Università degli Studi di Padova

Head Office: University of Padua

Department of General Psychology

Ph.D. COURSE IN: Brain, Mind and Computer Science

CURRICULUM: Neuroscience, Technology and Society

SERIES: XXXVI

THE COGNITIVE EFFECTS OF COMPUTATIONAL THINKING

Coordinator: Prof. Anna Spagnolli

Supervisor: Prof. Barbara Arfé

Co-Supervisor: Prof. Tullio Vardanega

Ph.D. student: Dr. Chiara Montuori

«E come tutte le più belle cose
Vivesti solo un giorno, come le rose»

A Susanna, che è dentro di voi,
Beatrice e Margherita,
che perseverate in un dolce e raro legame,
tenendovi sempre per mano
con il sole e con la tempesta.

TABLE OF CONTENTS

ABSTRACT

In Europe, a significant percentage of 13-year-old students lacks essential digital and problem-solving skills that are crucial for contemporary societies. Computational Thinking (CT), integrated into school curricula in various countries, emerges as a promising educational program to cultivate 21st-century skills, including digital literacy and analytical thinking. CT involves a problem-solving approach inspired by computer science, encouraging the breakdown of complex problems into manageable parts and fostering systematic thinking. This thesis focuses on coding, which is a component of CT that involves creating, modifying, and evaluating program text, symbols, and familiarity with programming concepts. Teaching tools for CT include unplugged coding, educational robotics (ER), and plugged/virtual coding.

While the integration of CT into schools is widely acknowledged, research on the cognitive benefits of CT and coding interventions, especially through randomized trials, is still limited. Existing studies suggest positive effects on computational thinking, problem-solving, and executive functions abilities. However, non-experimental or quasi-experimental studies underpin most of the existing literature. This call for a need of more robust evidence. The studies presented in the present thesis aim to address this gap by examining the consistency and generalizability of effects across different executive functions and age groups.

Starting from the results of previous studies, the present thesis investigated the cognitive effects of CT interventions on children of different age (range from 4 to 16 years), with the aim of exploring the cognitive effects of computational thinking.

We first conducted a systematic review and meta-analytic study on the cognitive effects of computational thinking, *Study 1* was a review of the effectiveness of CT intervention on the development of core and higher order executive functions, trying to gain a comprehensive understanding of the cognitive effects of computational thinking, with a specific focus on the age range 4-16 years.

Second, we explored how the gains of preschoolers in coding skills following an intervention based on a combination of unplugged coding and educational robotics transfer to plugged (computer-based) coding abilities and to EFs such as planning, response inhibition, and visuo-spatial skills. *Study 2* investigated the effects of tangible coding intervention on preschoolers' EFs and visuo-spatial skills through a cluster randomized controlled trial. Forty-seven (47) preschoolers from 4 class groups, with no prior exposure to coding, were randomly assigned to an experimental (unplugged coding and ER, 2 classes) or control (standard school activities, 2 classes) instructional condition.

Third, we tested the age-related effects of coding interventions, *Study 3* addressed this research question. Four-hundred thirty-seven (437) primary school children, attending the first year or the fourth year of primary school, from 5 schools from different socio-economic status (SES) participated in the study. None of the children had been previously exposed to coding. Within each age group (first or fourth grade) children were assigned to a treatment and a control group. The first-grade group comprised 273 children: 128 assigned to the treatment condition and participating in coding labs immediately after the pretest (T1) and 145 controls, assigned to standard STEM activities (math and technology) and receiving the coding intervention only after the posttest (T2).

Although the teaching of CT is compulsory, we still know very little about the effectiveness of CT programs and the cognitive functions these programs work best on. Consequently, recommendations for instructional practice are also lacking. Testing the effectiveness of CT intervention may help develop such recommendations and identify the best instructional tools and programs for teaching coding from an early age. Overall, the results of this dissertation provide insight into the specific cognitive skills that are most affected by CT interventions. In addition, the research results contribute to identifying coding tools that are suitable for learning CT concepts and developing coding skills at a young age.

INTRODUCTION

In Europe, one in three 13-year-old students lacks basic digital skills and problem-solving skills needed in modern societies (OECD, 2015; PISA 2017). Computational Thinking (CT), considered a digital skill and problem-solving ability that is part of school curricula in several countries, is a promising educational program to develop the 21st-century skills (e.g., digital literacy, creative problem solving, analytical thinking).

CT is characterized by a problem-solving approach inspired by computer science, encourages students to break down complex problems into smaller, manageable parts, analyze patterns, and develop algorithms, fostering a systematic and organized approach to problem-solving (Wing, 2006). Coding, including the skills to create, modify, and evaluate program text, fragments, symbols, and the familiarity with programming concepts and procedures, is one of the instruments of CT skills and as such will also be a focus of this thesis. Tools used to teach CT skills and coding during the early school years are unplugged coding, meaning programming without digital devices, educational robotics (ER), which is giving instructions to a programmable robot to perform specific actions in a physical environment (Chen et al., 2017), and plugged/virtual coding, i.e., developing computer program to teach a computer, or a virtual character, to achieve a goal in a virtual environment (Arfé et al., 2019; Fessakis et al., 2013; Kalelioglu, 2015; Zhang et al., 2014).

Nowadays, little doubts exist about the need to introduce CT into schools as a central tool for developing students' digital literacy (Nardelli & Ventre, 2015; Resnick, 2007; Roman-Gonzalez et al., 2017). Consequently, last years have witnessed a growing body of research focused on CT and its integration into classroom instruction through coding (e.g., Fessakis et al., 2013; Israel et al., 2015; Lye & Koh, 2014; Saez-Lopez et al., 2016; Shute et al., 2017). However, the benefits of these activities in terms of cognitive skills development have been only explored through few empirical studies.

Even though a substantial number of studies reported that learning to code could improve computational thinking (e.g., Lye & Koh, 2014; Tran, 2018; Webb, 2010; Werner et al., 2012), and problem solving (Tonbuloglu, 2019; Shim et al., 2017; Fessakis et al., 2013), most of them were not randomized trials (RCT) that can provide strong causal evidence. Just three RCT studies on first graders show that educational robotics boosts children's working memory and inhibition skills (Di Lieto et al., 2020) and coding can significantly enhance planning and inhibition skills (Arfé et al., 2019). Although there is evidence from recent experimental studies (e.g., Arfé et al., 2019; Arfé et al., 2020; Di Lieto et al., 2020a) that teaching CT leads to children's improved EF skills, there is still uncertainty about the consistency, robustness, and generalizability of these effects across different (higher order and core) executive functions and across different age.

Computational thinking extends beyond coding and programming to include a wider range of problem-solving strategies that can be applied in various contexts. By practicing CT, children can improve their problem-solving abilities and may also exercise their core executive functions. Indeed, as core executive function skills are involved in problem-solving, engaging in CT practices can influence these core EFs as well. By actively practicing and enhancing these higher-order skills through CT, individuals may experience improvements in their core EF components. Therefore, it is plausible to hypothesize that the practice and enhancement of higher-order EF skills through CT may lead to observable improvements in the core EF components upon which they rely.

Examining the effects of coding interventions on EFs skills is important not only to gain a better understanding of the cognitive effects of coding, but also because core and higher order EFs underpin academic performance and adaptive behaviors and thus effects on these cognitive skills could likely transfer to more complex abilities and tasks. Preschool years are a critical period for the development of brain regions that subserve EFs abilities and, as such, might constitute a crucial developmental period to target the malleability of EFs (Ganesan & Steinbeis, 2022; Zelazo & Carlson, 2012). Therefore, interventions aimed at enhancing EFs can be especially effective within this time window (Scionti et al., 2020), which can be also the perfect span to assess the early cognitive effects of coding.

