Education and Information Technologies
https://doi.org/10.1007/s10639-022-11445-2

A meta-analysis of the effectiveness of programming teaching in promoting K-12 students' computational thinking

Enwei Xu[1] · Wei Wang[1] · Qingxia Wang[1]

Received: 20 June 2022 / Accepted: 30 October 2022

Abstract
Computational thinking is considered to be an important competence in the intelligent era, and the incorporation of computational thinking as an integral part of school education beginning in childhood has been proposed. However, the ways in which computational thinking can be taught more effectively the context of in K-12 programming teaching remain unclear. This paper reports the results of a meta-analysis of 28 empirical studies on K-12 programming teaching that were published in international education journals in the 21st century to determine which teaching methods and programming tools are most effective in promoting the computational thinking of K-12 students. The results show that (1) programming teaching can promote the improvement of K-12 students' computational thinking (ES = 0.72, z = 9.9, $P < 0.01$), with an overall effect at the upper-middle level (95% CI[0.60,0.83]); (2) scaffolding programming (ES = 1.84, z = 11.9, $P < 0.01$) and problem-based programming (ES = 1.14, z = 5.57, $P < 0.01$) are the most effective teaching methods and can significantly promote the development of K-12 students' computational thinking ($chi^2 = 40.58$, $P < 0.01$); (3) since differences in the effect of programming tools between groups are not significant ($Chi^2 = 6.47$, $P = 0.09$), it is impossible to determine which programming tools are most effective; and (4) intervention duration (ES = 0.72, z = 11.9, $P < 0.05$, 95% CI[0.60, 0.83]) and learning scaffold (ES = 0.83, z = 6.27, $P < 0.05$, 95% CI[0.57, 1.09]) are both key moderating variables that affect the improvement of computational thinking. Based on these results, suggestions are provided for future research and practice.

Keywords Computational thinking · Programming tool · Teaching method · Effectiveness · Meta-analysis

✉ Enwei Xu
244924608@qq.com

Extended author information available on the last page of the article

1 Introduction

As an important problem-solving ability in the intelligent era, computational thinking (CT) refers to the basic level of literacy that citizens should possess in the 21st century (Wing, 2006, 2010). The origin of CT lies in a thinking process focused on problem solving in real situations (Shute et al., 2017), and this concept includes the five core elements of abstraction, decomposition, algorithmic thinking, generalization and evaluation (Selby & Woollard, 2013). Moreover, the core focus of programming teaching (PT) is also the solution of problems, and this notion includes the four key elements of problem decomposition, the application of algorithms, abstraction and automation (Shute et al., 2017). Therefore, PT is considered to be the best approach to the cultivation of CT due to its unique potential to cultivate logical thinking and innovative ability (Zhang & Ji, 2018; Scherer 2016). Accordingly, some countries have incorporated CT into their technical curricula to cultivate students' CT in the form of subject courses (e.g., South Korea, Kim 2020; China, Ministry of Education of the People's Republic of China, 2020). The result of this trend is that a large number of programming tools (such as graphical programming, text programming, robot programming and unplugged programming) and teaching methods (such as collaborative programming, game programming, project programming and scaffold programming) have emerged in the context of PT practice in K-12 education in recent years (Grover & Pea, 2013; Lye & Koh, 2014).

Although several overview studies have made efforts to systematically review and meta-analyze the empirical literature pertaining to CT from a variety of perspectives, little attention has been given to the question of how to choose programming tools and corresponding teaching methods to teach CT. Therefore, the question of which programming tools and teaching methods are the most effective pathway for cultivating and improving the CT of K-12 students remains unanswered (Florez et al., 2017; Scherer, 2016), which entails that many K-12 teachers are unable to effectively implement CT teaching in the context of PT (Saad & Zainudin, 2022). For example, Lye and Koh (2014) used the framework proposed for Scratch by Brennan and Resnick (2012) to investigate 27 empirical studies related to teaching and learning CT via PT in a K-12 environment. The findings of those authors indicated which intervention methods were being used to cultivate CT and summarized the more popular intervention strategies and programming tools by reference to descriptive statistical data. However, this study did not indicate whether there were significant differences among these intervention strategies and programming tools or determine which kind of teaching intervention could be more effective in promoting the development of CT; accordingly, no guidelines for teaching CT more effectively could be developed without a reference to specific effect sizes. In addition, Hsu et al. (2018) reviewed certain studies published from 2006 to 2017 in the context of CT learning and teaching in the K-12 environment. These authors found that CT is used mainly in the context of programming and computer science activities, and most relevant studies have focused on project-based learning, problem-based learning, cooperative learning and game-based learning in the context of CT activities. However, the effect sizes of teaching strategies and programming tools have not yet been reported, so the question of how to teach CT more effectively in the context of PT remains unanswered. Moreover, Sun et al. (2021) reported the results of a meta-analysis of 22 empirical studies with the aim of determining the effectiveness of the use of educational games to improve students' CT, and Lai and Wong (2022) reported the results of a meta-analysis of 33 empirical studies with the aim of determining the effectiveness of collaborative or individual problem-solving for the improvement of students' CT. However, these analyses focused on the single teaching strategy and did not make comparisons with other teaching strategies or programming tools. On the other hand, previous empirical studies have reported inconsistent conclusions and even produced contradictory results. For example, some studies have shown that the teaching effect of graphical programming tools is greater than that of text programming (Vahldicka et al., 2020), but other studies have shown that graphical programming limits students' potential for logical thinking (Scherer, 2016) and that students' abilities to solve practical problems are not improved by this approach (Kalelioglu & Gulbahar, 2014). Some studies have also found that collaborative programming has a lower-middle level of effect on individual and group learning performance (Denny et al., 2015); however, other studies have shown that collaborative programming has an upper-middle level of effect on the development of students' CT (Yuksel, 2015) and that the use of collaboration in programming can improve students' programming skills (Denner et al., 2014) and programming confidence (Beck & Chizhik, 2013).

Angeli and Giannakos (2020) and Tikva and Tambouris (2021) both noted that teaching strategies and technical tools are important pathways for the promotion of students' CT in the context of PT and that the effectiveness of teaching methods and technical tools with respect to enhancing and promoting the development of CT should be studied and supported by evidence to determine whether and to what extent these factors lead to an increase or decrease in CT levels. Therefore, a comprehensive and reliable overall review is required to identify how to teach CT in the context of PT most effectively. Meta-analysis is a research method that is used to analyze empirical data drawn from multiple independent studies on the same research topic, and this method uses the average effect size of multiple empirical research data to characterize the effectiveness of its impact with the aim of reducing the uncertainty caused by single studies and thus obtaining more reliable conclusions (Lipsey & Wilson, 2000). In addition, Chen et al. (2018) proposed three frameworks for studying the effectiveness of learning based on the evidence produced by CSCL research, namely, the effectiveness of the teaching intervention itself, the effectiveness of the teaching environment or tools used and the effectiveness of the teaching method used.

To contribute to both research and practice, this study adopted a meta-analytic method and conducted a meta-analysis of the effectiveness of PT with respect to the improvement of K-12 students' CT based on the conditional framework of learning effectiveness proposed by Chen et al. (2018). The aim of this meta-analysis was to answer the following research questions:

1. What is the overall effect of PT on the improvement of K-12 students' CT? (i.e., the effectiveness of teaching intervention itself)
2. What are the effects of different teaching methods on K-12 students' CT in the context of PT, and are there significant differences across methods? (i.e., the effectiveness of teaching method)
3. What are the effects of different programming tools on K-12 students' CT in the context of PT, and are there significant differences across tools? (i.e., the effectiveness of programming tool)
4. If the effects of different teaching methods and different programming tools are heterogeneous, how do different moderating variables affect the differences among the research conclusions?

