Journal of Education and Learning (EduLearn)
Vol. 20, No. 4, November 2026, pp. 2632~2641
ISSN: 2089-9823, DOI: 10.11591/edulearn.v20i4.26548

Effects of robotics-assisted programming instruction on students' computational thinking: a meta-analysis

Haipeng Yang[1], Helmi Norman[1], M. Khalid M. Nasir[1], Nian Xu[2]
[1]Faculty of Education, Universiti Kebangsaan Malaysia, Selangor, Malaysia
[2]Faculty of Educational Sciences, Kashgar University, Xinjiang, China

Article Info

Article history:

Received Nov 16, 2025
Revised May 5, 2026
Accepted Jun 8, 2026

Keywords:

Computational thinking
Educational robotics
Meta-analysis
Programming instruction
Subgroup analysis

ABSTRACT

Evidence on whether educational robotics (ER)–assisted programming instruction improves students' computational thinking (CT) remains mixed. To synthesize existing findings, we conducted a meta-analysis of 33 empirical studies (39 effect sizes) published between 2015 and 2025. Eligible studies were identified through searches of Web of Science, ScienceDirect, Taylor & Francis Online, Wiley Online Library, SAGE Journals, IEEE Xplore, and Google Scholar. Effect sizes were calculated using Hedges' g, and a random-effects model was adopted because substantial heterogeneity was observed (I^2=79.61%, Q=186.369, p<0.001). The results showed a statistically significant moderate positive effect of ER-assisted programming instruction on CT (g=0.621, p<0.001). Subgroup analyses further indicated that intervention duration and programming mode explained part of the heterogeneity: longer interventions produced larger effects, and plugged programming yielded stronger gains than unplugged programming. These findings support ER as an effective approach for promoting CT in programming instruction and suggest that instructional duration and programming mode should be carefully considered in ER-supported learning design.

Corresponding Author:

Helmi Norman
Faculty of Education, Universiti Kebangsaan Malaysia
Selangor, Malaysia
Email: helmi.norman@ukm.edu.my

1. INTRODUCTION

In today's era of rapid information technology development, the widespread use of computers and the ongoing advancement of informatization and intelligence are profoundly shaping people's lifestyles and thought patterns, driving social progress and national development [1]. This trend is prompting education systems worldwide to rethink their talent development objectives and spotlight digital literacy education centered on computational thinking (CT) [2]. CT refers to the mental process of identifying problems and designing systematic, algorithmic solutions that can be effectively executed by humans or computational agents [3]. Many countries recognize that programming education is a key path to cultivating students' CT and are gradually incorporating it into basic education systems to foster students' logical reasoning, problem-solving, and innovation skills, thereby better adapting to the demands of a digital society [4]. Among the technical tools used in programming instruction, educational robotics (ER), as a learning medium that combines software and hardware, has injected new vitality into programming education [5].

In contrast to conventional programming methods (such as text-based or visual block programming), ER provides students with a visual and interactive programming learning experience, greatly reducing students' stereotype that programming is "boring and tedious" [6]. As a flexible and creative teaching tool, ER can create an experiential, context-based learning environment, thereby stimulating students' interest, motivation, and active engagement while supporting the development of CT [7]. In the process of building, programming, and interacting with robots, students will actively apply the key components of CT—including abstraction, decomposition, pattern recognition, algorithmic thinking, and debugging—and transfer these abilities to solving real-world problems [8]. From a constructivist learning perspective, ER provides students with opportunities for exploration and reflection, enabling them to apply new knowledge to different situations and cultivate innovative problem-solving skills [9]. With these advantages, a variety of ER kits, such as Micro:bit and Arduino, have been widely used in programming classrooms around the world [10]. These applications range from basic robot operation to comprehensive task-based projects involving design, programming, and testing, enabling students to develop CT skills through practice [11].

Although extensive research has shown the significant educational value and positive impact of integrating ER into programming instruction in promoting learners' CT [12], the empirical evidence remains inconsistent. For example, Yolcu and Demirer [13] noted that the experimental group using m-Bot for robotic programming significantly outperformed the control group, which only engaged in block programming, in terms of programming success rate and learning transfer, but no significant difference was found in CT. Similarly, Weng *et al*. [14] found no significant improvement in students' overall CT skills in a project-based programming course for first-year students at university. The researchers believe this may be related to short teaching cycles, insufficient feedback, and differences in learning motivation. On the other hand, Noh and Lee [15] argued that ER did not significantly promote CT among learners with some programming experience. In sum, these mixed findings suggest that ER may promote learning indirectly by increasing interest, engagement, and transfer, whereas its direct contribution to CT development may depend on learner characteristics and instructional conditions. Therefore, a consensus has yet to be reached on the actual effectiveness of ER in promoting CT.

