities: Maze Escape and Maze Creation	CT skills
Papadakis et al. [21]	Free choice project creation: Users build their projects by connecting blocks in logical sequences, allowing the characters on the screen to move, change their appearance, and/or make sounds	Programming environment
Relkin et al. [7]	Children's book (*Where the Wild Things Are* by Maurice Sendak) 1 Write a creative composition about what would happen at their own Wild Rumpus Party; 2 Group discussion; 3 Children programmed the KIBO to perform their Wild Rumpus party activities	Discussion and creative thinking

Table 7
Findings Revealing the Effects of CT in ECE.

Study	Skills and knowledge	Main findings
Kazakoff et al. [39]	Sequencing skills	In terms of sequencing skills, the post-test score was higher than the pre-test score.
Bers et al. [11]	Debugging, reconception, sequencing, and control flow	Children were interested in study robotics, programming, and CT using the TangibleK curriculum design.
Wang et al. [40]	CT skills	T-Maze can help children understand CT.
Elkin et al. [41]	Programming knowledge	The preschool children performed well on the Solve-It tasks.
Papadakis et al. [21]	Programming concepts	Fundamental programming concepts were successfully taught in the preschool classroom.
Cho & Lee [37]	Computational thinking	Several things are difficult for children to understand: programming, and distinguish between right and left.
Pugnali et al. [18]	Sequencing, loops, conditionals, debugging	The type of user interface has an effect on children's learning (i.e., positive academic and socio-emotional experiences).
Sung et al. [43]	Programming skills	The full-embody group is better than the low-embody group in programming skills (addition, pattern recognition, and fluent coding skills).
Bers et al. [17]	Basic computational thinking and coding skills	Begin teaching this new literacy as soon as possible (at 3 years old).
García-Valcárcel-Muñoz-Repiso& Caballero-González. [44]	Sequences, action-instruction correspondence, and debugging	In terms of sequences, action-instruction correspondence, and debugging dimensions, the experimental group outperforms the control group.
Nan et al. (2019)	Sequencing and problem-solving skills	There were significant differences in sequencing and problem-solving between the treatment and comparison groups when using the card-coded robotics curriculum.
Pila et al. [24]	Coding skills (concepts of sequencing, conditions, and loops)	Taught young children coding skills using digital apps were successful.
Relkin et al. [15]	Sequencing challenges, shortest path puzzles, missing symbol series, object decomposition, obstacle mazes, symbol shape puzzles, identifying technological concepts, and symmetry problems	*TechCheck* has good psychometric properties.
Saxena et al. [16]	CT learning: LEGO pattern (pattern recognition); Sequencing stories (sequencing); Direction game with Bee-Bot (algorithm design)	Students in grades K2 (ages 4 to 5) and K3 (ages 5 to 6) show their pattern recognition, sequencing, and algorithm abilities. In some complex problems, K1 students were unable to devise a correct algorithm.
Relkin et al. [7]	Computational thinking skills	Algorithms, modularity, and representation were improved in children who received CAL- KIBO.
Clarke-Midura et al. [19]	(1) Program organizer; (2) Arrow codes; (3) Grid pages, flip book; (4) Moveable agent; (5) Administration pages, with script; (6) Preset code strips (7) Scoring sheets.	The results revealed that some items (algorithmic thinking) had acceptable internal consistency reliability, as well as critical design decisions to validity evidence.
Critten et al. [20]	Communication; Collaboration	Children began to develop skills required for programming and coding, as well as computational thinking skills like collaboration, logical thinking, and debugging algorithms.
Monteiro et al. [49]	Learning activities each method (computational thinking, unplugged computational thinking, robotics, multiple approached)	As an initial framework for computational approaches in preschool: "expression and communication".
Welch et al. [38]	Children's reconception and constructed conception of a dynamic linear unit	Children used hand and arm movements (e.g., gestures) and verbal descriptions to express a constructed conception of a dynamic linear unit, and the coding toy influenced their expressions (the artifact).
Yang et al. (2022)	CT skills Sequencing ability; Self-regulation	Robot programming group outperformed sequencing ability and CT concepts than the block play group. However, the block play group outperformed sequencing ability than robot programming group.

observations. Second, Critten et al. [20] investigated children's communication and collaboration skills through observing their behavior in unplugged activities (e.g., dressing for a party, bathing a baby doll). The children were initially asked to identify the proper supplies for bathing a baby doll while their classmates were asked to point out any procedural mistakes to learn sequential structure. They were encouraged to work together to analyze flaws and their algorithms (debugging) in the correct order. The researchers "record the children's levels of communication with each other" ([20], p.10). Lastly, Welch et al. [38] conducted a case study and found that children could use hand and arm movements (e.g., gestures) and verbal descriptions to express a conception of a dynamic linear unit with robot toys [38].