2 Literature review

2.1 The operational definitions and outcome dimensions of CT

CT is considered to be an important competence in the intelligent era, and it has been suggested that CT should be made an integral part of school education beginning in childhood (Wing, 2010). At present, operational definitions of CT have not reached any consistent conclusions (Lu et al., 2022), but CT can be divided approximately into two perspectives by combing and analyzing the research results of different studies. One such perspective emphasizes a definition of CT that is related to the thinking mode used by computers based on the perspective of computer science. For example, Brennan and Resnick (2012) proposed a three-dimensional framework for CT based on a scratch programming environment. The first aspect of this framework includes CT concepts, which refer to the essential ideas that are utilized to describe programming knowledge (such as sequences, loops, and events). The second such aspect focuses on CT practices, which are the essential ideas that are utilized to describe programming processes (such as iteration, testing and model construction). The third aspect emphasizes CT-related perspectives, which denote the essential ideas that are utilized to describe the perceptions associated with programming interactions (such as expression and questioning). Another perspectives focus on the thinking process associated with problem solving from the perspective of ability development. For example, Chinese information technology curriculum standards stipulate that "Computing thinking is a thinking activity in the process of problem-solving, including abstracting, decomposition and modeling of problems with the help of tool, logical organization and processing of data, forming the migration of problem solving process" (Ministry of Education of the People's Republic of China, 2022, p. 6).

In the field of teaching practice, due to these different understandings of the operational definition of CT, paper or picture tests, open question tests, scratch project online evaluations, scale evaluations, questionnaire surveys and interviews are all used to evaluate and explain CT as well as its development. Roman et al. (2019) summarized these approaches into seven categories: diagnostic test, summary tool, feedback tool, coding process analysis, ability migration test, perceived attitude scale and factor evaluation scale. Tang et al. (2020) analyzed the evaluation methods used for CT in 96 journal articles systematically in terms of four aspects: educational background, evaluation structure, evaluation type, reliability and validity. All these authors found that traditional testing and performance evaluation are frequently used to evaluate CT skills, programming knowledge and attitudes and perceptions of CT. Lai and Wong (2022) classified all CT outcomes extracted from 33 empirical studies on PT into cognitive, affective and social competencies. Based on the conclusions of the studies mentioned above and synthesizing the two perspectives on the operational definition of CT, this study encoded the outcomes of CT into the three categories of academic achievement (focusing on CT concepts), thinking skills (focusing on CT practices) and social emotions (focusing on CT perspectives).

2.2 Programming tools and teaching methods in K-12 education

In the field of CT cultivation in K-12 education, a large number of programming tools and teaching methods are being incorporated into the classroom. Tikva and Tambouris (2021) noted that the most popular programming tools include web-based simulation creation tools (such as Agent Sheets and Agent Cubes), tangible media tools based on educational robots (such as Arduino and GoGo boards), graphical block-based programming tools (such as Scratch, Alice, APP inventor, and Game Maker) and text code programming tools (such as Python, C or C++)). Repenning et al. (2010) proposed that all programming tools that are incorporated into classroom teaching should exhibit the characteristics of a low threshold, a high ceiling, and support and migration applications to help programming beginners avoid tedious problems with programming syntax and create applications or solve practical problems quickly.

Simultaneously, the use of programming tools also requires corresponding teaching methods to teach students the process of learning programming to allow them solve problems through the use of programming tools with the aim of improving students' understanding of programming knowledge and encouraging the development of CT. For example, based on the Alice graphical block programming tool, Denner et al. (2012) used games and collaborative programming methods to effectively help middle school students understand problem modeling and algorithmic thinking and proposed that a combination of multiple teaching methods may be the most effective method in the context of PT. Based on the LEGO robot programming tool, Lee et al. (2014) used game design, collaborative programming and project-based programming methods to improve students' understanding of CT concepts and CT practice; in addition, these authors highlighted the importance of collaboration and teacher support in this context.

In conclusion, several programming tools and teaching methods have provided convenient conditions for PT in the context of K-12 education. The purpose of this meta-analysis is to investigate the effectiveness of PT with respect to the promotion of K-12 students' CT, especially in terms of the effects of teaching methods and programming tools on the promotion of CT development, and to provide evidence concerning whether and to what extent these factors contribute to increasing or decreasing levels of CT.

3 Method

This research employed the meta-analysis process proposed by Cooper (2010) as its design framework, selected relevant empirical literature published in international educational journals in the 21st century as the object of analysis and thus conducted a meta-analysis using RevMan 5.4 software.

3.1 Literature search strategy

The process of obtaining data for the meta-analysis was divided into three stages (Fig. 1). First, the advanced search functions in the Web of Science Core, Eric and Science Direct databases were used to search the literature published between January 2000 and October 2021. The search keywords were divided into the following two groups:

(1) programming-related keywords (such as "programming\coding\computer programming\pair programming");
(2) CT-related keywords (such as "computational thinking\CT").

Using different combinations of these keywords as search conditions, a total of 1475 studies were obtained. After importing these studies into Endnote and eliminating duplicate studies, 767 studies were obtained. Second, the titles and abstracts of the collected literature were read to select studies that pertained to the research theme according to the inclusion criteria used for the meta-analysis, following which citation backtracking was conducted based on the selected literature; a total of 72 studies were thus obtained. Third, we carefully read the full text of the included studies, eliminating studies that did not meet the inclusion criteria for the meta-analysis (such as no clear conclusion or incomplete data); ultimately, 28 studies were obtained.

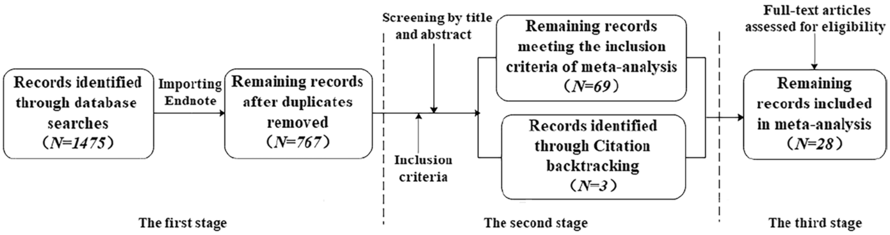

Fig. 1 Flow chart of literature acquisition for the meta-analysis

3.2 Inclusion criteria

The inclusion criteria used for the literature were as follows:

(1) The theme must be the impact of PT on CT, and the research object must be K-12 students.
(2) The research method must feature experimental or quasi-experimental designs using pre-post test data or comparative experimental data.
(3) The research results must report specific indicators that can be used to measure the effect of CT, such as sample size, mean, standard deviation or variance.

According to the inclusion criteria, two researchers read the full text of 72 studies and reached consensus via discussion and negotiation to resolve any emerging differences; this process led to a consensus rate of approximately 94.6%. Ultimately, 28 articles were retained for a total experimental sample size of 4154 with 98 effect quantities (because some studies included multiple effect quantities).

3.3 Coding design

Meta-analysis requires the extraction of key information from the literature, a process of encoding the characteristics of that information and the transformation of descriptive data into quantitative data. Based on the variables included in the experimental designs of 28 studies, this study constructed a literature eigenvalue coding table that contained 16 coding fields (Table 1).

Descriptive information mainly includes basic information regarding the literature, including the five fields of article serial number, article title, author's name, publication year and publication journal.