To clarify this issue, some scholars have used meta-analysis to synthesize research on how ER used in programming education influences students' CT. However, existing reviews and meta-analyses have mainly focused on a single educational stage, such as elementary education, or on a specific instructional approach, such as project-based learning, which limits a broader understanding of ER effectiveness across instructional contexts. In addition, prior syntheses have not sufficiently examined several moderating factors that may shape the effects of ER interventions, including educational stages [16], teaching strategies [17], interaction types [18], intervention duration [19], and programming methods [20]. Therefore, the present study uses a meta-analytic approach to offer a broader synthesis of how ER-assisted programming instruction affects students' CT. It estimates the overall effect of ER interventions and examines whether these moderating factors explain variation in the observed effects. The novelty of this study lies in three aspects. First, it specifically synthesizes empirical evidence on ER-assisted programming instruction for CT, rather than discussing ER in a broader and less focused manner. Second, it integrates evidence across different instructional contexts and systematically compares multiple moderators that may explain variations in effectiveness. Third, it provides more instructionally actionable evidence by identifying the conditions under which ER can better promote students' CT.

2. METHOD

2.1. Literature search and inclusion criteria

To find empirical studies on the impact of ER on students' CT in programming education, we conducted a systematic literature search of Web of Science, ScienceDirect, Taylor & Francis Online, Wiley Online Library, SAGE Journals, IEEE Xplore, and Google Scholar databases. We limited the search to peer-reviewed empirical publications (journal articles and conference proceedings) written in English and published between 2015 and 7 October 2025 (final search date: 7 October 2025). The search terms combined three main concept blocks using Boolean operators: ("educational robotics" OR "educational robot" OR "programming robot" OR "robotics education" OR "robot-assisted learning") AND ("programming education" OR "coding education" OR "computer programming") AND ("computational thinking"). Because search interfaces and field tags vary across databases, minor database-specific syntax adaptations were applied while preserving the same three-block logic and Boolean structure. After the database search, backward and forward snowballing was used to identify additional relevant studies from the reference lists and citations of the included articles. This process initially extracted 2,368 records.

Table 1 summarizes the inclusion and exclusion criteria used for study selection. This study followed the preferred reporting items for systematic reviews and meta-analyses (PRISMA) guidelines [21] to conduct a literature search and screening. To ensure the scientific nature of the included literature and its relevance to the research questions, we used the following inclusion and exclusion criteria. For the empirical research, only papers based on empirical research were included. Review and theoretical studies were not considered. In terms of combination of ER and programming education, the research must use programming ER projects as the teaching medium and measure students' CT during programming learning. Studies that do not include both elements will be excluded. Regarding the educational stage, the research subjects must include students in kindergarten, primary school, secondary school, or higher education. Lastly, for the research design, the literature must adopt an experimental or quasi-experimental design. Single-group experiments must report pre-test and post-test data; two-group experiments should include an experimental group that receives ER programming instruction and a control group that learns under other teaching conditions. Eligible studies were required to report sufficient statistical information—such as sample size, mean, standard deviation (SD), or T/F-value—to allow the calculation of the effect size (Hedges' *g*). Studies that did not meet these criteria were excluded.

In total, 33 independent studies met all inclusion criteria and were retained for the meta-analysis, as shown in the PRISMA flow diagram in Figure 1. Some studies reported multiple independent effect sizes, resulting in a total of 39 effect sizes for subsequent analysis. In such cases, we extracted more than one effect size from a single paper only when the effect sizes were based on independent, non-overlapping participant samples (e.g., different teaching classes or different grade-level cohorts). Accordingly, no paper contributed more than one effect size from the same participant sample.

Table 1. Inclusion and exclusion criteria

Domain	Inclusion	Exclusion
Study type	Empirical studies	Reviews, theoretical papers, or conceptual papers
Content focus	ER used in programming instruction, with CT measured as an outcome	No ER-based programming component or no CT outcome
Population	Students from kindergarten to higher education	Non-student samples or populations outside kindergarten to higher education
Design	Experimental or quasi-experimental designs	Non-experimental studies
Reporting	Sufficient statistical information to calculate Hedges' g	Insufficient statistical information to calculate effect sizes