Assessment methods of CT in early childhood

Regarding the assessment methods used for assessing children's CT knowledge/skills, two most frequently-used techniques in the selected studies include children's direct assessment (e.g., [7,24,25]) and observation (e.g., [17,20]). Child assessment was conducted to measure children's level of development and/or knowledge using psychological scales.

First, knowledge and skills assessments were designed and developed for evaluating children's CT skills and knowledge, as shown in Table 8. Papadakis et al. [21] used knowledge assessments to examine children's fundamental programming concepts among 120 children using ScratchJr. Fundamental programming concepts include understanding a single block, transforming individual blocks in an integrated operational program, creating a complex project, and understanding the blocks that make up a project. Bers et al. [11] used knowledge assessment to assess children's CT concepts (i.e., debugging, correspondence, sequencing, control flow). There are four steps for the debugging assessment, such as

Table 8
CT Concepts or Skills Assessed.

Study	Assessment methods
Bers [36]	Content creation and creativity
Bers et al. [11]	CT skills: Debugging, correspondence, sequencing, and control flow.
Papadakis et al. [21]	Knowledge assessments to examine fundamental programming concepts (e.g., sequences)
Pugnali et al. [18]	CT skills: Sequencing, Loops, Conditionals, and Debugging.
Nam et al. [45]	Sequencing and problem-solving skills
Pila et al. [24]	Pre- and post- gameplay assessments Four assessments: two in sequencing, one in conditional, and one in loops.
Angeli and Valanides [46]	Children's CT: Problem solving tasks (e.g., sequences of Bee-Bot's movements expressed in directional language, such as, MOVE FORWARD, TURN LEFT, MOVE FORWARD, and TURN RIGHT
Relkin et al. [15]	CT skills: Sequencing challenges, shortest path puzzles, missing symbol series, ob- ject decomposition, obstacle mazes, symbol shape puzzles, identifying technological concepts, and symmetry problems.
Saxena et al. [16]	CT knowledge: LEGO pattern (pattern recognition), Sequencing stories (sequencing), and Direction game with Bee-Bot (algorithm design).
Clarke-Midura et al. [19]	CT assessment: Program organizer, arrow codes, grid page, flip book, moveable agent, administration page with script, present code strips, and scoring sheets.
Gerosa et al. [48]	CT skills
Relkin et al. [7]	*TechCheck* assessment Algorithms, modularity, control structures, representation, hardware/software, and debugging.
Wang et al. [50]	Coding ability: Variable, Control, Modularity, and Algorithm
Yang et al. (2022)	CT concepts: Algorithms, modularity, control structures, representation, hardware/software, and debugging.

debugging a problem, debugging process, a hypothesis and cause of the problem, and solving the problem. In the assessment, children need to identify the correct programming instruction for each line for the robot Hokey-Pokey to dance [11]. Pila et al. [24] used pre- and post-knowledge assessments (i.e., familiarity with coding apps, knowledge of Daisy commands, ability to play Kodable, and understanding of coding knowledge) to assess children's CT knowledge.