Variable information mainly focuses on three variables related to experimental design across 28 studies, including independent variables (intervention strategy), dependent variables (CT) and moderating variables (learning stage, intervention duration, learning scaffold, programming tool and evaluation tool). Intervention strategy is used as the independent variable, which was summarized and integrated into the programming tool and teaching method based on the conditional framework of learning effectiveness proposed by Chen et al. (2018). During the coding process, it was found that the programming tool used could be included both as an intervention strategy and as a moderating variable, so the studies were coded separately. CT was the dependent variable and was coded into three categories: academic achievement, thinking skill and social emotion. Since the moderating variables would affect the relationship between the independent variable and the dependent variable, the experimental design variables included in the 28 studies were classified and integrated into five moderating variables (Table 1), in which context learning stage was coded as primary school or below, junior high school and senior high school; intervention duration was coded as 0 ~ 1 weeks, 1 ~ 4 weeks, 4 ~ 12 weeks and more than 12 weeks; learning scaffold was coded

Table 1 Coding table of literature eigenvalues

Descriptive information	Variable information			Data information
	Field	Type	Explanation	
Article serial number	Independent variable	Intervention strategy	Programming tool; Teaching method	Sample size
Article title	Dependent variable	Computational thinking	Academic achievement; Thinking skill; Social emotion	Average value
Author's name	Moderating variable	Learning stage	Primary school or below; Junior high school; Senior high school	Standard deviation
Publication year		Intervention duration	0–1 week; 1–4 weeks; 4–12 weeks; more than 12 weeks	Variance
Publication journal		Learning scaffold	Technical support; Teacher support; Resource support	
		Programming tool	Graphical block programming; Text programming; Educational robot programming; Unplugged programming	
		Evaluation tool	Self-compiled questionnaire or scale; Self-compiled test; Standardized test tools	

as technical support scaffold (learning process support provided by technical tools, such as resource sharing tools, mind maps or flow chart tools), teacher support scaffold (learning process support and guidance provided by teachers, such as by answering questions or guiding reflection), and resource support scaffold (learning support provided by media resources, such as guidance materials, learning materials and research cases) according to the learning support objects used throughout the process of learners' problem-solving; programming tool, when understood as a moderating variable, was coded as graphical block programming, code text programming, educational robot programming and unplugged programming; and evaluation tool was coded as self-made questionnaires or scales, self-made test papers and standardized test tools (such as the Dr. Scratch online project evaluation platform or Bebras international test questions).

These data were mainly used to calculate the effect size to measure the change in CT. The effect size was calculated in terms of the standardized mean difference (SMD) between the pre-post test within a single group of experiments or the comparative test of two groups of experiments. Studies featuring different experimental designs often use different formulas to calculate effect size. This study adopted the SMD calculation formula proposed by Morris (2008, p. 369):

$$SMD = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{(n_1+n_2-2)}}}$$

$\bar{x}_1$ and $\bar{x}_2$ represent the mean values, s_1 and s_2 represent the standard deviations, and n_1 and n_2 represent the sample sizes of the pre-post test or control experiment groups.

3.4 Data extraction and coding

Two researchers extracted the information from the literature in accordance with the data coding table (Table 1) and recorded them in Excel. During the data extraction stage, if an article contained different studies or a study contained different measurements of the same outcome variable, the result of each study was extracted separately. For example, the study conducted by Pérez et al. (2018) included the three outcome variables of academic achievement, thinking skill and social emotion, which were regarded as three studies, and the study conducted by Jun et al. (2016) measured the results of CT development based on a pre-post test using a single group experiment and a comparative test in two groups experiments simultaneously, which were also regarded as multiple independent studies. Following the data extraction, the consistency test coefficient of the two researchers was approximately 94%, and consensus was reached via discussion and negotiation. Subsequently, the Revman 5.4 tool was used to test the sample data for publication bias and heterogeneity, and a meta-analysis of the data was conducted based on the results of this base test.

3.5 Publication bias test and heterogeneity test

3.5.1 Publication bias test

If the sample literature analyzed by a meta-analysis does not represent the overall state of research in the field in question, the research is considered to exhibit publication bias, which can affect the accuracy and reliability of the meta-analysis. Therefore, the meta-analysis must test for publication bias in the sample data (Rothstein et al., 2005). The common method used to test publication bias is the funnel plot, which takes the effect size as the abscissa and the standard error as the ordinate. If the sample data are evenly distributed on both sides of the average effect size and concentrated in the upper region, the possibility of publication bias is very small. The funnel plot associated with this study (Fig. 2) shows that the data are evenly distributed in the upper part of the effective region, indicating that there is little possibility of publication bias in this context.

3.5.2 Heterogeneity test

The heterogeneity test results concerning the sample effect size can be used to select appropriate effect models for research. The I^2 value is often used to determine the degree of sample heterogeneity in research. It is generally considered that $I^2 \geq 50\%$ refers to medium-high heterogeneity, indicating that a random effect model should be used; otherwise, a fixed effect model should be used (Lipsey & Wilson, 2000). The heterogeneity test results of this study (Table 2) showed that $I^2 = 88\%$ and exhibited significant heterogeneity ($P < 0.01$). Therefore, the random effect model should be used to calculate the overall effect size to ensure scientificity and reliability.

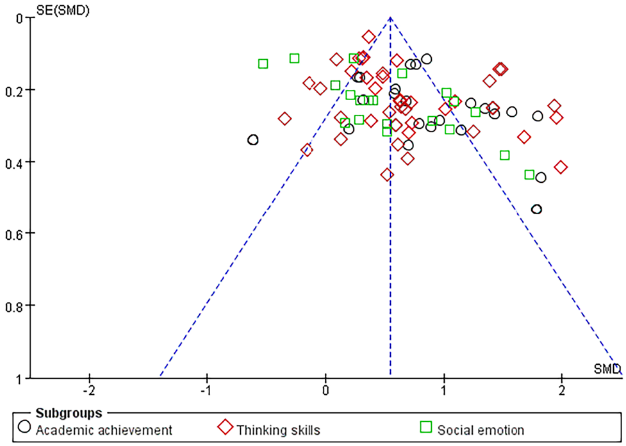

Fig. 2 Funnel plot of 98 effect quantities with regard to publication bias across 28 studies

Table 2 Heterogeneity test results

Effect model	Effect quantities	Combined effect size	95% Confidence interval		Heterogeneity test			
			Lower limit	Upper limit	Chi^2	df	*P*	I^2
Fixed effect model	98	0.55	0.51	0.59	794.27	97	<0.001	88%
Random effect model	98	0.72	0.60	0.83	794.27	97	<0.001	88%

4 Results

4.1 The overall effect size analysis

The random effect model was used to analyze 98 effect quantities across 28 empirical studies. After eliminating heterogeneity, the overall effect forest map (Fig. 3) was obtained. This map shows that the comprehensive effect size is 0.72 and is significant ($z=11.9$, $P<0.01$). According to Cohen's (1992) standard for effect size scoring, PT has a positive effect on improving the CT of K-12 students, and the overall effect is at the upper-middle level (95% CI[0.60,0.83]). Among the independent variables of intervention strategies (Table 3), programming tools ($ES=0.78$) and teaching methods ($ES=0.71$) all had positive impacts on the cultivation of CT, but there were no statistically significant differences between the groups in this context ($chi^2=0.35$, $P>0.01$). This finding indicates that the teaching methods and programming tools used in K-12 PT are both effective intervention strategies for improving the CT of K-12 students, but the effect size of programming tools is larger than that of teaching methods.

To understand the specific effects of K-12 PT in promoting the development of CT, this study analyzed three specific dimensions of CT outcomes. The results (Table 3) show that the teaching method used has a significant impact on the improvement of academic achievement ($ES=0.83$), thinking skill ($ES=0.74$) and social emotion ($ES=0.52$) with no significant differences between the groups ($chi^2=2.95$, $P>0.01$), and that the programming tool used also has a significant impact on the improvement of academic achievement ($ES=0.89$), thinking skill ($ES=0.78$) and social emotion ($ES=0.63$) with no significant differences between the groups ($chi^2=1.22$, $P>0.01$). Therefore, teaching method and programming tool both have positive effects on the three dimensions of CT, with no significant differences between the groups ($chi^2=0.35$, $P>0.01$). It is worth noting that these factors can significantly and effectively improve students' academic achievement ($ES=0.83$, $z=6.84$, $P<0.01$, 95% CI[0.59, 1.06]), but the improvements in students' thinking skills ($ES=0.74$, $z=9.15$, $P<0.01$, 95% CI[0.59, 0.90]) and social emotions ($ES=0.51$, $z=4.09$, $P<0.01$, 95% CI[0.26, 0.75]) are relatively small.