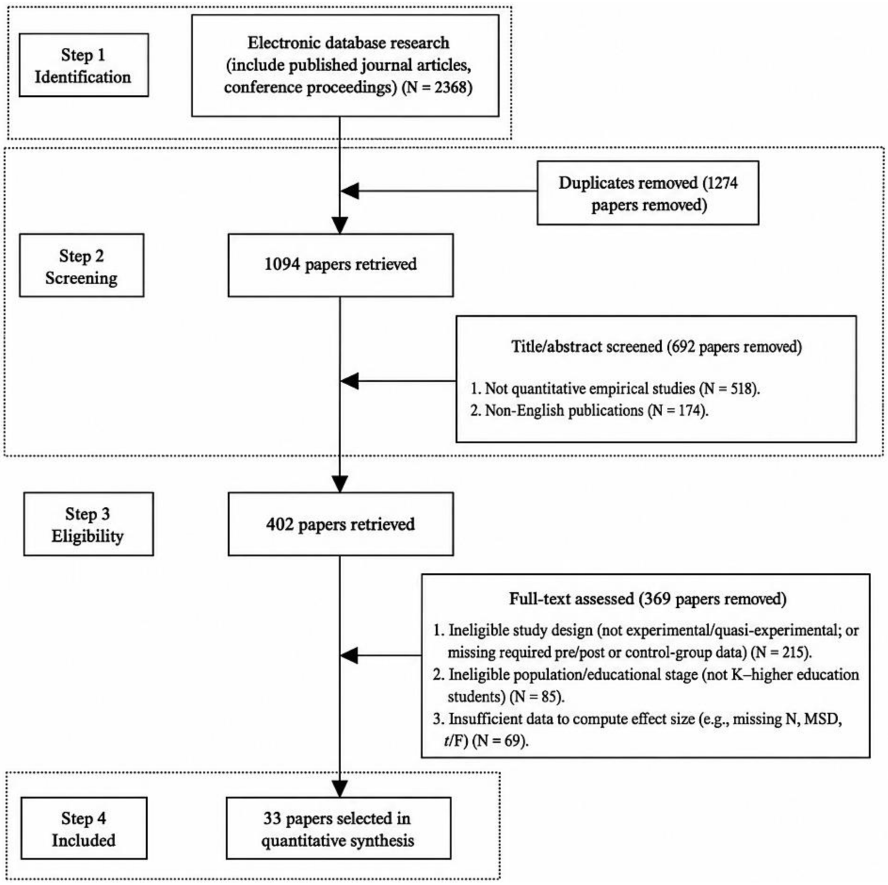

Figure 1. PRISMA-guided flowchart of the study identification and selection procedure

2.2. Coding of literature

For each study, we extracted basic information, including the author, publication year, sample size, gender distribution, intervention characteristics, and study outcomes. After independently completing the initial coding, the two coders assessed intercoder reliability, yielding a Cohen's kappa=0.896, indicating high agreement [22]. Any discrepancies were first resolved through discussion and joint review of the original reports. If consensus could not be reached, a third researcher adjudicated the final coding decision. Based on prior work [16], [17], [20], [23], we classified five moderating variables. Education stage was coded from the participant description (kindergarten; elementary school; secondary school (middle/high school); higher education). Teaching strategy was coded as the dominant instructional approach (problem-based learning: ill-structured problem solving; project-based learning: extended robotics project culminating in a tangible product/performance; design-based learning: iterative design-test-redesign cycles; lecture-based teaching: teacher-led explanation/demonstration followed by practice). Interaction type was coded as individual learning when students mainly worked individually (often with teacher guidance), and as group collaboration when tasks required teamwork with shared goals and joint artefact construction. Intervention duration was coded by total instructional time (≤1 day, >1 day and ≤1 month, or >1 month). Programming method was coded as plugged versus unplugged (screen-free) robot programming: plugged programming used a computer/tablet-based coding environment to create and execute code for robot control, whereas unplugged programming involved constructing and testing command sequences without a computer (e.g., on-device button programming or tangible command formats), with the robot serving as the execution agent.

2.3. Statistical procedures

The meta-analysis was conducted in comprehensive meta-analysis (CMA), version 3.3.0.7 [24]. Hedges' g was used as the effect size index because it corrects for bias in small samples [25]. Based on Cohen [26], effect sizes were interpreted as small (<0.50), moderate (0.50-0.79), and large (≥0.80). Heterogeneity was evaluated using the Q statistic and I^2 [27]. A fixed-effect model was applied when heterogeneity was low ($I^2<75\%$ or $P(Q)>0.05$), whereas a random-effects model was used when heterogeneity was substantial [28]. When heterogeneity was significant, subgroup analyses were performed to explore possible sources of between-study variation [29]. Publication bias was assessed with funnel plots and Egger's regression test [30]. A leave-one-out sensitivity analysis was also performed to examine whether the pooled estimate changed materially after the sequential removal of individual studies [31].

3. RESULTS AND DISCUSSION

3.1. Overall effect size

When merging effect sizes, we first conducted a heterogeneity test. The results ($I^2=79.61\%$, $Q=186.369$, $p=0.000$) indicated significant heterogeneity among the 33 included studies. Therefore, we used a random-effects model to combine the 39 effect sizes from the 33 included studies. Figure 2 shows that the final combined effect size of the 33 included studies was $g=0.621$ (95% CI: [0.512, 0.729], $Z=11.235$, $p<0.001$). This effect size exceeded the 0.50 threshold, indicating that ER-based programming activities have a statistically significant effect on students' CT development, and the effect size is moderate. These results indicate a significant overall positive effect of ER on students' CT, although substantial heterogeneity was observed across studies.

3.2. Publication bias

To further evaluate the stability of the pooled effect, publication bias was examined. We used funnel plots together with Egger's regression test to detect possible publication bias in this study. A total of 39 independent effect sizes were analyzed, and Figure 3 presents the funnel plot. As shown in Figure 3, the effect sizes of most studies were evenly distributed around the mean, indicating little evidence of publication bias. Egger's regression test yielded $t=1.720$ and $p=0.094$ ($p>0.05$), suggesting that no significant publication bias was detected. In summary, the validity of the literature included in this study is high, providing a robust and reliable data basis for subsequent effect analysis.

3.3. Sensitivity analysis

To assess the stability of the pooled result, a leave-one-out sensitivity analysis was conducted by re-estimating the overall effect after omitting one study at a time. Figure 4 presents the results of this procedure. Each estimate remained close to the original pooled effect, and the corresponding confidence intervals showed only slight variation across iterations. These findings indicate that the leave-one-out estimates (Hedges' g=0.605-0.646) remained close to the overall pooled effect (g=0.621), suggesting that no single study unduly influenced the result.