Whereas research on the implementation of coding in preschool and primary school is growing rapidly, the lack of experimental studies testing the effectiveness of these activities on preschoolers' cognitive development represents a significant gap in the research field. The studies conducted on preschoolers have so far only demonstrated the feasibility of teaching coding at this early age (Bers et al., 2014; Ching & Hsu, 2023). Very few experimental studies have instead tested the effectiveness of coding on preschoolers' cognitive development (Çakır et al., 2021; Çiftci & Bildiren, 2020; Nam et al., 2019), focusing on problem solving which is a complex cognitive skill.

In Italy, teaching CT from preschool became compulsory with a law issued by the Ministry of Education (motion n. 1-00117, 12 March 2019). However, to date the teaching of CT in Italian preschool and primary school is far from being homogeneous, as many teachers lack the fundamental training to achieve this goal. Introducing CT from preschool years is of primary importance, as it equips students with skills and mindsets necessary to thrive in the modern world, such as digital skills, logical reasoning, and creativity (Shute et al., 2017). Early exposure to CT lays a strong foundation for future learning in various disciplines, including Science Technology Engineering Mathematics (STEM) field (Leonard et al., 2016).

Considering the importance of enhancing executive functions (EFs) in childhood and the increasing diffusion of CT programs in schools, it is also crucial to investigate how age (and cognitive development with it) impacts on the children response to these interventions. Testing the age-related differences in the effectiveness of CT intervention is crucial for several reasons. Different age groups may respond differently to coding interventions. Understanding age-specific impacts help target interventions to meet the specific needs of each age group. In addition, identifying the age at which coding interventions are most effective allows educators and policymakers to optimize instructional strategies. This knowledge could improve the design of curricula and educational approaches, making them age-appropriate and more effective.

To date, RCT studies investigating age-related differences in the effectiveness of CT interventions on EFs and coding skills are lacking. The findings of this thesis project will contribute to fill this research gap and clarify the cognitive effects of CT.

Firstly, by tacking stock of the existing literature through a systematic review and meta-analytic study. Secondly, through randomized controlled trials, which investigated the effects of coding activities on the development of executive functions and coding abilities at different age levels: (a) preschool children and (b) first and fourth graders.

The PhD overall project is presented and discussed in this thesis dissertation in four chapters organized as following: chapter one will present a systematic review and meta-analytic study on the cognitive effects of computational thinking in children and adolescents. The SR and MA was carried out to take stock of the existing literature in the field, which provides the basis to the following experimental studies. The aim of the meta-analytic study was to synthesize the effectiveness of coding on the cognitive abilities of 4- to 16-year-old children and make an overview of the results of the existing randomized controlled trial (RCT) studies in the literature (Study 1). In the second chapter, an intervention study investigating the effectiveness of tangible coding on the cognitive development of preschoolers will be presented (Study 2). The study explored the effects of a CT intervention that combined unplugged coding and educational robotics (ER) on coding, visuo-spatial, and EFs skills.

In the third chapter, a randomized controlled trial testing children’s response to plugged coding intervention at different age level will be presented (Study 3). Four-hundred thirty-seven (437) primary school children, attending the first year or the fourth year of primary school, from 5 schools and 25 class groups from different socio-economic status (SES) areas of northern Italy participated in the study 3.

Finally, chapter four will present and discuss the general outcomes of the PhD project, their educational implications and future perspectives.

CHAPTER 1. THE COGNITIVE EFFECTS OF COMPUTATIONAL THINKING: A SYSTEMATIC REVIEW AND META-ANALYTIC STUDY[1]

Computational thinking (CT) is the mental ability to apply the concepts and reasoning typical of computing and computer science to solve problems (Wing, 2006). Thinking computationally entails developing four main component skills (a) the ability to analyze problems and decompose them into elements or parts (analytical thinking); (b) the ability to plan a sequence of actions or steps to get to the problem solution (algorithmic thinking); (c) the ability to monitor and correct errors in the execution of the plan (debugging, Flórez et al., 2017; Román-González et al., 2017; Strawhacker & Bers, 2019); and (d) the ability to identify the most relevant aspects of the problem and generalizable algorithms (abstraction), which allow applying to other problems what has been learned (Moreno-León et al., 2016; Román-González et al., 2017; Yaşar, 2018).

Since computational thinking encompasses not only proficiency in computer science methods but also domain-general problem-solving abilities, such as analyzing problems and planning, the development of computational thinking skills intersects with that of 21st-century competencies such as digital literacy skills, including programming or coding, as well as with foundational cognitive skills like analytical thinking, planning, and the ability to inhibit impulsive responses (Arfé et al., 2020). The growing spread of CT and programming into compulsory education (Zhang & Nouri, 2019) is thus supported by the idea that learning to think computationally is important not only for preparing students in the field of computer science (Nardelli, 2019; Wing, 2006) but also, more broadly, for providing them with a general cognitive toolkit to approach and solve everyday problems (Chen et al., 2017; Feurzeig & Papert, 2011; Nardelli, 2019; Wing, 2006).

[1] The present chapter has been published in Computers & Education in the form of the following research paper: Montuori, C., Gambarota, F., Altoé, G., & Arfé, B. (2024). The cognitive effects of computational thinking: A systematic review and meta-analytic study. *Computers & Education*, 210, 104961. https://doi.org/10.1016/j.compedu.2023.104961

Computer programming and code writing are two means of CT and two instrumental skills through which CT is taught and practiced in schools (Lye & Koh, 2014). They include the skills to create, modify, and evaluate codes and the knowledge about programming concepts and procedures. In school settings, children are commonly taught CT through educational robotics (ER), and virtual coding activities, which both involve programming and coding by means of technologies, or through unplugged coding without the use of technology. ER consists in developing a program or code string to give instructions to a robot, so that it can perform specific actions or achieve goals in a physical environment (Di Lieto et al., 2020; Chen et al. 2017; Keren & Fridin, 2014). Unplugged coding consists of tangible, paper and pencil or physical activity, by which children learn the basic concepts and procedures of CT and programming, whereas virtual coding involves the development of a computer program to teach a computer, or a virtual character, to achieve a goal in a virtual environment (Arfé et al., 2019; Fessakis et al., 2013; Kalelioglu, 2015; Zhang et al., 2014). Although in this systematic review and meta-analysis we will refer to programming and coding as two means through which CT is introduced and taught in schools, this does not imply assuming that coding and programming equal computational thinking nor that they always involve it. However, when children exposed to CT interventions in school learn to develop programs for "instructing" a computer, another child or a robot to solve a problem, they also learn to solve problems themselves, articulate and clarify their thoughts and plans, generate a clear sequence of commands (the code) and test their hypotheses, processes that can stimulate the development of other cognitive and metacognitive skills (Clements & Nastasi, 1999; Fessakis et al., 2013).

Past studies have reported positive cognitive effects of these CT activities especially in the domain of children's executive functions (EF, Arfé et al., 2019; Di Lieto et al., 2020a). EFs are a complex set of cognitive skills related to goal setting and the performance of goal-directed behaviors, and thus crucial for self-regulation and academic performance. Although models of EF may differ in the EF skills they focus on, there is general consensus on the distinction between core EFs (working memory, response inhibition and shifting or cognitive flexibility), and complex or higher-order EF skills, like planning and problem solving (Diamond & Ling, 2016). Children's command over, or inhibition, of impulsive responses, their working memory (WM) capacities, encompassing their ability to monitor and update temporarily stored information, and their cognitive flexibility which entails switching perspectives, shifting attention between mental sets or tasks, and readily adjusting behavioral responses to different tasks or environments represent core executive functions (Anderson, 2002; Gioia et al., 2000; Miyake et al., 2000) from which higher-order executive functions, such as planning, problem-solving, and reasoning develop (Diamond & Ling, 2016; Miyake et al., 2000; Thayer & Lane, 2000). Welsh & Pennington (1988), for instance, focusing on higher order EF skills, describe executive functioning as the ability to maintain an appropriate problem-solving set for achieving a goal, and define it as a complex skill set that encompasses not only planning skills and a mental representation of the task and of the outcome, but also inhibitory control, that is, the capacity to inhibit or defer prepotent responses.