As aforementioned, Saxena et al. [16] examined preschoolers' CT skills with task-based assessments. Some researchers used stories method to assess children's sequencing skills [45]. Nam et al. [45] based on Baron-Cohen et al. [53]'s research in which five types of stories were included: "Mechanical 1 (objects interacting causally with each other), Mechanical 2 (people and objects acting causally on each other), Behavioral 1 (a single person acting in everyday routines not requiring attribution of mental states), Behavioral 2 (people acting in social routines, involving more than one person, but not requiring attribution of mental states), and Intentional (people acting in everyday activities requiring attribution of mental states)" ([45], p.393). Another study conducted by Nam et al. [45] modified from Ward's [54] instrument to assess children's problem-solving skills (e.g., categorization, patterns, numbering, measuring, diagramming, statistics). Results show that the students have improvement in CT abilities in which the post-test score was higher than the pre-test score in sequencing and problem-solving skills when using the card-coded robotics curriculum [45]. Furthermore, Relkin et al. [7] used *TechCheck* assessment to assess children's CT skills. *TechCheck* consists of 15 multiple-choice questions and six assessments (i.e., algorithms, modularity, control structures, representation, hardware/software, and debugging design process). Table 8 shows the key CT concepts or skills assessed in different studies.

The second commonly used method is observation. Through observation, researchers could evaluate children's CT programming knowledge [16,17] (e.g., pattern recognition, sequencing, and algorithm design), and their learning behavior such as communication and collaboration skills [20,49]. For example, Monteiro et al. [49] recorded how children interact and communicate with tangible robots using the PTD framework. Several key categories (i.e., curricular content, learning objectives, intervention methodology, children's responses) as well as positive and negative behaviors were mapped to the proposed PTD framework using thematic analysis. Positive behaviors include children's involvement and motivation, skill development, and methodological features. Negative aspects include classroom management, learning progress, and children's participation. Future studies could refine the learning programs to meet students' needs such as difficulty in understanding the task and its goals (e.g., itinerary representation of a programmed route of a robot), and barriers to social development (e.g., difficulties in promoting cooperation in coding activities) ([49], p.11). Bers et al. [17] observed the classroom dynamics with KIBO. Six aspects observed included: "1) curriculum sessions (number and duration of each session), 2) student groups (size, organization and composition of the group), 3) tutoring (rotation among groups, number of students per teacher/tutor), 4) materials (types of crafts and recycled materials used, organization of robotic kits, availability, accessibility of materials in the classroom), 5) organization (allocation of the robots in the classroom: one per group, stations, corners), and 6) didactic strategies (how the project was introduced, the role of teachers and students" ([17], p.137). Saxena et al. [16] conducted classroom observations to examine children's performance and interactions, as well as teachers' instructional practices during CT activities. Child engagement observed cover the aspects shown in Table 9.

Discussion

This review analyzed a total of 26 studies conducted in different countries from 2010 to 2022 regarding CT tools, knowledge, activities, impacts, and challenges and opportunities for learning and teaching in the crucial field of ECE. We found that most of the studies were conducted in the United States. Some important points were summarised as follows. First, most studies used KIBO as the platform in CT in early childhood research. Second, several studies used the PTD Framework as the theoretical framework [6,11,17,18,36,49]. Third, we have summarized and found that a number of studies showed that the CT studies were effective in terms of enhancing children's early CT skills, coding skills, communication and collaboration skills, CT, and programming concepts. Fourth, most studies were found to use a quantitative research method. Two frequently-used assessment techniques were child assessment and observation.

Opportunities of teaching and learning CT in ECE settings

Benefits of learning CT were categorized by previous research in terms of cognitive and non-cognitive abilities. First, with age-

Table 9
Observations Involved to Examine Child Engagement in the CT Activities.

Study	Observations
Pugnali et al. [18]	Children's positive technology development (e.g., positive conduct and community building) during activities
Bers et al. [17]	Assessed the children's performance with KIBO
Saxena et al. [16]	Lesson observations, children's verbalization and actions on computational thinking materials reflect children's thought processes.
Critten et al. [20]	Communication and collaboration abilities in CT activities
Monteiro et al. [49]	Children interaction with technologies (ScratchJr, two robots)

appropriate tools and curriculum, teachers could reduce the cognitive overload and engage students in learning basic CT skills such as sequence, debugging, and action-instruction correspondence [44], loops, conditionals [18], shortest path puzzles, missing symbol series, object decomposition, obstacle mazes, symbol shape puzzles and symmetry problems [15], pattern recognition [16], and algorithm design [38]. Children who learn computational skills and computer science concepts could gain problem-solving strategies that are considered to be a way of human thinking to facilitate their learning and living [13].