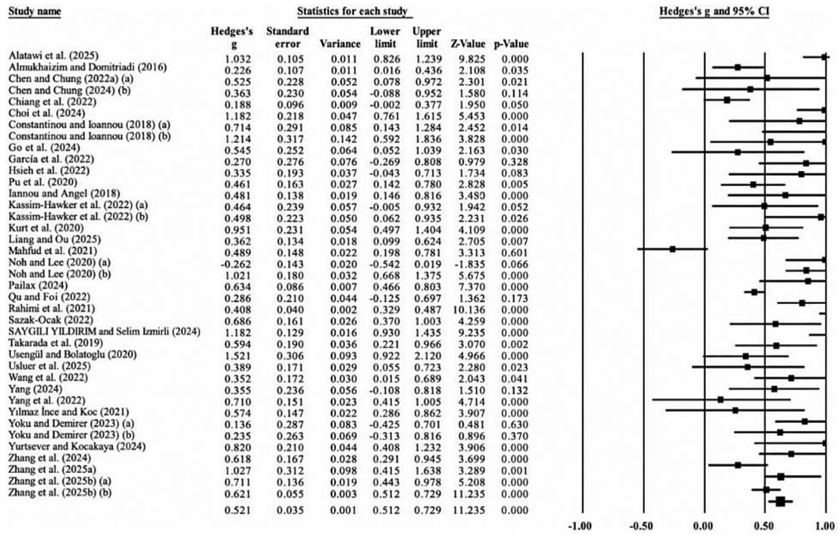

Figure 2. Forest plot of overall effects on students' CT

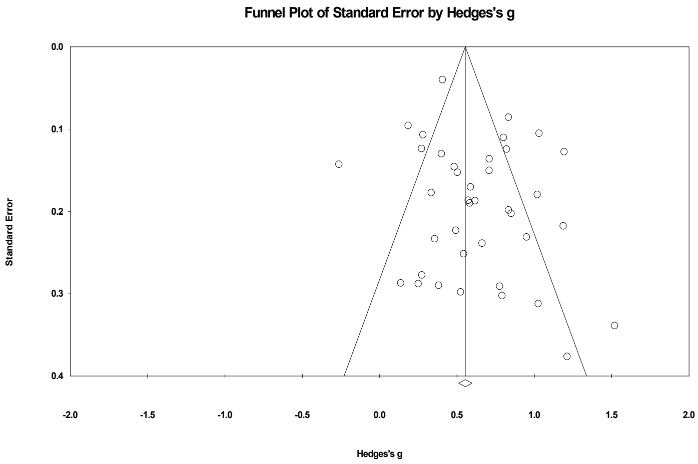

Figure 3. Funnel plot of standard error

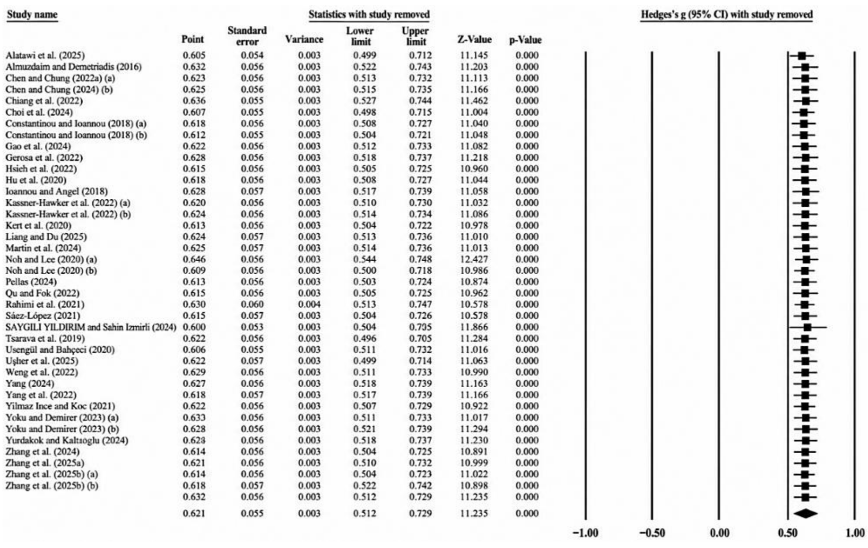

Figure 4. Sensitivity analysis (with one study removed)

3.4. Subgroup analysis

Because of the significant heterogeneity we observed when combining effect sizes in Section 3.1., we performed subgroup analyses on the five variables (education stage, teaching strategies, interaction types, intervention duration, and programming types) to explore potential sources of variation. The criterion was that a p-value greater than 0.10 for a subgroup indicated no significant difference in effect size between subgroups, indicating that the moderator variables were not the source of heterogeneity [28]. Therefore, further analyses are needed to explore other potential moderators that may explain the observed heterogeneity. Specific results are shown in Table 2. The use of ER in programming learning activities showed no significant differences across different education levels, teaching strategies, or interaction types. However, significant differences were observed across different intervention durations and programming types. Specifically, in terms of intervention duration, long-term intervention had the largest effect size ($g=0.669^{***}$), followed by short-term ($g=0.646^{***}$) and medium-term ($g=0.415^{***}$). As for programming methods, plug-in programming had a moderate effect size ($g=0.658^{***}$), while unplugged programming had a small effect size ($g=0.470^{***}$).