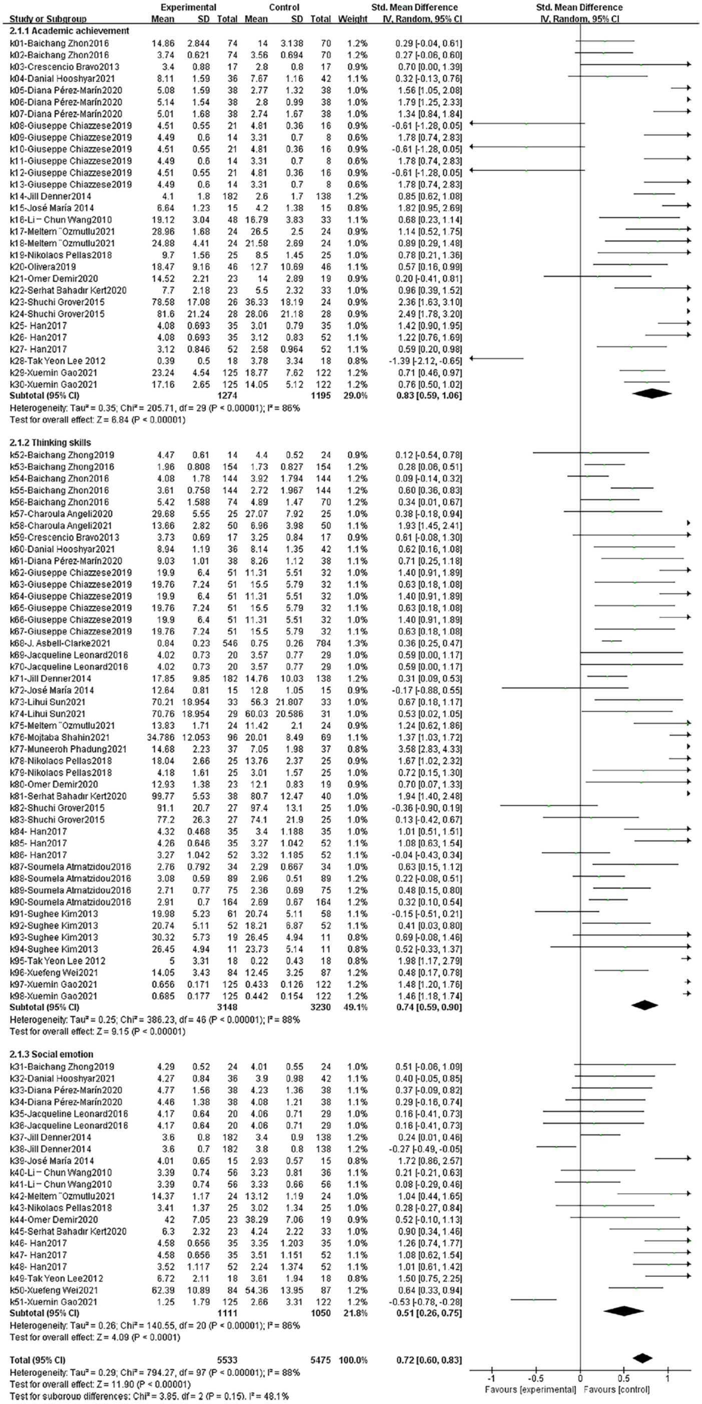

Fig. 3 Overall effect forest map

Table 3 Test of intervention strategies with respect to the three dimensions of CT outcomes

Intervention category	CT Result	Effect quantity	Effect size	95% confidence interval		Heterogeneity test		Two-tailed test	Intergroup effect	
				Lower limit	Upper limit	I^2	df(P)	Z(P)	Chi^2(P)	
Teaching method	Academic achievement	25	0.83	0.58	1.09	87%	24(<0.001)	6.36(<0.001)	$Chi^2=2.95$ P=0.23	$Chi^2=0.35$ P=0.56
	Thinking skill	34	0.74	0.56	0.92	88%	33(<0.001)	7.99(<0.001)		
	Social emotion	19	0.52	0.26	0.78	87%	18(<0.001)	3.9(<0.001)		
	Combined effect size	0.71 [0.58,0.84], Z=10.65($P<0.001$)								
Programming tool	Academic achievement	11	0.89	0.57	1.22	66%	10(<0.001)	5.38(<0.001)	$Chi^2=1.22$ P=0.54	
	Thinking skill	30	0.78	0.54	1.02	88%	29(<0.001)	6.47(<0.001)		
	Social emotion	9	0.63	0.30	0.97	71%	8(<0.001)	3.69(<0.001)		
	Comprehensive effect size	0.78 [0.60,0.95], Z=8.73($P<0.001$)								

Table 4 Test of different intervention strategies for CT

Intervention strategy	Effect size type	Effect quantity	Effect size	95% confidence interval		Heterogeneity test		Two-tailed test	Intergroup effect	
				Lower limit	Upper limit	I^2	df(P)	Z(P)	Chi^2(P)	
Teaching method	Collaborative programming	23	0.52	0.33	0.72	82%	22(<0.001)	5.32(<0.001)	$Chi^2=40.58$ $P<0.001$	$Chi^2=0.35$ $P=0.56$
	Project programming	23	0.73	0.48	0.98	91%	22(<0.001)	5.66(<0.001)		
	Game programming	25	0.70	0.47	0.93	84%	24(<0.001)	6.05(<0.001)		
	Problem-based programming	4	1.14	-0.28	2.56	95%	3(<0.001)	5.57(<0.001)		
	Scaffolding programming	3	1.84	1.48	2.20	0%	2(0.60)	9.9(<0.001)		
	Combined effect size	0.71 [0.58,0.84], Z=10.65($P<0.001$)								
Programming tool	Code text	8	0.63	0.22	1.04	75%	7(<0.001)	3.02(<0.001)	$Chi^2=6.47$ $P=0.09$	
	Graphical block	18	0.83	0.52	1.13	87%	17(<0.001)	5.31(<0.001)		
	Educational robot	18	0.92	0.60	1.24	88%	17(<0.001)	5.56(<0.001)		
	Unplugged	6	0.39	0.09	0.70	52%	5(0.07)	2.52(0.01)		
	Comprehensive effect size	0.78 [0.60,0.95], Z=8.73($P<0.001$)								

4.2 The effects of different teaching methods

The comprehensive effect size of teaching method as an intervention strategy on the cultivation of CT is 0.71 (Table 4), which is statistically significant ($z = 10.65$, $P < 0.01$), thus indicating that teaching methods can significantly improve the CT of K-12 students with an effect size at the middle-upper level (ES = 0.71, 95% CI[0.58,0.84]). This study compared the application effects of five teaching methods, i.e., collaborative programming, project-based programming, game programming, problem-based programming and scaffolding programming, based on the codes presented in 28 studies. In this context, the comprehensive effect sizes of scaffolding programming (ES = 1.84) and problem-based programming (ES = 1.14) are both larger than 0.8, thus indicating a significant level of impact; the comprehensive effect sizes of project-based programming (ES = 0.73) and game programming (ES = 0.70) are at the upper-middle level; and the comprehensive effect size of collaborative programming (ES = 0.52) is at the medium level. There was a significant difference in terms of the intergroup effect (chi2 = 40.58, $P < 0.01$), thus indicating that the use of scaffolding programming and problem-based programming as teaching methods can more effectively promote the development of CT in K-12 students than the use of other teaching methods.