On top of learning CT and computer science concepts, students could also gain a set of non-cognitive skills such as critical thinking, collaboration and communication skills [17,43]. Through making and interacting with artifacts, students could foster their creativity and curiosity (Alves-Oliveira et al., 2020), enhance body-material interaction and hand-eye coordination (Casellato et al., 2017). Further, students could also interact with the CT-enabled kits such as robotic devices and block play which encourage them to socialize and communicate with each other (Han et al., 2005) and improve their self-regulation [25,26]. These skills are fundamental for children to develop positive learning mindsets and attitudes of using Information and Communication Technologies (ICT) at a young age that facilitate them for their future studies. As such, CT provides great learning opportunities for students to develop their cognition and social skills that empower them to perform well and achieve goals in their future. After a decade of CT implementation, the review has emerged to document theoretical and empirical evidence of how to develop a CT curriculum ([27]; Weintrop et al., 2021), and assess CT understandings (e.g., Cutumisu et al., 2019).

Challenges of teaching and learning CT in ECE settings

Although CT learning provides rich opportunities to explore our digitalized world, learning and teaching CT could be challenging in ECE settings. First, one study has identified that children do not gain rich CT concepts (e.g., iterations, conditionals) at a young age (Bers et al., 2020). Challenges could be found when CT has been conceptualized differently in different age ranges, and teachers needed to choose age-appropriate concepts when teaching young learners CT. With technological advancements, many smart tools and devices are designed with a low floor (the ability to create simple rules without prior programming concepts), but also with high ceilings (the ability for children to build their solutions) to engage children in programming and CT learning (Relkin et al., 2019).

Furthermore, based on our systematic analysis (Tables 8 and 9), there is a lack of valid and reliable CT assessments for young children, since most of the studies did not report the scientific evidence of the psychometric properties of their instruments used. This could be due to a lack of consensus on CT frameworks and definitions [19]. Several assessments were found to measure different CT-related skills and abilities; for example, Cittá et al. (2019) designed a paper-pencil test that assessed the students' ability to write and interpret an algorithm using a chessboard. Bers et al. [17] measured children's CT through a set of robot-based challenges. Protocols and checklists have been developed to assess students' progressive cognitive abilities through interview-based and paper-based assessments. These studies highlight how different CT assessments for young children measure different CT skills and practices. There could be other CT-related abilities that are rarely paid attention to but are also important such as spatial reasoning and self-regulation.

Third, although young children could play with the programmable toys, they found it challenging when they socialized with other classmates to solve problems together and negotiate with other peers (Yelland, 2011). Also, robotics and programming in early childhood may cause gender bias. It is found that girls tend to be demotivated by these types of boy-dominated toys (Sullivan et al., 2019). This may lead to digital inequity in early childhood education and society.

Based on our literature review, the implementation of CT is still continuing to develop across countries. Teachers and students meet various opportunities and challenges of learning CT in early childhood education. In order to understand the development of CT education, we analyzed the definitions and taxonomies (the thinking steps) of CT. Then, we examined the importance of teaching CT at the kindergarten level. Although CT in early childhood education provides rich opportunities to enable kindergarteners to explore the digitized world, the research identified that educators meet various challenges, including designing age-appropriate materials for young learners (Bers et al., 2020), and limited reliable CT assessments to examine their understanding and learning performance [19]. These challenges bring opportunities to improve the CT instructional design and assessment methods and address students' learning needs in CT education. Findings of this systematic review informs future endeavors in *theorizing* a digital learning framework that can integrate CT into early childhood education.

Recommendations for future research

This review identifies a number of scarce but successful studies on CT curriculum that promotes student learning around the world. To begin with, this study provides important recommendations and guidelines for future CT researchers and educators to create useful and meaningful learning designs and tools to foster children's CT understandings and mindsets. Second, we found no studies comparing different CT tools. We hope that future researchers will be able to compare the strengths and weaknesses of various CT tools so that researchers and educators can easily select a suitable CT tool. Moreover, we also found that no studies investigate whether socioeconomic status (SES) and gender have an impact on children's CT. However, many researchers already confirmed that SES and gender influences children's learning of STEM (e.g., [55–57]). As a result, we suggest future researchers to fill this knowledge gap. Furthermore, we found that most studies were conducted in developed countries (i.e., the USA, Greece, Spain, Portugal, and the United Kingdom). Therefore, future research needs to investigate how CT curricula in early childhood can be applied in developing countries.