Table 2. Subgroup analysis of ER effects on students' CT

Variables	Group	Number of effect sizes (k)	g	95% confidence interval		Z	Q (b)	P(Q)
				Lower	Upper			
Educational stage	Kindergarten	6	0.515***	0.330	0.700	5.448	1.542	0.673
	Elementary school	16	0.622***	0.438	0.807	6.600		
	Middle school	12	0.690***	0.470	0.910	6.152		
	Higher education	5	0.619***	0.412	0.826	5.861		
Teaching strategies	Game-based learning	4	0.711***	0.409	1.014	4.604	0.881	0.927
	Problem-based learning	8	0.581***	0.265	0.897	3.606		
	Project-based learning	21	0.637***	0.497	0.776	8.961		
	Lecture method	2	0.666***	0.354	0.978	4.183		
	Design-based learning	4	0.514**	0.175	0.853	2.969		
Interaction type	Group work	36	0.615***	0.500	0.731	10.470	0.253	0.615
	Individual work	3	0.685***	0.441	0.928	5.508		
Intervention duration	<1 day	2	0.646**	0.267	1.026	3.340	8.122	0.017
	>1 day and ≤1 month	9	0.415***	0.298	0.532	6.968		
	>1 month	28	0.669***	0.533	0.806	9.641		
Programming methods	Plugged programming	32	0.658***	0.524	0.791	9.671	3.756	0.053
	Unplugged programming	7	0.470***	0.336	0.605	6.857		

Note: ***p-values <0.001, **p-values <0.01; Q(b)=between-group heterogeneity statistic; and P(Q)=p value for between-group heterogeneity.

3.5. Discussion

3.5.1. Interpretation of the main findings

This study applied meta-analysis to integrate 33 empirical studies published between 2015 and 2025. The results showed that integrating ER into programming instruction significantly improved students' CT, yielding an overall moderate effect ($g=0.621$, $p<0.001$). Such a result aligns with earlier studies [32], [33] and further supports the educational value of integrating ER into programming learning activities. The subgroup analyses further showed that the beneficial impact of ER on students' CT was evident across different educational stages, teaching strategies, and interaction types, with relatively similar effect sizes. However, intervention duration and programming methods explained part of the variation in the observed effects.

With regard to intervention duration, long-term interventions lasting more than one month produced the strongest effects, whereas short-term interventions lasting less than one day also showed significant benefits. By contrast, interventions lasting from one day to one month showed relatively weaker effects. This pattern suggests that the relationship between intervention duration and CT outcomes may be non-linear. A possible explanation can be drawn from cognitive load theory [34]. Short-term activities often involve clear goals and highly focused tasks, which may quickly attract students' attention and support immediate engagement in CT-related learning [35]. In medium-term interventions, however, task demands may increase before students have developed stable learning strategies, which may reduce the benefits at this stage. Longer interventions, in contrast, provide more time for repeated practice, feedback, and reflection, which may gradually strengthen abstraction, algorithmic thinking, and problem solving [36]. These findings indicate that teachers should pay attention not only to the duration of robotics-supported programming activities, but also to how tasks are sequenced throughout the intervention period. The duration of instruction should therefore be aligned with task complexity, learning objectives, and the level of guidance provided to students [37].

Another important finding was that plugged programming was more effective than unplugged programming in promoting students' CT, which is consistent with previous studies [38]. This difference may be understood from the perspective of embodied cognition [39]. Plugged programming provides direct command-execution-feedback cycles, such as robot movement or sensor responses, which help students connect abstract programming logic with observable outcomes [40]. By contrast, unplugged activities rely more heavily on symbolic reasoning and usually provide less immediate feedback [41]. So, teachers may use unplugged activities only as introductory supports for basic concepts, but instructional design should place greater emphasis on plugged programming when the aim is to promote deeper understanding of algorithmic logic, code execution, and problem solving [42].

3.5.2. Implications for classroom practice

This study offers several practical insights for classroom teaching. ER should be incorporated into regular programming instruction as a planned teaching resource rather than being used only in occasional demonstration lessons or enrichment activities. When the instructional goal is to promote students' CT, robotics-supported programming should be linked to explicit CT objectives and implemented as part of a coherent teaching process. Teachers should also pay attention to how robotics activities are arranged over time. Rather than relying only on isolated robotics sessions, they should organize robotics-supported programming as a continuing sequence of learning tasks so that students have sufficient opportunities to practice, revise, and consolidate CT skills. Notably, when cultivating students' CT skills is the primary objective, plugged programming should be prioritized in classroom implementation. Compared with unplugged programming, plugged environments provide stronger support for helping students connect programming code with observable outcomes and problem-solving processes. Unplugged activities may still be useful during the introductory stage to support basic concepts such as sequencing or decomposition, but they should generally play a supporting role when the goal is to develop deeper CT skills.