As higher order EF skills like problem solving involve core EFs, practicing and enhancing these higher order skills through computational thinking (CT) can impact on their core EF components too. In addition, since core EFs also underpin the ability to achieve academic outcomes in reading (Meixner et al., 2019; Nouwens et al., 2021), writing (Altemeier et al., 2008; Salas & Silvente, 2020), and mathematics (Welsh et al., 2010), improvements in problem solving skills could have transfer effects on academic skills (Scherer et al., 2019).

Although recent experimental studies provide evidence of a causal link between the teaching of CT and the improvement of EF skills in children (e.g., Arfé et al., 2019; Arfé et al., 2020; Di Lieto et al., 2020a), there remains uncertainty regarding the consistency, robustness, and generalizability of these effects across different (higher order and core) executive functions. Moreover, it remains unclear whether the different instructional tools employed to teach CT skills, such as educational robotics (ER), virtual coding or unplugged coding, yield different effects. At this early stage of research, there are very few experimental studies that have directly compared the efficacy of these different types of intervention (e.g., Çınar & Tüzün, 2020). Thus, the only way to compare the efficacy of these types of CT programs is through systematic reviews or meta-analyses, which result now urgent because several countries are currently integrating CT into their school curriculum. Discussing evidence coming from experimental studies is important to allow educators and policymakers to make informed instructional decisions.

Attempts to conduct rigorous systematic reviews or meta-analyses on the cognitive effects of CT have been limited. What is even more important is that, to the best of our knowledge, none of the existing systematic reviews or meta-analyses have examined the causal relationship between CT and executive functions. Most of the current systematic reviews and meta-analyses have focused on one hand on studies that did not allow to draw robust inferences on causal effects, on the other hand they focused on a rather large and heterogeneous set of cognitive functions and academic skills (see Liao, 2000; Liao & Bright, 1991; Scherer et al., 2019; Scherer et al., 2020). The present study contributes to fill this literature gap and advance our understanding of the beneficial effects of CT on executive functions, a specific set of cognitive skills crucial for academic performance and adaptive behavior. This investigation can increase our theoretical comprehension of CT, as observing transfer effects between different domains, like CT and EFs, provides insights into the shared cognitive foundations that underpin their connection. Determining the impact of CT interventions on children's EF has also practical implications. By identifying the executive functions that are most influenced by computational thinking (CT), we can determine the optimal age or grade level for implementing CT programs within the school curriculum. Indeed, different EF skills have different windows of plasticity (e.g., Davidson et al., 2006; Ganesan & Steinbeis, 2022).

1.1 The link between CT and Cognitive Abilities

Recent years have witnessed a steady increase of studies exploring the association between CT and a wide range of cognitive abilities (e.g., Arfé et al., 2019; Finke et al., 2022; Polat et al., 2021; Román-González et al., 2017). Research is divided into correlational and intervention studies.

Correlational studies identify an association between CT and cognitive functions, but only intervention studies can bring causal evidence on the cognitive effects of CT.

Unfortunately, most studies in this area are correlational. They have documented associations between CT and a wide range of cognitive and academic skills, including nonverbal intelligence (Marinus et al., 2018), visuo-spatial skills (Finke et al., 2022; Román-González et al., 2017; Tsarava et al., 2019), and mathematical cognition and mathematical skills (Gerosa et al., 2021; Liu et al., 2019; Román-González et al., 2018; Tsarava et al., 2019). The correlations observed are generally moderate (e.g., Román-González et al., 2017) and in some cases weak (e.g., Román-González et al., 2018). Stronger associations are instead reported between CT and problem-solving (Polat et al., 2021; Román-González et al., 2017). The main limitation of these findings is that although correlational studies are sometimes discussed as if they show causality, causality cannot be determined by correlations.

To investigate the causal effects of CT, researchers have carried out intervention studies (see Arfé et al., 2019, 2020; Di Lieto et al., 2020a). In general, these studies report positive effects of ER and virtual coding activities on first graders' EFs, specifically working memory (Di Lieto et al., 2020a), inhibitory control (Arfé et al., 2019, 2020, Di Lieto et al., 2020a), and planning (Arfé et al., 2020). However, findings vary between studies, particularly when higher order EFs, like problem-solving, are considered. For instance, Çakır et al. (2021) and Nam et al. (2019) found that experiencing coding improved preschoolers and first graders' problem solving, whereas Çiftci and Bildiren (2020) reported insignificant effects of coding on the problem-solving skills of 4–5-year-old preschool children.

One of the problems in comparing the results of these studies is, however, that the instructional methods and tools used in these interventions were different: to train children's problem solving skills, Çakır et al. (2021) and Nam et al. (2019) used ER, whereas Çiftci and Bildiren (2020) employed game-based drag-and-drop exercises from the code.org platform, https://code.org/ , that is, virtual coding. In ER interventions, children are engaged in tangible hands-on coding activities, and they materially interact with robots designing, assembling, and programming them to perform actions in a physical learning environment (Kazakof & Bers, 2014). This experience is more closely related to children's everyday sensorimotor and concrete experience of the world than virtual coding, which involves children learning to program sprites (virtual objects) to perform actions in a virtual world presented on a screen, typically using block-based visual programming and a computing device (e.g., Kalelioğlu, 2015; Sáez-López et al., 2016). This second learning experience implies greater abstraction and perspective-taking skills: to drive the sprite toward the objective, the child must take the perspective of the virtual character on the bi-dimensional screen. From the age of 3 to 6, children's reasoning and problem solving is more closely related to their concrete experience of objects and the physical world (Barrouillet, 2015; Ping & Goldin-Meadow, 2008), but with the beginning of schooling they more often use mental representations to test their hypotheses and thus, their problem solving likely becomes more abstract (Novack et al., 2014). These developmental changes may affect also their ability to learn CT concepts through different tools, allowing more or less tangible or concrete experiences.

Another dimension across which CT intervention programs vary is the degree to which coding or programming activities are structured (Lee et al., 2013; Socratous & Ioannou, 2021). Some instructional interventions ask children to solve structured computational problems, often consisting in logic games with a clear and sole correct solution (e.g., code.org, Arfé et al., 2019; Çiftci & Bildiren, 2020), whereas other interventions focus on unstructured and ill-defined problems (e.g., Erol & Çırak, 2022) such as creating a story or designing a videogame. These problems involve several possible solutions and steps (Zhang & Nouri, 2019).

It is unclear whether these different instructional methods have similar effectiveness on the development of children's EFs and whether it depends on children's age. Although all EFs have an extended window of developmental plasticity that span from early childhood to late adolescence, core and higher order EFs have different developmental trajectories and developmental peaks. Consequently, they could also be more sensitive to the effects of interventions during different time periods. The most intensive changes in core EF, inhibitory control, working memory and cognitive flexibility, occur during the preschool period, from 3 to 6 years (Carlson, 2005; Scionti et al., 2020; Traverso et al., 2015) and in the transition to elementary school (Garon et al., 2008;Macdonald et al., 2014; Zelazo et al., 2003), although they continue to develop until adolescence (Brocki & Bohlin, 2004; Huizinga et al., 2006)

For higher order EFs, such as planning, the greatest developmental changes occur later. Planning, for example, begins to develop between the ages of 5 and 6 (Usai et al., 2014), with a steep developmental growth curve from the age of 6 to 9 (McGuckian et al., 2023) and a second remarkable developmental shift from the age of 9 to 15-17 years, related to the development of prefrontal regions (Luciana et al., 2009). Stimulating the child's EFs through CT within these developmental windows may lead to the greatest effects. On the other hand, a certain development of EF skills may be necessary to perform CT tasks, or benefit from CT interventions. For instance, problems whose solution requires more steps, or longer algorithms, require sufficient command over impulsive responses and working memory skills.