Recommendations for ECE practitioners

The selected studies suggested useful recommendations for CT curriculum development in early childhood education. First, teachers who teach CT subjects should receive extensive teacher training (e.g., workshops, seminars), and collaborate closely with CT experts in the education field to develop reasonable assessments and tools to evaluate the effectiveness of the CT curriculum in early childhood education in the meantime. Second, we suggest that teachers should design meaningful and developmentally appropriate projects to enhance children's CT and higher order thinking. For example, Relkin et al. [7] designed a project to encourage students to write creative compositions about what would happen at their own Wild Rumpus Party, which could promote children's creative thinking. Third, inspired from artifact-centric activity theory (ACAT), we recommend that educators should use meaningful artifacts such as KIBO, Bee-Bots and Matatalab to scaffold CT concepts (e.g., sequencing, modularity, representations) [7,20,25,26]. Further, some strategies were also considered. For example, Rehmat et al. [47] suggested the use of playful experiences to promote students' motivation and encouragement. Questioning and modeling techniques could help students understand the robot's movements and its related CT competencies, such as problem decomposition, abstraction, algorithm and procedures, pattern recognition, and debugging/troubleshooting.

Limitations of this review

There are several limitations in this study. The first limitation is we only looked at existing papers written in English. The second limitation is there is a small amount of literature on the CT curriculum for kindergarten classrooms, and we only selected journal articles in this field, excluding conference papers, editorials, etc. We found that most studies were conducted in developed countries, such as the USA, Greece, Spain, Portugal, and the United Kingdom. We recommend that future research needs to investigate how CT curricula in early childhood can be applied in developing countries. Next, we hope that future researchers will be able to compare the strengths and weaknesses of various CT tools so that researchers and educators can easily select a suitable CT tool. Last but not least, we suggest future researchers can fill the gap regarding how SES and gender influences children's learning of CT.

Conclusion

This review contributes to the mapping of learning content in existing CT curricula, CT tools, learning outcomes, and assessment methods in ECE settings, extending the line of research on CT in K-12 settings. This paper also identifies the challenges and opportunities of CT in ECE for researchers and practitioners as a reference. The findings of this study can inform future research in terms of advancing CT tools, pedagogical methods, research methods, and assessment for early CT education and provide researchers and educators with a guide for the design, implementation, and evaluation of age-appropriate CT curricula for children.

Author statement

Statements on Ethics & Conflicts: This article does not contain any studies with human participants or animals performed by author. On behalf of author, the corresponding author states that there is no conflict of interest.

**References marked with an asterisk indicate articles included in this systematic review.*

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.caeo.2023.100122.