3.5.3. Limitations and future research

Several issues should be taken into account when interpreting the findings of this meta-analysis. To begin with, the review was restricted to studies published in English, which means that relevant evidence reported in other languages may not have been captured. In addition, the included studies showed considerable heterogeneity, indicating that the impact of ER on students' CT may vary according to factors not examined in the present analysis. Another constraint concerns the moderator analysis. Because the reporting quality and level of detail differed across primary studies, only five moderators could be coded consistently and compared across studies. Further meta-analytic research may broaden the language scope of the search and investigate additional moderators, including teacher expertise, task complexity, and assessment approaches, when primary studies provide sufficient information.

4. CONCLUSION

This meta-analysis synthesized 33 empirical studies to examine whether robotics-supported programming can enhance students' CT. The findings show that ER can be a meaningful instructional approach for supporting CT development, while intervention duration and programming methods help explain variation in outcomes across studies. These results suggest that educators should integrate ER into regular programming instruction in ways that align activity duration, learning tasks, and instructional goals, and should prioritize programming approaches that more directly support students' engagement with code, feedback, and problem solving when CT is the primary objective. At the policy level, schools and curriculum planners should provide sustained curricular, technical, and professional support so that robotics-supported programming can be implemented purposefully and consistently across classroom settings.

ACKNOWLEDGMENTS

The authors gratefully acknowledge Wenyu Zhang and Lin Xiaofeng, PhD candidates at Universiti Kebangsaan Malaysia for their valuable contributions to the literature coding process in this meta-analysis.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Haipeng Yang	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Helmi Norman	✓	✓								✓	✓	✓	✓	
M. Khalid M. Nasir	✓	✓					✓			✓	✓	✓	✓	
Nian Xu		✓			✓	✓		✓			✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

This study is a meta-analysis and all data analyzed were obtained from previously published studies that are publicly accessible and ethically approved by their respective authors and institutions.

ETHICAL APPROVAL

This study is a meta-analysis conducted following the PRISMA guidelines. It did not involve the collection of primary data from human participants or animals and therefore did not require ethical approval.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [HN], upon reasonable request.

REFERENCES

[1] Y. Yu and Y. Jiang, *Understanding the digital revolution and its influences*. Singapore: Springer Nature Singapore, 2024, doi: 10.1007/978-981-97-4939-3.

[2] R. W. M. Mee, F. S. C. Yob, L. S. Pek, M. F. Abd Rauf, Y. Mingmei, and A. Derahvasht, "Building digital thinkers: a bibliometric analysis of computational thinking in children's education for a sustainable future," *Contemporary Educational Technology*, vol. 17, no. 3, 2025, doi: 10.30935/cedtech/16309.

[3] J. Peng, M. Hao, B. Chen, and M. Wang, "Cultivating computational thinking in a STEM course through the ICDCO instructional model," *Education and Information Technologies*, vol. 30, no. 17, pp. 25137–25175, 2025, doi: 10.1007/s10639-025-13716-0.

[4] E. Xu, W. Wang, and Q. Wang, "A meta-analysis of the effectiveness of programming teaching in promoting K-12 students' computational thinking," *Education and Information Technologies*, vol. 28, no. 6, pp. 6619–6644, 2023, doi: 10.1007/s10639-022-11445-2.

[5] Z. Mamatnabiyev, C. Chronis, I. Varlamis, Y. Himeur, and M. Zhaparov, "A holistic approach to use educational robots for supporting computer science courses," *Computers*, vol. 13, no. 4, 2024, doi: 10.3390/computers13040102.

[6] L. Greifenstein, I. Graßl, U. Heuer, and G. Fraser, "'Help me solve it' or 'solve it for me': effects of feedback on children building and programming robots," in *SIGCSE 2024 - Proceedings of the 55th ACM Technical Symposium on Computer Science Education*, 2024, vol. 1, pp. 401–407, doi: 10.1145/3626252.3630752.

[7] A. Badeleh, "The effects of robotics training on students' creativity and learning in physics," *Education and Information Technologies*, vol. 26, no. 2, pp. 1353–1365, 2021, doi: 10.1007/s10639-019-09972-6.

[8] C. H. Chen and H. Y. Chung, "Fostering computational thinking and problem-solving in programming: integrating concept maps into robot block-based programming," *Journal of Educational Computing Research*, vol. 62, no. 1, pp. 406–427, 2024, doi: 10.1177/07356331231205052.

[9] S. Bai and P. Tian, "Educational robotics may enhance students' conceptual knowledge, applied skills, and learning attitude in STEM education: A meta-analysis," *Educational Technology and Society*, vol. 28, no. 4, pp. 274–303, 2025, doi: 10.30191/ETS.202510_28(4).SP06.

[10] F. A. Nannim, N. E. Ibezim, G. C. Agbo, C. Mgboji, S. O. R. Ngwoke, and M. Mosia, "Development of a project-based arduino learning app: fostering robotics programming competence among preservice teachers of computer and robotics education," *ACM Transactions on Computing Education*, vol. 25, no. 1, 2025, doi: 10.1145/3719016.

[11] A. V. Pou, X. Canaleta, and D. Fonseca, "Computational thinking and educational robotics integrated into project-based learning," *Sensors*, vol. 22, no. 10, 2022, doi: 10.3390/s22103746.