1.2 Prior Meta-Analytic Studies on the Cognitive Effects of Coding/Programming

The meta-analytic studies evaluating the cognitive effects of CT and programming are very few (Liao, 2000; Liao & Bright, 1991; Scherer et al., 2019; Scherer et al., 2020). To our knowledge, only three meta-analyses have addressed the cognitive effects of coding/computer programming. Liao and Bright (1991) examined 65 studies targeting the relationship between computer programming and cognitive skills such as planning, reasoning skills, and metacognition, without restrictions on grade level (age) or study design (experimental or not). They included both experimental and nonexperimental studies in their meta-analysis and did not differentiate between cognitive functions but considered only a general cognitive effect of computer programming. Their meta-analysis reports a moderate effect of learning computer programming on students' cognitive skills (d = 0 .41). Liao (2000) performed a second, updated meta-analysis summarizing the results of 22 studies with participants from preschoolers to college students targeting a rather heterogenous set of cognitive skills, such as critical thinking, creative thinking, metacognitive skills, problem solving, spatial skills, and conceptual transfer. Their meta-analysis revealed strong effects of computer programming on programming skills (d = 2.48), but only moderate effects on critical thinking, reasoning, and spatial skills (d = 0.37–0.58), and insignificant effects on creative thinking (d = −0.13). Scherer and colleagues (2019) obtained similar findings in a meta-analysis of 105 studies, selecting studies published since 1965. Also, this meta-analysis considered a broad range of cognitive skills and age levels, covering an age span from preschool to university years. Based on the idea that learning computer programming can bring cognitive benefits in several domains, the authors examined the effects of programming on a broad range of cognitive skills, assessing both *near transfer* and *far transfer* effects. *Near transfer* effects are those of computer programming interventions on programming or coding, whereas *far transfer* effects are those of computer programming/coding interventions on cognitive skills less strongly related to coding, such as spatial skills, reasoning and metacognition, or academic achievements, like achievements in mathematics or literacy. Similar to Liao and Bright (1991), also in this meta-analysis the participants in the primary studies varied widely in age, ranging from preschool years to university. Moreover, although Scherer et. applied stricter criteria for study inclusion than Liao and Bright (1991), they considered eligible both standard experimental studies with a pre-posttest randomized control trial design and quasi-experimental studies that reported posttest only measures.

Scherer et al.'s (2019) meta-analytic study confirmed an averagely moderate effect of programming on cognitive abilities (g = .49). The findings showed larger *near transfer* (e.g., to programming, g = .75) than *far transfer* (e.g., to different cognitive abilities, g = .47) effects, but also considerable variation in *far transfer* effect sizes, which the various cognitive skills assessed could explain. Separate meta-analyses for each cognitive skill revealed the beneficial effects of learning computer programming, with effect sizes ranging from g = .73 for creative thinking to g = .37 for spatial skills. Moderator analyses revealed no significant moderating effect of age.

This meta-analysis has significantly contributed to advancing our knowledge on the effects of CT, revealing that certain cognitive functions are more responsive to coding/programming interventions compared to others. However, the inclusion of several quasi-experimental studies with nonequivalent pretest-posttest measures or posttest-only measures in the computation of the effect sizes could have may have affected the findings obtained. Without equivalent pretest and posttest measures there is no way to tell if the observed improvement is real, and due to the intervention, or if it is due to the different tasks performed, or type of skills assessed, in the posttest and pretest. It may indeed be that the posttest task/skill is simply easier than the pretest one. These quasi-experimental studies provide weaker and unreliable evidence of the causal link between the learning of computer programming and cognitive development.

In a subsequent meta-analysis, Scherer and coll. (2020) explored also the effectiveness of different instructional approaches or programming tools (visual programming and robotics) on the acquisition of programming knowledge and skills. Yet, in this study they did not examine the cognitive effects of the different instructional methods.

1.3 The systematic review and meta-analytic study

In instructional research, systematic-reviews and meta-analyses provide the best evidence to support instructional practice (Crocetti, 2016). Combining data from several independent studies increases the statistical power of the analysis and the precision of the effect estimates. The reliability of a meta-analysis relies, however, not only on the on the quantity of studies included but also on the robustness and reliability of their original findings from those studies. For instance, when the aim is to assess instructional effects, including in the meta-analysis studies that lack experimental rigor may represent, as noted earlier, an important limitation. If the research focus is on causality, rigorous hypotheses testing requires controlled experimental trials, consisting of randomized or cluster randomized trials (CRTs; CONSORT guidelines, Campbell et al., 2004).

Although a substantial number of studies in the field of CT report that learning to code improves CT (see, for example, Bers et al., 2014; or Özcan et al., 2021) and EFs (e.g., Arfé et al., 2019; Di Lieto et al., 2020a; Çakır et al., 2021), only some of these are experimental or (cluster) randomized trials (Arfé et al., 2019; Di Lieto et al., 2020a; Özcan et al., 2021). Differently from Scherer et al. (2019), in the present systematic review and meta-analysis we consider only these experimental trials that effectively controlled for the potential impact of repeated testing and practice effects when evaluating the effectiveness of the intervention.

This systematic review and meta-analysis also focus on a narrower set of cognitive skills than those considered by Liao (2000) and Scherer et al. (2019): core and higher order EFs, like response inhibition, WM, cognitive flexibility, planning, and problem solving. Since these skills underpin children's performance across several complex cognitive and academic tasks, determining the effects of CT and programming on these EFs can help explain why the benefits of CT/programming documented by prior metanalyses are so broad, but also vary across tasks. An additional goal is to compare the effects of using different instructional tools (e.g., ER and virtual coding) in the teaching of CT/programming.

Unlike previous meta-analytic studies that have explored the effects of computational thinking (CT) and coding across a wide age range, spanning from preschool to university years (Liao & Bright, 1991; Scherer et al., 2019), the focus of this systematic review and meta-analysis is on the preschool and school years, a time period which represents a critical window for investigating the effects of EF-focused interventions (Luciana & Nelson, 2002; Traverso et al., 2015).

1.3.1 Research questions

We addressed three research questions:

1) Which EFs are most impacted by the teaching of CT/coding?

2) Does the cognitive effectiveness of CT vary with children's age?

2) Which instructional modality (educational robotics/unplugged coding/virtual coding) is the most effective in enhancing executive function skills in learners aged 4 -16 years?

The first research question (Which EFs are most impacted by the teaching of CT?) was addressed both through a systematic-review and meta-analysis, whereas the second and third research questions (Does the cognitive effectiveness of CT vary with children's age? and Which instructional modality is most effective?) were only addressed by a systematic review of the literature. Indeed, the number of studies considered in the meta-analysis were insufficient to statistically test the effects of age and intervention type through moderator analyses.

1.3.2 Method

The systematic review and meta-analysis were conducted in accordance with the guideline of the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA 2020, Page et al., 2021).

Literature Search, Screening and Eligibility Criteria. Studies were selected according to the following keywords: ("executive functions" OR "cognitive control" OR "cognitive abilities" OR "planning" OR "inhibition" OR "working memory" OR "cognitive flexibility" OR "shifting" OR "problem solving") AND ("computational thinking"* OR "coding" * OR "programming" OR "educational robotics") using Scopus, ERIC, Web of Science, PsychInfo, Google Scholar, ACM Digital Library, and IEEE Xplore as databases. Both research articles and conference papers were included in the study, provided they were peer reviewed.