References

[1] Grover S, Pea R. Computational thinking in K–12: a review of the state of the field. Educ Res 2013;42(1):38–43.
[2] Kong SC. A framework of curriculum design for computational thinking development in K-12 education. J Comput Educ 2016;3(4):377–94.
[3] Papert S. Mindstorms: children, computers, and powerful ideas. New York: Basic Books; 1980.
[4] Wing JM. Computational thinking. Commun ACM 2006;49(3):33–5.
[5] Wing J. Research notebook: computational thinking—what and why. Link Mag 2011;6:20–3.
[6] Bers MU. Coding, robotics and socio-emotional learning: developing a palette of virtues. *Pixel-Bit.* Revista de Medios y Educación 2021;62:309–22.
[7] Relkin E, de Ruiter LE, Bers MU. Learning to code and the acquisition of computational thinking by young children. Comput Educ 2021;169:104222.
[8] Su J, Yang W. Artificial intelligence in early childhood education: A scoping review. Computers and Education: Artificial Intelligence 2022;100049.
[9] Voogt J, Fisser P, Good J, Mishra P, Yadav A. Computational thinking in compulsory education: towards an agenda for research and practice. Educ Inf Technol 2015;20(4):715–28.
[10] Hsu TC, Chang SC, Hung YT. How to learn and how to teach computational thinking: suggestions based on a review of the literature. Comput Educ 2018;126: 296–310.
[11] Bers MU, Flannery L, Kazakoff ER, Sullivan A. Computational thinking and tinkering: exploration of an early childhood robotics curriculum. Comput Educ 2014;72:145–57.
[12] Zhang L, Nouri J. A systematic review of learning computational thinking through Scratch in K-9. Comput Educ 2019;141:103607.
[13] Wing JM. Computational thinking and thinking about computing. Philos Trans R Soc, A 2008;366(1881):3717–25.
[14] Brennan K, Resnick M. New frameworks for studying and assessing the development of computational thinking. In: Proceedings of the 2012 annual meeting of the American educational research association1; 2012. p. 25.
[15] Relkin E, de Ruiter L, Bers MU. TechCheck: development and validation of an unplugged assessment of computational thinking in early childhood education. J Sci Educ Technol 2020;29(4):482–98.
[16] 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-Pacific Educ Res 2020;29(1):55–66.
[17] Bers MU, González-González C, Armas-Torres MB. Coding as a playground: promoting positive learning experiences in childhood classrooms. Comput Educ 2019;138:130–45.
[18] Pugnali A, Sullivan A, Bers MU. The impact of user interface on young children's computational thinking. J Inf Technol Educ. Innovat Practice 2017;16:171.
[19] Clarke-Midura J, Silvis D, Shumway JF, Lee VR, Kozlowski JS. Developing a kindergarten computational thinking assessment using evidence-centered design: the case of algorithmic thinking. PeerJ Comput Sci 2021;31(2):117–40.
[20] Critten V, Hagon H, Messer D. Can pre-school children learn programming and coding through guided play activities? A case study in computational thinking. Early Childhood Educ J 2021:1–13.
[21] Papadakis S, Kalogiannakis M, Zaranis N. Developing fundamental programming concepts and computational thinking with Scratch Jr in preschool education: a case study. Int J Mobile Learn Org 2016;10(3):187–202.
[22] Tang X, Yin Y, Lin Q, Hadad R, Zhai X. Assessing computational thinking: a systematic review of empirical studies. Comput Educ 2020;148:103798.
[23] Weintrop D, Beheshti E, Horn MS, Orton K, Trouille L, Jona K, Wilensky U. Interactive assessment tools for computational thinking in high school STEM classrooms. In: International conference on intelligent Technologies for interactive entertainment. Springer; 2014. p. 22–5.
[24] Pila S, Aladé F, Sheehan KJ, Lauricella AR, Wartella EA. Learning to code via tablet applications: an evaluation of Daisy the Dinosaur and Kodable as learning tools for young children. Comput Educ 2019;128:52–62.
[25] Yang W, Ng DTK, Gao H. Robot programming versus block play in early childhood education: effects on computational thinking, sequencing ability, and self-regulation. Br J Educ Technol 2022;00:1–25.
[26] Yang W, Luo H, Su J. Towards inclusiveness and sustainability of robot programming in early childhood: child engagement, learning outcomes and teacher perception. Br J Educ Technol 2022;00:1–25.
[27] Lye SY, Koh JHL. Review on teaching and learning of computational thinking through programming: what is next for K-12? Comput Human Behav 2014;41: 51–61.
[28] Shute VJ, Sun C, Asbell-Clarke J. Demystifying computational thinking. Educ Res Rev 2017;22:142–58.
[29] Lockwood, J., & Mooney, A. (2018). Developing a computational thinking test using Bebras problems.
[30] Bakala E, Gerosa A, Hourcade JP, Tejera G. Preschool children, robots, and computational thinking: a systematic review. Int J Child Comput Interact 2021;29: 100337.
[31] Bers MU. Coding and computational thinking in early childhood: the impact of ScratchJr in Europe. Eur J STEM Educ 2018;3(3):8.
[32] Kalogiannakis M, Papadakis S. Pre-service kindergarten teachers acceptance of "ScratchJr" as a tool for learning and teaching computational thinking and Science education. In: Proceedings of the 12th Conference of the European Science Education Research Association (ESERA), Research, practice and collaboration in science education. Dublin: Dublin City University and the University of Limerick; 2017. p. 21–5.
[33] Bati K. A systematic literature review regarding computational thinking and programming in early childhood education. Educ Inf Technol 2021:1–24.
[34] Moher D, Altman DG, Liberati A, Tetzlaff J. PRISMA statement. Epidemiology 2011;22(1):128.
[35] Fraenkel JR, Wallen NE, Hyun HH. How to design and evaluate research in education. 9th ed. New York, NY: McGraw-Hill Education; 2015.
[36] Bers MU. The TangibleK robotics program: applied computational thinking for young children. Early Childhood Res Practice 2010;12(2):n2.
[37] Cho Y, Lee Y. Possibility of improving computational thinking through activity based learning strategy for young children. J Theor Appl Inf Technol 2017;95(18): 4385–93.
[38] Welch LE, Shumway JF, Clarke-Midura J, Lee VR. Exploring measurement through coding: children's conceptions of a dynamic linear unit with robot coding toys. Educ Sci 2022;12(2):143.
[39] Kazakoff ER, Sullivan A, Bers MU. The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. Early Childhood Educ J 2013;41(4):245–55.
[40] Wang D, Wang T, Liu Z. A tangible programming tool for children to cultivate computational thinking. Sci World J 2014;2014.
[41] Elkin M, Sullivan A, Bers MU. Programming with the KIBO robotics kit in preschool classrooms. Comput Schools 2016;33(3):169–86.
[42] Portelance DJ, Strawhacker AL, Bers MU. Constructing the ScratchJr programming language in the early childhood classroom. Int J Technol Des Educ 2016;26(4): 489–504.
[43] Sung W, Ahn J, Black JB. Introducing computational thinking to young learners: practicing computational perspectives through embodiment in mathematics education. Technol Knowl Learn 2017;22(3):443–63.
[44] García-Valcárcel-Muñoz-Repiso A, Caballero-González YA. Robotics to develop computational thinking in early childhood education. Comunicar. Media Educ Res J 2019;27(1).
[45] Nam KW, Kim HJ, Lee S. Connecting plans to action: the effects of a card-coded robotics curriculum and activities on Korean kindergartners. Asia-Pacific Educ Res 2019;28(5):387–97.
[46] Angeli C, Valanides N. Developing young children's computational thinking with educational robotics: an interaction effect between gender and scaffolding strategy. Comput Human Behav 2020;105:105954.