4.3 The effects of different programming tools

Table 4 shows that the comprehensive effect size of the programming tool on the cultivation of CT is 0.78, which is statistically significant ($z = 8.73$, $P < 0.01$), thus indicating that programming tools can significantly improve the CT of K-12 students with an effect size at the upper-middle level (ES = 0.78, 95% CI[0.60,0.95]). Based on the coding table presented in 28 studies, this study specifically compared the application effects of four kinds of programming tools, namely, code text, graphical block, educational robot and unplugged programming. Among these tools, educational robot programming (ES = 0.92) and graphical block programming (ES = 0.83) both have significant impacts on the cultivation of CT; code text programming (ES = 0.63) has an upper-middle level of impact; and unplugged programming (ES = 0.39) has a lower-middle level of impact. Although among the programming tools used as intervention strategies, both educational robots and graphical blocks have the largest and most significant impact on the comprehensive effect size of CT, there are no significant differences between the intergroup effects ($chi^2 = 6.47$, $P > 0.05$), which does not allow us to explain which programming tools are more effective.

4.4 Moderator effect size analysis

The two-tailed test of the 98 effect quantities shown in the overall forest map (Fig. 3) showed significant heterogeneity ($I^2 = 88\%$, $z = 11.9$, $P < 0.01$), thus indicating differences among different effect sizes, which could be attributed to the

Table 5 Test of moderating effect

Moderating variable	Effect size type	Effect quantity	Effect size	95% confidence interval		Heterogeneity test		Two-tailed test	Intergroup effect	
				Lower limit	Upper limit	I^2	df(P)	Z(P)	Chi^2(P)	
Learning stage	Primary school	59	0.78	0.62	0.93	88%	58(<0.001)	10.06(<0.001)	$Chi^2=4.97$ P = 0.08	$Chi^2=1.34$ P = 0.85
	Junior high school	25	0.51	0.30	0.71	86%	24(<0.001)	4.79(<0.001)		
	Senior high school	14	0.88	0.44	1.32	87%	13(<0.001)	3.93(<0.001)		
	Comprehensive effect size	0.72 [0.60,0.83], Z = 11.9($P<0.001$)								
Intervention duration	1–4 weeks	9	0.35	0.20	0.50	31%	8(P = 0.17)	4.54(<0.001)	$Chi^2=15.9$ $P<0.001$	
	4–12 weeks	50	0.74	0.55	0.92	88%	49(<0.001)	7.74(<0.001)		
	More than 12 weeks	39	0.78	0.59	0.96	89%	38(<0.001)	8.27(<0.001)		
	Comprehensive effect size	0.72 [0.60,0.83], Z = 11.9($P<0.001$)								
Learning scaffold	Technical support	3	0.69	-1.39	2.78	96%	2(<0.001)	0.65(P = 0.52)	$Chi^2=7.87$ $P<0.05$	
	Resource support	6	0.39	0.09	0.70	52%	5(P = 0.07)	2.52(<0.01)		
	Teacher support	14	0.94	0.69	1.19	78%	13(<0.001)	7.45(<0.001)		
	Comprehensive effect size	0.83 [0.57,1.09], Z = 6.27($P<0.001$)								
Programming tool	Graphical block	55	0.68	0.53	0.83	90%	54(<0.001)	9.0(<0.001)	$Chi^2=3.30$ P = 0.35	
	Text code	8	0.63	0.22	1.04	75%	7(<0.001)	3.02(<0.05)		
	Educational robot	11	0.80	0.31	1.30	88%	10(<0.001)	3.18(<0.001)		
	Unplugged	6	0.39	0.09	0.70	52%	5(P = 0.07)	2.52(<0.01)		
	Comprehensive effect size	0.67 [0.55,0.80], Z = 10.36($P<0.001$)								
Evaluation tool	Self-compiled scale	16	0.55	0.28	0.82	89%	15(<0.001)	3.96(<0.001)	$Chi^2=4.83$ P = 0.09	
	Self-compiled test	37	0.58	0.38	0.77	85%	36(<0.001)	5.81(<0.001)		
	Standardized test	30	0.86	0.65	1.07	87%	29 <0.001)	8.03(<0.001)		
	Comprehensive effect size	0.68 [0.55,0.81], Z = 10.23($P<0.001$)								

influence of moderating variables other than sampling error. To explore the key factors affecting CT in further detail, subgroup analysis was conducted to investigate potential moderating variables that may lead to significant heterogeneity, such as the learning stages, intervention durations, learning scaffolds, programming tools and evaluation tools included in the 28 experimental designs. The results (Table 5) show that different moderating variables have positive effects on CT, with no statistically significant differences between the groups ($chi^2 = 1.34$, $P > 0.05$). In this context, the intervention duration ($ES = 0.72$, $P < 0.05$) and learning scaffold ($ES = 0.83$, $P < 0.05$) have significant effects and could be considered to be the key moderating variables for promoting the improvement of CT. The specific results are as follows:

(1) Different learning stages had a positive impact on CT, and there were no significant differences between groups ($chi^2 = 4.97$, $P > 0.05$). The ranking of effect sizes began with senior high school ($ES = 0.88$), followed by primary school ($ES = 0.78$) and junior middle school ($ES = 0.51$). This finding indicates that although learning stage has a positive effect on the cultivation and development of CT, it cannot be regarded as the decisive factor affecting CT in the context of PT. In addition, researchers are more inclined to conduct empirical studies in primary school (accounting for 61% of the total number of studies conducted); however, the comprehensive effect size of senior high school is the highest, thus indicating that senior high school may be the best stage in which to cultivate CT.

(2) Different intervention durations had a significant positive impact on CT, and the differences between groups were significant ($chi^2 = 15.9$, $P < 0.01$). The effect sizes associated with this factor indicated an upward trend with increasing intervention duration. After more than 12 weeks of intervention, the improvement in CT reached an upper-middle level ($ES = 0.78$). This finding indicates that intervention duration is positively correlated with the effect of CT, such that the longer the intervention duration is, the larger the effect size.

(3) Different learning scaffolds had a positive effect on CT, and the differences between groups were statistically significant ($chi^2 = 7.87$, $P < 0.05$). In this context, the teacher support scaffold ($ES = 0.94$) exhibited a high degree of significant effect, the technical support scaffold ($ES = 0.69$) reached a medium-upper degree of effect, and the resource support scaffold ($ES = 0.39$) was associated only with a lower-medium degree of effect. This finding indicates that the learning scaffold featuring the support of the teacher has the best effect on the promotion of CT.

(4) As moderating variables, different programming tools had positive impacts on CT, and there were no significant differences between groups ($chi^2 = 3.30$, $P > 0.05$). In this context, educational robot programming ($ES = 0.8$) had a significant level of effect, graphical block programming ($ES = 0.68$) and code text programming ($ES = 0.63$) both had upper-middle levels of effect, and unplugged programming ($ES = 0.39$) had a lower-middle level of effect. This finding indicates that the programming tool used, taken as a moderating variable, cannot be regarded as the decisive factor affecting CT; accordingly, any programming tool can be used as a tool to support the teaching content.

(5) Different evaluation tools had positive impacts on CT, and there were no significant differences between groups ($chi^2 = 4.83$, $P > 0.05$). In this context, the comprehensive effect size of the standardized test was the highest (ES = 0.86), reaching a significant level of effect, while the self-made test (ES = 0.58) and the self-made questionnaire or scale (ES = 0.58) reached only a medium level. This finding indicates that although the evaluation tool that is selected based on different experimental contents has a positive promotional effect on CT, it cannot be regarded as the decisive factor affecting CT.