To perform our literature search and identify eligibility criteria we followed Population, Intervention, Comparison, Outcome (PICO) guidelines (Methley et al., 2014). The following eligibility criteria were applied:

- Participants (P): the study participants had to be children or adolescents, between 4 and 16 years old.
- Interventions (I): CT and coding interventions could be based on tangible (ER, unplugged coding) or virtual coding (visual programming) instructional activities.
- Comparison (C): the study had to be an experimental, randomized trial or a CRT, or a matched group trial. Thus, only studies involving equivalent pretests-posttests and an experimental and

a control/comparison group randomly assigned to an experimental or control condition, or matched based on specified criteria, were considered eligible.

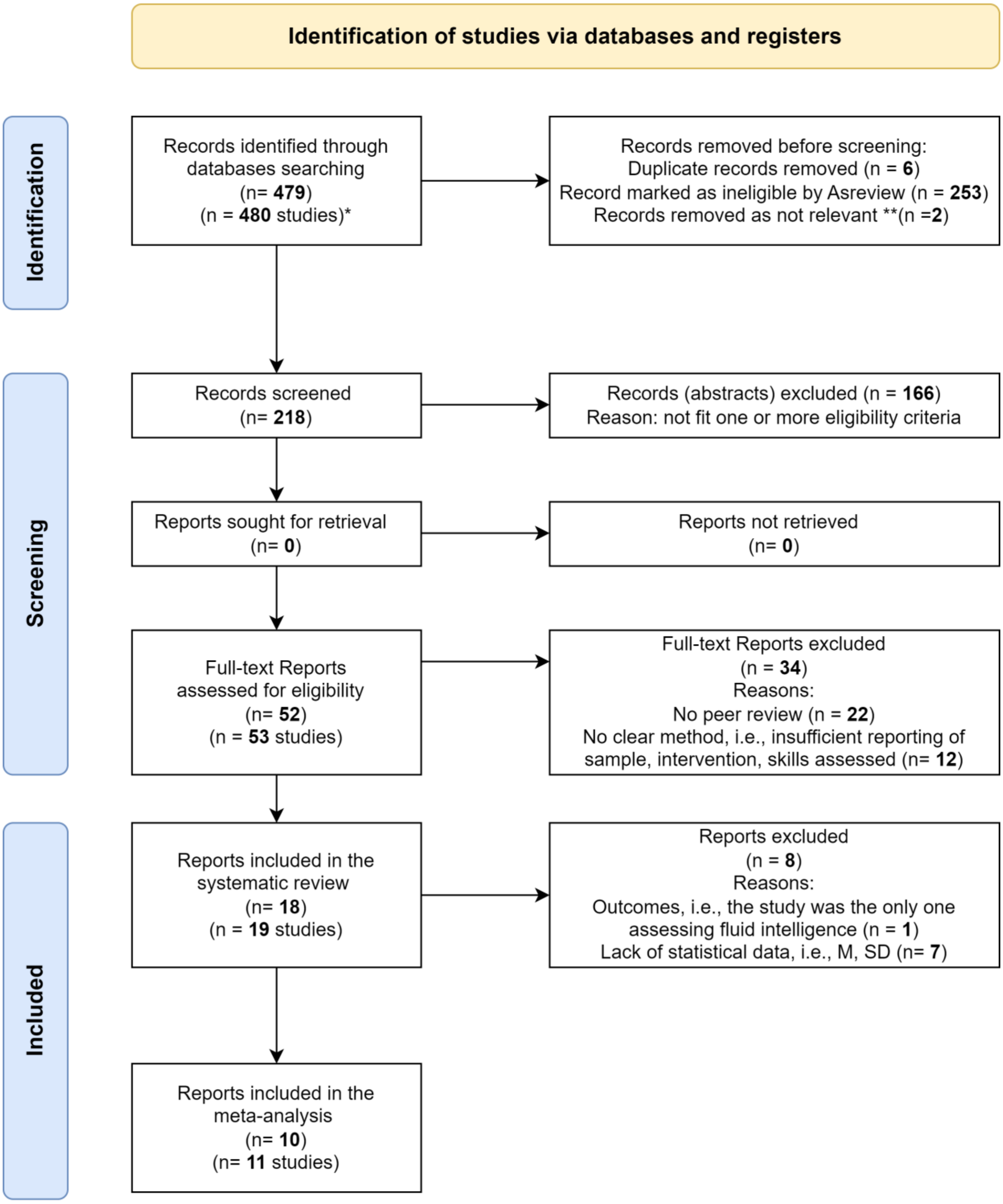
- Outcomes (O): dependent measures (outcomes) were executive functions (e.g., WM, response inhibition, shifting, planning or problem-solving). Fluid intelligence, which relies on cognitive flexibility, was also considered, but only for the systematic review because just one study was found for this outcome. Additional eligibility criteria were:
    - only papers written in English were considered.
    - only peer reviewed publications ranging from period from 2006 (when Wing provided the first definition of computational thinking) to 2022 were considered.
    - the studies should report sufficient information about the intervention tested (e.g., structure, kind of activities).
    - to be included in the meta-analysis all studies should specify sample size, participants' age, and provide effect sizes or means and standard deviations of the pretest and posttest performance.

The PRISMA diagram (Page et al., 2021) in Figure 1 shows the literature selection process. A total of 479 records were identified based on our literature search. An open-source software for systematic reviews, ASReview (https://asreview.nl/), was used for the initial screening of this literature. Two examiners (the first author and a master's student) checked all the abstracts of the records removed by ASReview. After removing duplicates and records marked as irrelevant by ASRview. The abstracts of all remaining records (n = 218) were independently screened by the two judges (the first author and a master student). Based on this screening, 166 further publications were excluded, because they did not fit one or more eligibility criteria. There were 52 papers (53 studies, as one paper presented two studies) assessed for eligibility, after the initial screening; 22 (mainly conference paper) were subsequently excluded because they were not peer-reviewed studies, 12 were excluded because they did not report sufficient information regarding the participants (e.g., age or grade level), interventions or outcomes assessed (the specific measure used, effect sizes means or standard deviations). From this selection process, 18 papers (19 studies) were included in the systematic review. Due to the strict eligibility criteria, only 10 publications (corresponding to 11 studies) were included in the meta-analysis. Of the eight studies included in the systematic review and excluded from the meta-analysis, seven were excluded due to lack of statistical data (i.e., means and standard deviations: Brown et al., 2008; Çakır et al., 2021; Çiftci & Bildiren, 2020; Çınar & Tüzün, 2020; Demir, 2021; Lai & Yang, 2011; Oluk & Saltan, 2015), and one study was excluded because it was the only one assessing fluid intelligence (Özcan et al., 2021). Although meta-analyses normally involve a larger number of studies, meta-analyses on few studies (10 studies or less) are not uncommon, especially in some research areas, like the assessment of new interventions (Mathes & Kuss, 2018).

Disagreements between the two examiners were solved via discussion with the last author of the study, to reach a consensus. There are various methods for calculating IRR (Belur et al., 2021); we used the percentage of agreement between coders (90% agreement) and the κ statistic measure ($\kappa = 0.85$).

Figure 1.1

Flow diagram describing the literature search and the selection of eligible studies (adapted from the PRISMA Statement; Page et al., 2021)



Note. * the number of studies does not correspond to the number of reports bacause one report has two studies; ** the study did not address the target topic of the present systematic review

Studies Coding and Data Extraction. Coding for study descriptors and quality indicators was performed by the first author and, subsequently and independently by the last author. The first author read all 19 studies that met the inclusion criteria and classified the studies by intervention type into one of the three following categories: (1) virtual/plugged coding, (2) unplugged coding, and (3) educational robotics. Instructional activities were further coded as: (a) structured, when children were asked to solve well-defined problems with a clear given goal and correct solution, e.g., giving instruction to a sprite or robot to reach a specific target, or (b) unstructured, when children solved open-ended (or creative) problems with no predefined solution, such as developing a project or designing a new game. There were only two cases of disagreement between the first and last author, which were resolved through discussion.