[12] S. Y. Yang, Y. C. Lin, and Y. T. Lin, "Improving elementary students' computational thinking skills through an educational robot intervention: a quasi-experimental study," *International Journal of Learning, Teaching and Educational Research*, vol. 23, no. 9, pp. 343–360, 2024, doi: 10.26803/ijlter.23.9.18.
[13] V. Yolcu and V. Demirer, "The effects of educational robotics in programming education on students' programming success, computational thinking, and transfer of learning," *Computer Applications in Engineering Education*, vol. 31, no. 6, pp. 1633–1647, 2023, doi: 10.1002/cae.22664.
[14] C. Weng, I. M. Matere, C. H. Hsia, M. Y. Wang, and A. Weng, "Effects of LEGO robotic on freshmen students' computational thinking and programming learning attitudes in Taiwan," *Library Hi Tech*, vol. 40, no. 4, pp. 947–962, 2022, doi: 10.1108/LHT-01-2021-0027.
[15] J. Noh and J. Lee, "Effects of robotics programming on the computational thinking and creativity of elementary school students," *Educational Technology Research and Development*, vol. 68, no. 1, pp. 463–484, 2020, doi: 10.1007/s11423-019-09708-w.
[16] Y. Zhang and Y. Zhu, "Effects of educational robotics on the creativity and problem-solving skills of K-12 students: a meta-analysis," *Educational Studies*, vol. 50, no. 6, pp. 1539–1557, 2024, doi: 10.1080/03055698.2022.2107873.
[17] F. Ouyang and W. Xu, "The effects of educational robotics in STEM education: a multilevel meta-analysis," *International Journal of STEM Education*, vol. 11, no. 1, 2024, doi: 10.1186/s40594-024-00469-4.
[18] H. Woo, G. K. LeTendre, T. Pham-Shouse, and Y. Xiong, "The use of social robots in classrooms: a review of field-based studies," *Educational Research Review*, vol. 33, 2021, doi: 10.1016/j.edurev.2021.100388.
[19] G. Nugent, B. Barker, N. Grandgenett, and V. I. Adamchuk, "Impact of robotics and geospatial technology interventions on youth STEM learning and attitudes," *Journal of Research on Technology in Education*, vol. 42, no. 4, pp. 391–408, 2010, doi: 10.1080/15391523.2010.10782557.
[20] J. Wang, W. Fan, and J. Yu, "Comparing the effects of unplugged activities and plugged activities on the development of students'computational thinking: a meta-analysis," *Educational Technology Research and Development*, vol. 73, no. 5, pp. 3247–3267, 2025, doi: 10.1007/s11423-025-10522-w.
[21] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, "Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement," *Journal of clinical epidemiology*, vol. 62, no. 10, pp. 1006–1012, 2009, doi: 10.1016/j.jclinepi.2009.06.005.
[22] M. J. Page *et al.*, "PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews," *The BMJ*, vol. 372, 2021, doi: 10.1136/bmj.n160.
[23] H. Lee and J. H. Lee, "The effects of robot-assisted language learning: a meta-analysis," *Educational Research Review*, vol. 35, 2022, doi: 10.1016/j.edurev.2021.100425.
[24] M. Borenstein, "Comprehensive meta-analysis software," in *Systematic Reviews in Health Research*, M. Egger, J. P. T. Higgins, and G. D. Smith, Eds. Wiley, 2022, pp. 535–548, doi: 10.1002/9781119099369.ch27.
[25] L. Ling and M. A. Ariel, "Evaluation of various estimators for standardized mean difference in meta-analysis," *Statistics in medicine*, vol. 40, no. 2, pp. 403–426, 2021.
[26] J. Cohen, "A power primer," in *Methodological issues and strategies in clinical research*. 4th ed., A. E. Kazdin, Ed. Washington: American Psychological Association, 2016, pp. 279–284, doi: 10.1037/14805-018.
[27] G. J. Choi and H. Kang, "Heterogeneity in meta-analyses: an unavoidable challenge worth exploring," *Korean Journal of Anesthesiology*, vol. 78, no. 4, pp. 301–314, 2025, doi: 10.4097/kja.25001.
[28] M. Borenstein, L. V. Hedges, J. P. T. Higgins, and H. R. Rothstein, *Introduction to meta-analysis*. Chichester, UK: John Wiley & Sons, Ltd., 2009, doi: 10.1002/9780470743386.
[29] J. P. T. Higgins, J. Thomas, J. Chandler, M. Cumpston, T. Li, M. J. Page, and V. A. Welch, *Cochrane Handbook for Systematic Reviews of Interventions*, 2nd ed. Chichester, UK: John Wiley & Sons, Ltd., 2019, doi: 10.1002/9781119536604.
[30] H. R. Rothstein and B. J. Bushman, "Publication bias in psychological science: Comment on Ferguson and Brannick (2012)," *Psychological Methods*, vol. 17, no. 1, pp. 129–136, 2012, doi: 10.1037/a0027128.
[31] N. A. Patsopoulos, E. Evangelou, and J. P. A. Ioannidis, "Sensitivity of between-study heterogeneity in meta-analysis: proposed metrics and empirical evaluation," *International Journal of Epidemiology*, vol. 37, no. 5, pp. 1148–1157, 2008, doi: 10.1093/ije/dyn065.
[32] L. Hong, "The impact of educational robots on students' computational thinking: a meta-analysis of K-12," *Education and Information Technologies*, vol. 29, no. 11, pp. 13813–13838, 2024, doi: 10.1007/s10639-023-12415-y.
[33] M. Usher, N. Reznik, G. Bronshtein, and D. Kohen-Vacs, "Bridging virtual and physical: exploring students' computational thinking and creativity in robot-guided vs. simulation-based learning," *Journal of Learning Analytics*, vol. 12, no. 2, pp. 35–49, 2025, doi: 10.18608/jla.2025.8837.
[34] J. Sweller, "Cognitive load during problem solving: effects on learning," *Cognitive Science*, vol. 12, no. 2, pp. 257–285, 1988, doi: 10.1207/s15516709cog1202_4.
[35] J. Wirth, J. Künsting, and D. Leutner, "The impact of goal specificity and goal type on learning outcome and cognitive load," *Computers in Human Behavior*, vol. 25, no. 2, pp. 299–305, 2009, doi: 10.1016/j.chb.2008.12.004.
[36] T. Sapounidis and D. Alimisis, "Educational robotics for STEM: A review of technologies and some educational considerations," *Science and Mathematics Education for 21st Century Citizens: Challenges and Ways Forward*, pp. 167–190, 2020.
[37] S. Kucuk and B. Sisman, "Pre-service teachers' experiences in learning robotics design and programming," *Informatics in Education*, vol. 17, no. 2, pp. 301–320, 2018, doi: 10.15388/infedu.2018.16.
[38] B. Zurnacı and Z. Turan, "Educational robotics or unplugged coding activities in kindergartens?: comparison of the effects on pre-school children's computational thinking and executive function skills," *Thinking Skills and Creativity*, vol. 53, 2024, doi: 10.1016/j.tsc.2024.101576.
[39] M. Wilson, "Six views of embodied cognition," *Psychonomic Bulletin and Review*, vol. 9, no. 4, pp. 625–636, 2002, doi: 10.3758/BF03196322.
[40] O. Mubin, C. J. Stevens, S. Shahid, A. Al Mahmud, and J.-J. Dong, "A review of the applicability of robots in education," *Technology for Education and Learning*, vol. 1, no. 1, 2013, doi: 10.2316/journal.209.2013.1.209-0015.
[41] F. J. Agbo, L. O. Okpanachi, P. Ocheja, S. S. Oyelere, and G. Sani, "How can unplugged approach facilitate novice students' understanding of computational thinking? An exploratory study from a Nigerian university," *Thinking Skills and Creativity*, vol. 51, 2024, doi: 10.1016/j.tsc.2023.101458.
[42] M. Chevalier, C. Giang, A. Piatti, and F. Mondada, "Fostering computational thinking through educational robotics: a model for creative computational problem solving," *International Journal of STEM Education*, vol. 7, no. 1, 2020, doi: 10.1186/s40594-020-00238-z.