5 Discussion

5.1 The effectiveness of PT itself on CT

The meta-analysis indicates that PT has a positive effect on the improvement of CT among K-12 students as well as a positive promotional effect on the three dimensions of CT outcomes. The associated improvement in academic achievement is significant, but the corresponding improvements in thinking skills and social emotion are relatively small. Some studies have reported that in the context of classroom teaching, CT and subject courses exhibit a higher degree of integration and greater consistency, which is significantly effective in promoting academic achievement (Lei et al., 2020). In addition, students' social emotions affect the efficiency of programming; if the programming process is both interesting and exciting, i.e., challenging and rewarding, this situation can encourage students to fully realize their potential to solve problems and thus improve their thinking skills and cognition (Papadakis, 2018; Wei et al., 2021). Similarly, their thinking skills also affect their cognition and social emotion throughout the programming process (Wei et al., 2021), and students who have more motivation and positive social emotions tend to learn better and exhibit better academic achievement (Sison, 2008). This finding sheds light on the subtle relationships among academic achievement, thinking skills and social emotions with regard to the outcomes of CT in the context of PT, which affects the three specific dimensions of CT as well as, in turn, CT as a whole. Therefore, future empirical research should focus more carefully on social emotions and thinking skills to promote the overall improvement of CT.

5.2 The effectiveness of the teaching method used with regard to CT

The use of various teaching methods as an intervention strategy exhibits an upper-middle level of effect. Scaffolding programming and problem-based programming are more effective in promoting the development of CT. As some studies have found, a combination of problem-based teaching with learning scaffolds is the most popular teaching mode (Wang et al., 2021; Chen et al., 2021), and the use of cooperative problem solving in PT as well as the provision of problem-solving scaffolds are extremely beneficial for beginners in programming or for dealing with poorly structured and complex problem solving (Witherspoon et al., 2017; Wang et al., 2021), which can encourage students to explain and reflect on subject knowledge and to overcome the meta-cognitive difficulties they face in the context of understanding and solving problems (Bulu & Pedersen, 2012). In addition, this meta-analysis found that although collaborative programming could positively promote the development of CT, it did not exhibit the expected higher effectiveness. Although some studies have proposed that collaborative programming can produce more diverse programming schemes and that this approach ultimately has a greater potential to solve problems (Denner et al., 2014), PT involves a variety of of semantic and syntactic knowledge, coding skills and algorithm logic in practice. If collaborative tasks are not designed carefully, they can impose greater cognitive burdens on students, thus causing strong negative emotions and learning disabilities (Rogerson & Scott, 2010). Therefore, not every programming task in the context of PT requires cooperation (Siddiq & Scherer, 2017).

5.3 The effectiveness of the programming tool used with regard to CT

The investigation of the use of programming tools as an intervention strategy indicates that all kinds of programming tools can positively promote CT; however, the effect size differences between groups in this case are not significant, which does not allow us to explain which programming tools are most effective in promoting the development of CT in the context of PT. Some studies have shown that educational robot programming and graphical block programming can allow tedious grammar learning to be avoided and can reduce the cognitive load associated with reading, understanding and creating text codes (Wang et al., 2021; Sengupta et al., 2015), contribute to the creation of coding sequences, code functions and psychological models of real problem solutions (Lieto et al., 2017), and facilitate observation of the specific behaviors of real or virtual objects after programming with the aim of obtaining immediate feedback (Grover & Pea, 2013), which can improve students' motivation to learn programming and effectively promote the application of students' thinking ability to the construction of programming knowledge and the task of solving problems in real collaborative situations (Lye & Koh, 2014). However, the results of the meta-analysis show not only that the use of programming tools can have a positive impact on CT but also that these effects are equivalent and do not exhibit significant differences. Clark (1994, p. 27) noted that "media and their attributes only affect the cost or speed of learning but only the use of adequate instructional methods will affect learning". Therefore, the use of essentially different learning tools does not significantly affect learning effectiveness, and the acceptance of students and the usability of programming tools should be taken into account in the context of PT rather than blindly using emerging technologies, which can increase students' cognitive burdens and exacerbate the social emotions of anxiety (Zhang et al., 2021).

5.4 The moderating effects of moderating variables on CT

As part of the meta-analysis, we conducted subgroup analysis on the potential moderating variables contained in 28 experimental designs. The results indicate that the moderating variables of learning stage, intervention duration, learning scaffold, programming tool and evaluation tool can all impact the effect of PT on CT.

With regard to learning stage, although different learning stages can positively affect the cultivation and development of CT, the intergroup effect is not statistically significant, indicating that it cannot be regarded as a decisive factor affecting CT. Students' thinking cognition varies across different learning stages, which is reflected in the effectiveness and pertinence of programming knowledge and problem situation design at different learning stages. Therefore, other variables may be more important, such as intervention duration and learning scaffold, for improving students' CT.

With respect to intervention duration, the comprehensive effect size exhibits an upward trend with increasing intervention duration. Therefore, intervention duration is positively correlated with the cultivation effect of CT. This result is consistent with the conclusions of some studies. For instance, cognitive thinking differs from subjective knowledge, and the cultivation and development of the former is a process of gradual accumulation (Halpern, 2001). CT, as a problem-solving ability, cannot be cultivated or greatly improved over a short period of time but only via long-term teaching practice (Saad & Zainudin, 2022). Therefore, future empirical research should consider these limitations over a longer period of teaching practice.

With regard to programming tools, the programming tool, as a moderating variable, has a positive impact on CT with no significant differences between groups, which is consistent with the analysis results concerning the use of programming tools as an intervention strategy. Therefore, when viewed either as an intervention strategy or as a moderating variable, the use of programming tools is not a decisive factor affecting CT, although it can be used as a tool to support teaching content.

With respect to learning scaffolds, the three types of learning scaffolds have positive impact on CT. The comprehensive effect size of the teacher-supported learning scaffold on CT cultivation is the highest, indicating that PT, as a teaching practice aimed at solving problems and promoting the development of thinking ability, cannot be separated from effective learning scaffolds. This conclusion is consistent with the findings of some previous studies; for instance, learning scaffolds are an effective way of encouraging students to participate in cooperation, engage in problem solving and improve their high-order thinking (Reiser & Brian, 2004), and learning scaffolds can both reduce task difficulty and negative emotions and encourage learners to participate in learning tasks featuring problem solving (Wood et al., 2006). As a form of learning support, the purpose of a learning scaffold is to help students use learning strategies more effectively to adjust the process of problem solving in the context of programming. In this process, the teacher-supported learning scaffold is more targeted, instructive and timely, and so it has the most significant impact on CT.

In terms of evaluation tools, Dr. Scratch online project evaluation and Bebras international test questions have been recognized as effective and reliable standardized measurement tools by international scholars. However, only 36.14% of the studies included in this meta-analysis used standardized evaluation tests, and 63.86% of the studies used self-made questionnaires or evaluation scales and self-made test papers; in addition, the results indicated no differences between groups at a medium level of effect. This finding indicates that standardized evaluation scales are not suitable for evaluating CT in all teaching situations. As noted by RN (2002, p. 91), "the measurement tools pointing to thinking ability have limitations in evaluating students in different contexts and should be measured from multiple perspectives by combining standardized test with self-report". Therefore, in the context of the specific PT practice of cultivating CT, we must adapt the standardized methods of measuring CT appropriately based on the programming knowledge and learning situations associated with different stages with the aim of improving our ability to assess the resulting changes in students' CT.

5.5 Some suggestions for PT practice

The results of the meta-analysis not only show that PT can effectively improve the CT of K-12 students but also indicate the programming tools and teaching methods that are most effective. To teach CT more effectively in the context of PT practice, in addition to the suggestions mentioned in the preceding discussion, the following reference suggestions are provided.

(1) Focus on the problem-based programming and scaffolding programming teaching methods.

Problem-based programming and scaffolding programming both have a strong synergistic effect with PT in the context of cultivating CT. Because problem-based PT practice involves a range of semantic and grammatical knowledge as well as coding skills and algorithmic logic, it is necessary to provide learning scaffolding in this context to help students identify poorly structured problems in real-life situations while reducing their cognitive load, which can drain their mental resources. Therefore, teachers should focus on the teaching methods of problem-based programming and scaffolding programming by designing real problems and effective learning scaffolds, which can significantly improve students' CT in the context of PT practice. In addition, teachers should encourage students to debate and negotiate during the problem-solving process, provide timely teacher-supported learning scaffolds, and help students become problem solvers and collaborative scheme designers who can exhibit CT.