Each study was coded for thirteen study characteristics, as reported in Table 1.1: (1) study design (randomized control, matched design), (2) participants' grade level and (3) age group (younger or older), (4) sample size, (5) participants (i.e., typical development or atypical development), (6) intervention modality (virtual coding, ER, unplugged coding), (7) intervention tool (e.g., code.org, Scratch, LEGO), (8) whether the intervention was structured or unstructured, (9) intervention activities (the specific activities performed during the intervention), (10) type of control group (i.e., passive or active control group), (11) control group activities (e.g., business-as-usual activities or other types of STEM or programming activities), (12) overall intervention length (in minutes), and (13) main outcomes (with effect sizes when available). Outcome variables were coded in five categories: problem-solving, planning, working memory, inhibition, and cognitive flexibility. Some studies reported intervention duration in hours, some in minutes, and some in hours per week. Thus, all durations were converted to minutes to have the same unit of measurement across studies.

Quality Assessment. Study quality was assessed by means of a checklist (see supplemental material, Quality indicators checklist in the extended table) across seven indicators: (1) quality of the experimental design, (2) presence/absence of an active control group, (3) pretest equivalence or pretest scores controlled in analyses, (4) same pre-posttest measures (5) reliability of the assessment tool (6) method clarity (7) intervention clarity (see Table 1). Each quality indicator was scored as 1 (met) or 0 (not met). If information concerning a specific quality indicator was not provided, it was assumed the indicator was not met, and the study scored 0. A synthesis of the quality assessment is reported in the *tables* folder of the supplemental material (Table quality assessment extended).

Table 1.1

Definitions for study characteristics and quality indicators

	Definition
Study characteristics	
Study design	Type of experimental design: experimental or matched design
Grade level	Participants' grade level. If not provided, age was used to determine likely grade
Age group	Younger or older participants
Sample size per condition	Total number of participants in each condition
Participants	Participants' characteristics: typical or atypical development
Intervention modality	Type of intervention: i.e., virtual/plugged coding, unplugged coding, educational robotics
Intervention tool	Type of tool used in the intervention (e.g., code.org, Scratch, KIBO robot)
Intervention's structure	Structured or unstructured (e.g., fixed games with one solution, or open ended or creative problem-solving activities)
Intervention activities	Specific activities carried out during each intervention session: e.g., game-design, code.org games
Type of control group	Type of control condition (i.e., passive or active)
Control group's activities	Specific activities in which the control group was engaged: e.g., business-as-usual activities or other types of STEM or programming activities
Intervention length	Overall intervention length in minutes. Wherever minutes could not be determined, the most informative length unit, such as number of sessions, was used

Main outcome(s)	The dependent measure(s)
Quality indicators	
Quality of the experimental design	True random experiment or matched-group experiment (i.e., random assignment of individual participants or classes to conditions)
Control condition	Is an active or business-as-usual control group present?
Pretest equivalence	Intervention and control participants showed equivalence in the dependent measure (outcome) at the pretest or pretest measures were covaried
Pre-posttest measures	Intervention and control group dependent measures were assessed at pre and posttest with the same assessment tools
Reliability of the assessment tool	Reliability coefficients for all dependent/outcome measures are reported or the reliability can be inferred by the use of a standardized and validated assessment tool
Method clarity	The study description provides sufficient information on the procedure and the instruments used for the cognitive assessment
Intervention clarity	The intervention is explained in detail with reference to its duration, tools used, and the specific activities performed in each session

Meta-analysis: Data extraction. Data extraction was performed by the two examiners who also performed the literature screening. Once the final pool of studies was selected, one extracted all data (i.e., sample size, means and standard deviations or effect sizes) for the meta-analysis, and the other checked the data extraction subsequently.

In case of papers with missing data, we contacted the study authors before excluding them from the meta-analysis. One paper (Arfé et al., 2019) contained two studies with a different pool of participants. To simplify the data structure, the two experiments were treated as different studies (2019a and 2019b). In the *data* folder of the supplemental material, the full table summarizing the data and outcomes of all included studies is presented (i.e., *meta table cleaned*).

Table 1.2 summarizes all included studies, distinguishing between those included in the systematic review only, and those included both in the systematic review and meta-analysis. In the Results section, we provide detailed descriptions for each study.

1.3.3 Statistical Analysis

Effect Size Computation. We used in the MA the d_{ppc2} effect size measure as Morris (2008) proposed, where the difference between pre-means and post-means for the experimental and control groups is standardized using the pretest standard deviations. The d_{ppc2} has been demonstrated to be the most appropriate measure for this research design. Despite being called *d*, the effect is computed by default (Morris, 2008) by applying the Hedges' correction for small samples. For measures where lower values correspond to better performance (e.g., number of errors) we changed the sign of the effect, thus, positive values always mean that the CT is effective. For computing the effect size sampling variance, we imputed the pre–post correlation because it was missing information from the majority of included papers. We decided to use a pre–post correlation of 0.7. We included a sensitivity analysis using different correlation values (0.5, 0.7, and 0.9) in the supplementary materials.

Between and within studies the same underlying psychological construct could be measured with different instruments. We thus decided to assign the same label to effects referring to the same underlying psychological construct. For example, measures derived from WM, the Backward Corsi Block Tapping subtest (BVN test; Bisiacchi et al., 2005) and Matrix Path (BVS Corsi; Mammarella et al., 2008) were all coded as WM. Similarly, scores at standardized tests, ad hoc problem-solving tasks and self-report scales assessing problem solving were coded as problem solving. In this way, we classified the reported outcomes according to the underlying EF: inhibition, WM, shifting, planning, and problem solving. Thus, multiple effect sizes referred to the same construct were aggregated to obtain a single measure. Aggregating multiple statistically dependent effect sizes requires imputing the correlation between different measures. Following Borenstein and colleagues (2009, pp. 225–233), we used a correlation of 0.5, but with an inverse-variance weighted average

(Viechtbauer, 2010). A sensitivity analysis using different correlation values (0.3, 0.5, and 0.7) is presented in the supplemental material.

Statistical Model. When multiple outcomes are collected from the same pool of participants, a situation of statistical dependency emerges, which, if ignored, brings strongly biased meta-analytic estimations (Cai & Fan, 2020; Cheung, 2019; Van den Noortgate et al., 2015). The most appropriate approach is to consider the correlation between multiple outcomes using a multivariate MA model (Cai & Fan, 2020). Given the limited number of papers, we decided to use a *fixed-effect* model as Cai and Fan (2020) suggest. Despite the *random-effects* model allowing to generalize conclusions at the population level, the between-papers variability estimation (i.e., the focus of the *random-effects* model) can be strongly biased with a limited number of papers (Cheung, 2013; Veroniki et al., 2016). The multivariate fixed-effect model estimates the average effect size for each outcome and requires considering the correlation between different measures. Again, the included papers did not report this correlation, thus, we imputed a value of 0.5. As before, we conducted a sensitivity analysis using different values (0.3, 0.5, and 0.7), again presented in the supplemental material.

The MA model was implemented in R (R Core Team, 2021) using the *metafor* package (Viechtbauer, 2010). For hypothesis testing, we used the *omnibus* Wald χ^2 test. For each outcome, we reported the average effect size, standard error, the 95% confidence interval, and the associated Wald z test. Further details about the analytic approach can be found in the supplemental material. All code and data to reproduce the analysis (supplemental material) are available online in the Open Science Framework repository (https://osf.io/uvbcd/).

1.4 Results

1.4.1 Characteristics of Included Studies

The sample comprised 19 studies included in the systematic review, for a total of 1527 participants. Of the primary 19 studies, 11 were also included in the meta-analysis (see Table 1.2 and Figure 1.1). The overall sample size of the MA was of 862 participants (n = 433 in experimental conditions and n = 429 in control conditions).