BIOGRAPHIES OF AUTHORS

Haipeng Yang is a PhD candidate at the Faculty of Education, Universiti Kebangsaan Malaysia, specializing in Learning Technology and Innovation. He holds a Bachelor of Science in Computer Science and a Master of Science in Educational Technology. His research interests include educational robotics, programming education, and the application of artificial intelligence in teaching and learning. His academic work aims to promote innovative and equitable digital education through the integration of emerging technologies and evidence-based pedagogical practices. He can be contacted at email: p136759@siswa.ukm.edu.my.

Helmi Norman is deputy director (Instructional Technology) at the Centre of Innovation in Learning and Teaching, Faculty of Education, Universiti Kebangsaan Malaysia. He holds a Ph.D. in Resources and IT. His expertise lies in digital learning and futuristic education, particularly in the design, implementation, and evaluation of emerging technologies in teaching and learning. He has been involved in pioneering work on instructional technology, digital pedagogy, and the development of online and blended learning frameworks in higher education. He can be contacted at email: helmi.norman@ukm.edu.my.

M. Khalid M. Nasir is a senior lecturer and media coordinator at the Centre of Innovation in Learning and Teaching, Faculty of Education, Universiti Kebangsaan Malaysia. He holds a Ph.D. in ICT and Human Resources from Colorado State University, USA. His expertise includes computer education, educational technology, instructional innovation, and the community of inquiry (CoI) framework in online learning. He has extensive experience in research and development projects focused on digital pedagogy, online learning models, and technology integration in higher education. He can be contacted at email: mdkhalid@ukm.edu.my.

Nian Xu is a lecturer at the Faculty of Educational Science, Kashgar University, China. He holds a bachelor's degree in Preschool Education and a master's degree in Curriculum and Instruction. His research focuses on educational technology integration, digital pedagogy, and instructional innovation in teacher education. He has participated in several research projects related to technology-enhanced learning, curriculum reform, and the development of teachers' digital competence in western China. He can be contacted at email: xuniankashi@gmail.com.