(2) Focus on the learning scaffold supported by teachers and strengthen the teaching training provided to teachers.

The cultivation of CT cannot be separated from the support of an effective learning scaffold. The teacher-supported learning scaffold is not only more targeted, more directive and timelier, it also has the most significant effect on CT. Only when teachers realize the importance of CT with respect to the development of students and apply the appropriate teaching methods to the design of teaching activities can students' CT be developed effectively. Therefore, it is particularly important to strengthen the CT teaching training provided teachers, especially preservice teachers, to allow them to provide an effective learning scaffold for the PT that is oriented on the cultivation of CT. As a form of learning support, the teacher-supported learning scaffold aims to help students use learning strategies and thinking skills more effectively to adjust the process of solving problems in the context of programming. This scaffold plays a very important role in the cultivation of students' CT in the context of PT practice. If teachers purposefully and persistently guide students to combine CT with programming knowledge content and problematic situations, students' academic achievement, thinking skills and social emotions can be effectively improved.

5.6 Limitations

This meta-analysis faces certain limitations, and subsequent studies can address these deficiencies. First, the sample size was insufficiently large because the search language was limited to English, which may have led to the exclusion of relevant studies published in other languages. Second, the dimensional information regarding the moderating variables was limited, and future research can explore more dimensional information associated with CT by conducting relevant studies. Third, this meta-analysis was time-limited because more studies were published during the process of completing this meta-analysis, as is the case for most review articles.

6 Conclusion

Using a meta-analytic approach, this study conducted a comprehensive quantitative analysis of 28 empirical studies pertaining to the cultivation of CT in the context of PT in K-12 education, involving a total of 4154 samples and 98 effect quantities, thereby effectively answering the question of how CT can be taught more effectively, which has been raised by previous studies. The following conclusions can be drawn:

(1) PT can improve the CT of K-12 students, and the overall effect size is at the upper-middle level (ES = 0.72, z = 11.9, $P < 0.01$, 95% CI[0.60, 0.83]). With regard to the dimension of CT outcomes, PT can significantly and effectively improve students' academic achievement (ES = 0.83, z = 6.84, $P < 0.01$, 95% CI[0.59, 1.06]), but it is insufficient with respect to improving students' thinking skills (ES = 0.74, z = 9.15, $P < 0.01$, 95% CI[0.59, 0.90]) and social emotions (ES = 0.51, z = 4.09, $P < 0.01$, 95% CI[0.26, 0.75]).

(2) The effectiveness of the teaching method used is at the upper-middle level (ES = 0.71, z = 10.65, $P < 0.01$, 95% CI[0.58, 0.84]). In this context, scaffolding programming and problem-based programming can promote the development of CT in K-12 students more effectively than other teaching methods.

(3) The effectiveness of the programming tool used is at the upper-middle level (ES = 0.78, z = 8.73, $P < 0.001$, 95% CI [0.60, 0.95]), but this result cannot explain which programming tools are most effective, as there are no significant intergroup effect differences among different programming tools ($chi^2 = 6.47$, $P > 0.05$).

(4) All five moderating variables contained in the 28 experimental designs have positive impacts on the CT of K-12 students. In this context, intervention duration (ES = 0.72, z = 11.9, $P < 0.05$, 95% CI[0.60, 0.83]) and learning scaffold (ES = 0.83, z = 6.27, $P < 0.05$, 95% CI[0.57, 1.09]) can be regarded as the decisive factors that affect CT in the context of PT.

Author contributions All authors contributed to the study conception and design. Enwei Xu and Qingxia Wang conducted the literature search and were involved in the analysis and interpretation of data as well as writing-review and editing. Enwei Xu and Wei Wang drafted and critically revised the manuscript. All authors read and approved the final manuscript.

Funding This work was supported by the graduate scientific research and innovation project of Xinjiang Uygur Autonomous Region named "Research on In-Depth Learning of High School Information Technology Courses for the Cultivation of Computing Thinking" (Project No.: XJ2022G190), and the independent innovation fund project for doctoral students of the College of Educational Science of Xinjiang Normal University named "Research on Project-based Teaching of High School Information Technology Courses From the Perspective of Discipline Core Literacy" (Project No.: XJNUJKYA2003).

Data availability The datasets supporting the conclusions of this article are included within the article or contact the first author of this article to obtain the original data.

Declarations

Conflict of competing interest All authors have no relevant financial or non-financial competing interests.

References

Angeli, C., & Giannakos, M. (2020). Computational thinking education: issues and challenges. *Computers in Human Behavior, 105*, 106-185. https://doi.org/10.1016/j.chb.2019.106185.

Beck, L., & Chizhik, A. (2013). Cooperative Learning Instructional Methods for CS1:Design, Implementation and Evaluation. *ACM Transactions on Computing Education, 13*(3), 1–21. https://doi.org/10.1145/2499947.2492686.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *2012 annual meeting of the American educational research association* (pp. 1–25).

Bulu, S. T., & Pedersen, S. (2012). Supporting problem-solving performance in a hypermedia learning environment:the role of students' prior knowledge and meta-cognitive skills. *Computers in Human Behavior, 28*(4), 1162–1169. https://doi.org/10.1016/j.chb.2012.01.026.

Chen, J., Wang, M., Kischner, P., & Tsai, C. (2018). The role of collaboration, computer use, learning environments, and supporting strategies in CSCL: A meta-analysis. *Review of Educational Research, 88*(6), 799–843. https://doi.org/10.3102/0034654318791584.

Chen, C. H., Liu, T. K., & Huang, K. (2021). Scaffolding vocational high school students' computational thinking with cognitive and metacognitive prompts in learning about programmable logic controllers. *Journal of Research on Technology in Education, 20*, 1–18. https://doi.org/10.1080/15391523.2021.1983894.

Clark, R. E. (1994). Media will never influence learning. *Educational Technology Research and Development, 42*(2), 21–29. https://doi.org/10.2307/30218684.

Cohen, A. (1992). Antecedents of organizational commitment across occupational groups: a meta-analysis. *Journal of Organizational Behavior*. https://doi.org/10.1002/job.4030130602.

Cooper, H. (2010). *Research synthesis and meta-analysis:A step-by-step approach* (4th ed.). Sage.

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education, 46*(3), 277–296. https://doi.org/10.1080/15391523.2014.888272.

Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: can they be used to measure understanding of computer science concepts? *Computers & Education, 58*(1), 240–249. https://doi.org/10.1016/j.compedu.2011.08.006.

Denny, P., Cukierman, D., & Bhaskar, J. (2015). Measuring the effect of inventing practice exercises on learning in an introductory programming course. *Koli Calling Conference on Computing Education Research*. ACM.

Florez, F. B., Casallas, R., Hernandez, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation' s way of thinking: Teaching computational thinking through programming. *Review of Educational Research, 87*(4), 834–860. https://doi.org/10.3102/0034654317710096.

Grover, S., & Pea, R. (2013). Computational thinking in k-12: a review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189X12463051.

Halpern. (2001). Assessing the effectiveness of critical thinking instruction. *The Journal of General Education, 50*(4), 270–286. https://doi.org/10.2307/27797889.

Hsu, T., Chang, S., & Hung, Y. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers Education, 126*, 296–310. https://doi.org/10.1016/j.compedu.2018.07.004.

Jun, S. J., Han, S. K., & Kim, S. H. (2016). Effect of design-based learning on improving computational thinking. *Behaviour & Information Technology*, 1–11. https://doi.org/10.1080/0144929x.2016.1188415.

Kalelioglu, F., & Gulbahar, Z. (2014). The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education, 13*(1), 33–50.

Kim, Y. (2020). The effects of pbl-based data science education classes using app inventor on elementary student' computational thinking and creativity improvement. *Journal of the Korean Association of Information Education, 24*(6), 551–562. https://doi.org/10.14352/jkaie.2020.24.6.551.