The study samples included participants from 4 to 16 years, with a predominance of preschoolers or primary school children (12 studies, 63%). Five studies (26%) involved children from grade 5 to grade 8, and only one study involved older students (10^{th} graders). The sample size of the included studies ranged from 28 to 187, with a majority of studies with samples > 50. Among the studies selected, 12 tested the effects of virtual coding interventions with children from preschool to grade 8, and seven tested the efficacy of tangible coding interventions (ER or unplugged coding activities) with preschoolers and students from grade 1 to grade 10. Only 31% of the interventions consisted of unstructured coding/programming activities (n = 6). They were all addressed to older students (grades 5 to 8) and involved virtual coding. Among the outcome measures, problem solving skills were the most frequently assessed (13 studies, 68%), inhibition skills were assessed in 5 studies, planning was assessed in 3, working memory in 2, cognitive flexibility in 2, and fluid intelligence in one study only.

Quality of the Studies. A synthesis of the studies' quality is reported in the extended Table Studies characteristics and quality *indicators* in supplemental material. The overall quality was high. When all seven quality indicators were considered, the studies satisfied on average 78.20% of the quality criteria, with 8 studies meeting 100% of quality indicators.

1.4.2 Results of the Systematic Review

Table 1.2 summarizes the characteristics of the selected studies and their main outcomes; effect sizes are also reported if available. As in Scherer et al. (2019), we distinguished between *near transfer* effects (i.e., transfer between similar/closely related tasks or skills) and *far transfer* effects (i.e., the transfer between dissimilar tasks, which require different skills or strategies; Perkins & Salomon, 1992). As shown by Table 1.2, the majority of the studies assessed *far transfer* effects. In the following sections, the results are presented with reference to the three research questions of the study.

Table 1.2

Studies Included in the Systematic review and Meta-analysis (Effect Sizes or Significant Effects are Reported)

Author(s)/year	Age group	Grade	Sample size	Intervention type	Coding tool	Structured/ Unstructured	Intervention length	Included SR/MA	Effects sizes or Sign. effects
Akcaoglu & Koehler (2014)	Older	Grades 5-8	44	Virtual coding	Microsoft Kubo	U	900 min	SR/MA	Problem-solving (d = 1.05)
Brown et al (2008)	Older	Grades 5-6	113	Virtual coding	Scratch	U	180 min	SR	Problem-solving
Çınar & Tüzün (2020) *	Older	Grade 10	81	Educational robotics	LEGO Mindstorms NXT 2.0	S	2160 min	SR	Problem-solving n.s.
Demir (2021) *	Older	Grades 9-11	34	Coding unplugged	Algorithm cards	S	--	SR	Problem-solving
Erol & Çırak (2022) *	Older	Grade 6	34	Virtual coding	Scratch	U	1680 min	SR/MA	Problem-solving ($\eta^2 = 0.26$)
Lai & Yang (2011)	Older	Grade 6	130	Virtual coding	Scratch	U	--	SR	Problem-solving
La Paglia et al (2017)	Older	Grades 5-6	60	Educational robotics	LEGO Mindstorms	S	1800 min	SR/MA	Problem-solving
Nam et al (2010)	Older	Grade 6	60	Virtual coding	Scratch	U	480 min	SR/MA	Problem-solving
Oluk & Saltan (2015) *	Older	Grade 6	65	Virtual coding	Scratch	U	720 min	SR	Problem-solving n.s.
Özcan et al (2021)	Older	Grade 4	174	Virtual coding	Code.org + Scratch	S+U	1200 min	SR	Fluid intelligence (cognitive flexibility) n.s.
Pardamean et al (2011)	Older	Grade 5	85	Virtual coding	Logo programming	S	640 min	SR/MA	Problem-solving[#]

Arfé et al (2019a)	Younger	Grade 1	80	Virtual coding	Code.org	S	480 min	SR/MA	Inhibition (d= -0.65) Planning (d=0.95) Problem-solving (**NT**, d=1.62)
Arfé et al (2019b)	Younger	Grade 2	38	Virtual coding	Code.org	S	480 min	SR/MA	Inhibition (d= -1.05) Planning (d=0.93) Problem-solving (**NT**, d=1.91)
Arfé et al (2020)	Younger	Grade 1	179	Virtual coding	Code.org	S	480 min	SR/MA	Inhibition (d= -0.71) Planning (d= 1.27) Problem-solving (**NT**, d=1.31)
Çakır et al (2021)	Younger	Preschool	40	Educational robotics	LEGO WeDo 2.0	S	1920 min	SR	Problem-solving
Çiftci & Bildiren (2020)	Younger	Preschool	28	Virtual coding	Code.org	S	480 min	SR	Problem-solving n.s.
Di Lieto et al (2020a)	Younger	Grade 1	187	Educational robotics	Bee Bot	S	1200 min	SR/MA	Inhibition (d= 0.69) Working memory (d= 0.65) Cognitive flexibility n.s.
Di Lieto et al (2020b)	Younger	Grade 1	42	Educational robotics	Bee Bot	S	1200 min	SR/MA	Inhibition Working memory n.s. Cognitive flexibility n.s.
Nam et al (2019)	Younger	Preschool	53	Educational robotics	TurtleBot	S	720 min	SR/MA	Problem-solving (η^2 = 0.17)

Note: Age group = the category by which results are reported; Effect sizes = Significant major effects; S = Structured; U = unstructured; NT = near transfer effects; n.s. = non-significant; * the comparison group was active control group exposed to coding/programming activities (e.g., visual programming tool, learning programming and algorithms during Information Computer Technology class); # non-equivalent pretest scores were not covaried in the analyses.

Which EFs are most impacted by the teaching of CT?

The systematic review revealed that CT interventions are generally effective in boosting children's EFs. Of the 19 studies examined in this systematic review, only four reported non-significant effects of CT interventions (Çiftci & Bildiren, 2020; Çınar & Tüzün; 2020; Oluk & Saltan; 2015; and Özcan et al., 2021).

Problem Solving. Sixteen out of 19 studies examined the effects of CT programs on children's problem solving, assessed by problem solving tests or ad hoc tasks (Akcaoglu & Koehler, 2014; Brown et al., 2008; Lai & Yang, 2011; Nam et al., 2010) or self-report problem solving measures (Erol & Çırak, 2022; Oluk & Saltan, 2015). Six studies tested the effects of game-design or project-development activities, while the remaining 10 studies explored the efficacy of more structured virtual coding or educational robotics programs.

Game-design and project-development interventions. Significant and positive effects of game-design and project-development interventions were found in studies in which students' problem-solving skills were assessed by the Program for International Student Assessment (PISA) of the Organisation for Economic Co-operation and Development (EOCD, 2013; Akcaoglu and Koehler, 2014; Nam et al., 2010). Akcaoglu and Koehler (2014) proposed to 11-14-year-old students a 15-hours game-design activities in which students had to design digital games through Kodu Game Lab, a 3D game development environment for visual programming. The authors found a large effect of the intervention program, Cohen's $d = 1.05$, on participants' problem solving.

Similar findings are reported by Nam et al. (2010), despite their intervention program, based on project-development activities, was approximately half the time that of Akcaoglu and Koehler's (2014): 8-hours versus the 15-hours. The authors involved 12-year-old students in four weeks project-development activities based on Scratch, an open-source block-based visual programming tool used to create interactive stories and games, finding significant improvements of the participants on PISA problem solving assessment.

Project-development activities based on Scratch result effective also when compared with other experimental trainings (active control groups) and for students with diverse background. Lai and Yang (2011) assessed the efficacy of visual programming activities based on Scratch on sixth grade students' problem solving and reasoning skills assessed by a problem-solving test (Chan et al., 1991). During the intervention, the students received instruction on the basic operations and tools of Scratch, scaffolding in problem solving and subsequently developed their own project. The problem-solving skills of the students involved in the Scratch-based program improved significantly more than those of an active control group, participating in Adobe Flash learning activities.