[47] Rehmat AP, Ehsan H, Cardella ME. Instructional strategies to promote computational thinking for young learners. J Digit Learn Teacher Educ 2020;36(1): 46–62.
[48] Gerosa A, Koleszar V, Tejera G, Gómez-Sena L, Carboni A. Cognitive abilities and computational thinking at age 5: evidence for associations to sequencing and symbolic number comparison. Comput Educ 2021;2:100043.
[49] Monteiro AF, Miranda-Pinto M, Osório AJ. Coding as literacy in preschool: a case study. Educ Sci 2021;11(5):198.
[50] Wang L, Geng F, Hao X, Shi D, Wang T, Li Y. Measuring coding ability in young children: relations to computational thinking, creative thinking, and working memory. Curr Psychol 2021:1–12.
[51] Matatalab. (2020). Coding Set. https://matatalab.com/en/coding-set.
[52] National Research Council. Education for life and work: developing transferable knowledge and skills in the 21st century. National Academies Press; 2012.
[53] Baron-Cohen S, Leslie AM, Frith U. Mechanical, behavioural and Intentional understanding of picture stories in autistic children. Brit J Dev Psychol 1986;4(2): 113–25.
[54] Ward CSD. *Developmental* versus *academic mathematics education: effects on problem-solving performance and attitudes toward mathematics in kindergarten children*. Peabody College for Teachers of Vanderbilt University; 1993.
[55] Ho MT, La VP, Nguyen MH, Pham TH, Vuong TT, Vuong HM, Vuong QH. An analytical view on STEM education and outcomes: examples of the social gap and gender disparity in Vietnam. Child Youth Serv Rev 2020;119:105650.
[56] Perez-Felkner L, Felkner JS, Nix S, Magalhães M. The puzzling relationship between international development and gender equity: the case of STEM postsecondary education in Cambodia. Int J Educ Dev 2020;72:102102.
[57] Su J, Yang W, Zhong Y. Influences of gender and socioeconomic status on Children's use of robotics in early childhood education: a systematic review. Early Educ Dev 2022:1–17.