Lai, X., & Wong, K. W. (2022). Collaborative versus individual problem solving in computational thinking through programming: a meta-analysis. *British Journal of Educational Technology, 53*, 150–170. https://doi.org/10.1111/bjet.13157.

Lee, Y. Y., Mauriello, M. L., Ahn, J., & Bederson, B. B. (2014). CTArcade: Computational thinking with games in school age children. *International Journal of Child-Computer Interaction, 2*, 26–33. https://doi.org/10.1016/j.ijcci.2014.06.003.

Lei, H., Chiu, M. M., Li, F., Wang, X., & Geng, Y. J. (2020). Computational thinking and academic achievement: a meta-analysis among students. *Children and Youth Services Review*, 118105439. https://doi.org/10.1016/j.childyouth.2020.105439.

Lipsey, M., & Wilson, D. (2000). Practical meta-analysis. *Internatinal Educational and Professional*, 92–160. London

Lieto, M., Inguaggiato, E., Castro, E., Cecchi, F., & Dario, P. (2017). Educational robotics intervention on executive functions in preschool children: a pilot study. *Computers in Human Behavior, 71*(C), 16–23. https://doi.org/10.1016/j.chb.2017.01.018.

Lye, S. Y., & Koh, J. (2014). Review on teaching and learning of computational thinking through programming: what is next for k-12? *Computers in Human Behavior, 41(dec.)*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012.

Lu, M., Odell, K., & Epp, C. (2022). A scoping review of computational thinking assessments in higher education. *Journal of Computing in Higher Education, 34*, 416–461. https://doi.org/10.1007/s12528-021-09305-y.

Morris. (2008). Estimating effect size from the pretest-posttest-control design. *Organizational Research Methods, 11*, 364–386.

Ministry of Education of the People's Republic of China. (2022). *Information technology curriculum standard for compulsory education* [2022-05-07]. http://www.moe.gov.cn/srcsite/A26/s8001/202204/W020220420582361024968.pdf.

Ministry of Education of the People's Republic of China. (2020). *Information technology curriculum standard for senior high School* [2020-05-20]. http://www.moe.gov.cn/srcsite/A26/s8001/202006/t20200603_462199.html.

Papadakis, S. (2018). Is pair programming more effective than solo programming for secondary education novice programmers? A case study. *International Journal of Web-Based Learning and Teaching Technologies, 13*(1), 1–16. https://doi.org/10.4018/ijwltt.2018010101.

Pérez-Marín, R., Hijón-Neira, B. A., & Pizarro, C. (2018). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? *Computers in Human Behavior, 105,* 105849. https://doi.org/10.1016/j.chb.2018.12.027.

Reiser, & Brian, J. (2004). Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *Journal of the Learning Sciences, 13*(3), 273–304. https://doi.org/10.2307/1466939.

Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (SIGCSE'10), 265–269. ACM Press.

Rn, E., & Rn, M. (2002). Critical thinking in nursing education: Literature review. *Mary Courtney, 8*(2), 89–98.

Rogerson, C., & Scott, E. (2010). The fear factor: How it affects students learning to program in a tertiary environment. *Journal of Information Technology Education, 9*(1), 147–171. https://doi.org/10.28945/1183.

Roman Gonzalez, M., Moreno-Leon, J., & Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In S. Kong, & H. Abelson (Eds.), *Computational thinking education* (pp. 79–98). Springer. https://doi.org/10.1007/978-981-13-6528-7_6.

Rothstein, H. R., Sutton, A. J., & Borenstein, M. (2005). Publication bias and meta-analysis: prevention, assessments and adjustments, 261–286. https://doi.org/10.1002/0470870168.ch14.

Saad, A., & Zainudin, S. (2022). A review of project-based learning (pbl) and computational thinking (ct) in teaching and learning. *Learning and Motivation*, *78*(12). https://doi.org/10.1016/j.lmot.2022.101802.

Scherer, R. (2016). Learning from the past-the need for empirical evidence on the transfer effects of computer programming skills. *Frontiers in Psychology*, *7*(1390). https://doi.org/10.3389/fpsyg.2016.01390.

Sengupta, D., Farris, V., & Wright, M. (2015). Programming in K-12 science classrooms. *Communications of the ACM, 58*(11), 34–35. https://doi.org/10.1145/2822517.

Selby, C., & Woollard, J. (2013). Computational thinking: The developing definition[DB/OL]. [2021-07-12]. http://eprints.soton.ac.uk/356481.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003.

Siddiq, F., & Scherer, R. (2017). Revealing the processes of students'interaction with a novel collaborative problem solving task: An in-depth analysis of think-aloud protocols. *Computers in Human Behavior, 76*, 509–525. https://doi.org/10.1016/j.chb.2017.08.007.

Sison, R. (2008). Investigating pair programming in a software engineering course in an Asian setting. In *2008 15th Asia-Pacific Software Engineering Conference* (pp. 325–331). IEEE.

Sun, L., Guo, Z., & Hu, L. (2021). Educational games promote the development of students' computational thinking: a meta-analytic review. *Interactive Learning Environments*. https://doi.org/10.1080/10494820.2021.1931891.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education, 148*, 103798. https://doi.org/10.1016/j.compedu.2019.103798.

Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in k-12 education: a conceptual model based on a systematic literature review. *Computers & Education, 162*(1), 104083. https://doi.org/10.1016/j.compedu.2020.104083.

Vahldicka, A., Faraha, P. R., Marcelino, M. J., & Mendes, A. J. (2020). A blocks-based serious game to support introductory computer programming in undergraduate education. *Computers in Human Behavior Reports, 2*, 100037. https://doi.org/10.1016/j.chbr.2020.100037.

Wang, C., Shen, J., & Chao, J. (2021). Integrating computational thinking in STEM education: A literature review. *International Journal of Science and Mathematics Education*. https://doi.org/10.1007/s10763-021-10227-5.

Wei, X., Lin, L., Meng, N., Tan, W., Kong, S. C., & Kinshuk. (2021). The effectiveness of partial pair programming on elementary school students' computational thinking skills and self-efficacy. *Computers & Education, 160*, 104023. https://doi.org/10.1016/j.compedu.2020.104023.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35.

Wing, J. M. (2010). Computational thinking: What and why? *The link magazine* [2022-05-20]. Retrieved fromhttp://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf.

Witherspoon, E. B., Higashi, R. M., Schunn, C. D., Baehr, E. C., & Shoop, R. (2017). Developing computational thinking through a virtual robotics programming curriculum. *ACM Transactions on Computing Education, 18*(1), 1–20.

Wood, D., Bruner, J. S., & Ross, G. (2006). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry, 17*(2), 89–100. https://doi.org/10.1111/j.1469-7610.1976.tb00381.x.

Yuksel, H. (2015). The effect of the computer assisted instruction on the academic achievement and retention of technical programme students' in vocational foreign language. *Procedia-Social and Behavioral Sciences, 174*, 2513–2518. https://doi.org/10.1016/j.sbspro.2015.01.924.

Zhang, J.B., & Ji, L.Y. (2018). Evolution of the orientation of information technology education in primary and secondary schools. *Research on Audio Visual Education, 39*(05), 108–114. https://doi.org/10.13811/j.cnki.eer.2018.05.016.

Zhang, Y., Luo, R., Zhu, Y., & Yin, Y. (2021). Educational robots improve k-12 students' computational thinking and stem attitudes: systematic review. *Journal of Educational Computing Research, 59*(7), 1450–1481. https://doi.org/10.1177/0735633121994070.

Authors and Affiliations

Enwei Xu[1] · Wei Wang[1] · Qingxia Wang[1]

Wei Wang
345979411@qq.com

Qingxia Wang
wqx19830623@163.com

[1] College of Educational Science, Xinjiang Normal University, No. 100 Guanjing Road, Shuimogou District, Urumqi, Xinjiang 830017, China