Brown and colleagues (2008) tested the effects of four 45-min lessons (approximately 4-hours program) based on Scratch on fifth and sixth graders' mathematical problem-solving. All students, predominantly African American, were from disadvantaged backgrounds and low-income families. The Scratch lessons, designed to introduce the students to efficient and inefficient mathematical problem-solving methods, were focused on learning debugging and loops, two key operations in CT and programming. Students' problem-solving strategies were assessed by ad-hoc designed problem-solving exercises, consisting of mathematics problems that could be solved using an efficient (e.g., multiplication or loop) or inefficient (e.g., repeated addition) strategy. The results again showed that the students in the experimental group improved in solving mathematical problems more than the control group, addressed to standard instructional activities.

Other studies have examined the effects of CT programs on students' self-reported problem-solving. Erol and Çırak (2022) used the Problem-Solving Inventory for Children (PSIC, Serin et al., 2010) to assess changes in the problem-solving of a group of Turkish sixth graders after a 12-week game-design intervention (24-hours) with Scratch. The PSIC inventory included items to assess participants' self-confidence, persistence, and determination in problem solving tasks. The students addressed to the game-design intervention were first introduced to programming basics (e.g., operations, control structures) and the use of Scratch, and then to semi-structured and free game-design activities. Their problem-solving scores were compared against those of an active control group who were also introduced to programming and asked to develop algorithms that solved problems during Information Technologies and Software classes. The findings revealed a significantly larger improvement in students' approach to problem-solving for the experimental than the active control group ($\eta^2 = .26$).

Only one study, by Oluk and Saltan (2015), reported unsignificant effects of Scratch-based programs. The authors tested the effects of a 12-hours algorithm development Scratch-based instructional program on sixth-grade students' self-reported of problem solving. The students in the experimental group (31 participants) learned algorithms with the help of Scratch, whereas the control group (34 participants) received standard curricular instruction on algorithm development. Their approach to problem-solving was assessed by the same problem-solving inventory used by Erol and Çırak (the PSIC, Serin et al., 2010). Neither of the two groups showed improvement in problem solving scores. It must be noted, however, that differently from Erol and Çırak's study, the instructional program tested in this study did not focus on game-design or project-design activities, which are comprehensive problem-solving tasks, but targeted only one of the components of problem solving: algorithmic thinking.

Other visual programming/educational robotics and unplugged coding interventions.

Other CT interventions based on different visual programming tools, such as *code.org* or *Logo programing*, *unplugged coding*, or *educational robotics* are more structured than game-design or project-development interventions, and for this reason were primarily addressed to younger children, preschoolers or first or second graders. These interventions too have been proven effective in stimulating children's problem-solving skills.

Five studies tested the effects on problem solving of structured visual programming (i.e., virtual coding) activities (Arfé et al., 2019a; Arfé et al., 2019b; Arfé et al., 2020; Çiftci & Bildiren, 2020; Pardamean et al., 2011). In three studies (Arfé et al., 2019a; Arfé et al., 2019b; Arfé et al., 2020), involving a total of 297 children, 1st or 2nd graders, Arfé et al. found consistent and significant *near transfer* effects on problem solving of a short, 8-hours, structured visual programming intervention based on *code.org*. The students practiced individually with *code.org* games, and after participated in classroom discussions regarding how they solved the coding problems. Like Scratch, *code.org* is a visual programming platform, in which children write their code by moving programming blocks to construct sequences of commands that give instructions to a sprite, or a character (Angry bird, a bee, a zombie) which execute them. Differently from Scratch, however, the games typically have a predefined structure and aim at a given objective (e.g., getting to a target or performing a specific action). Although structured, in Arfé et al.'s studies the intervention was designed to cause children to switch between scenarios, programming functions (e.g., loops, debugging) and types of problems to force children to maintain a problem-solving approach. Children's problem-solving abilities were assessed by asking children to solve new *code.org* games, similar to those solved during the instructional program. The performance of the children who received the intervention was compared to that of a wait-list group receiving standard STEM instruction. Effect sizes were large, ranging between d = 1.31 to 1.91.

Different results were obtained by Çiftci and Bildiren (2020), who used a problem-solving skill scale (Aydoğan et al., 2012) to test the *far transfer* effects of a structured 8-hours virtual coding intervention based on *code.org* on 28 4- to 5-year-old children's problem-solving abilities. The problem-solving scale aimed to assess the abilities shown by children when facing actual real-life problems. Children were shown pictures representing real life problems, explained by short stories that defined the problem and were asked to find the best answer to the problem. The study showed no significant improvement of children's problem-solving skills. The intervention improved though their logical, nonverbal, cognitive abilities measured by the Raven Colored Progressive Matrices Test.

One of the studies examined (Pardamean et al., 2011), yielded unconclusive findings due to a methodological flaw. The authors assessed fifth graders' creativity and problem-solving skills following a 16 40-minutes lessons course based on Logo programming. Logo is a simple programming language, by which children learn to program generating commands to control the movements of a turtle, a cursor, for creating drawings or geometric forms. In the study, the children in the Logo programming intervention worked in peers to solve geometric games. Their creative skills were assessed by a creative thinking figural test and their problem-solving skills were assessed by a logical word test and figural problem-solving test. The results revealed statistically significant effects of the intervention on children's creativity. At the posttest, children in the Logo program were also better in figural problem solving than the control group, who participated in standard ICT (Information and Computer Technology) curricular activities. However, as the experimental group outperformed the control group in figural problem solving also at the pretest and pretest scores were not covaried in the analyses, it is not possible to determine whether the outcome reflected true intervention effects or individual differences between the groups.

Other studies have examined the effects of educational robotics (ER) activities on children or early adolescents' problem-solving (Çakır et al., 2021; Çınar & Tüzün, 2020; La Paglia et al., 2017; Nam et al., 2019). La Paglia et al. (2017), involved 30 sixth graders (10-12 years) in an extra-curricular ER laboratory of 10 3-hour sessions. The laboratory activities employed a LEGO Mindstorms robot kit. Students participating in the laboratories worked in group to build a robot body and generate a program to assign it an artificial intelligence. The effects of the ER intervention were assessed by a metacognitive questionnaire assessing students' ability to apply metacognitive skills to problem solving, comparing the performance of the experimental group to that of a passive control group of 30 students not participating in any extra-curricular activity. The results revealed that the ER laboratory significantly improved the metacognitive control skills involved in problem-solving of the experimental group.

When problem-solving abilities are assessed by self-report measures, the results are less clear. Çınar and Tüzün (2020) compared the improvement in problem solving of two groups of 10th graders: one participating in ER activities, and the other in object-oriented visual programming (active control group). During the experimental ER intervention, which lasted 12-weeks (approximately 36 hours), the students worked in groups using Lego Mindstorms NXT 2.0 to build robots and subsequently manipulate the program to train their programming skills. The comparison, active control, group performed CT activities in a visual programming environment individually or in group. During the practical problem-solving sessions guideline questions were used to scaffold the problem solving of both groups. Problem solving was assessed through the Problem-Solving Inventory (PSI) developed by Heppner (1988), which provides a self-assessment of behaviors and approaches associated with successful problem solving. No significant changes were observed in the problem-solving scores of the two groups.

Only two studies have examined the effects of ER interventions on the problem-solving of preschoolers and first graders (Çakır et al., 2021; Nam et al., 2019). Nam and colleagues (2019) tested the effects of a 90 minutes -8-sessions- card-coded robotic course on 5–6-year-old children's mathematical problem solving. Children were divided in two groups: the experimental group participated in the card-coding robotic course, while the comparison group participated in daily school activities and performed problem-solving activities as outlined in the national school curriculum (e.g., board games). A TurtleBot, which is a card-coded robot, was used for the ER intervention. The instructional program consisted of several activities: problem identification, planning with the support of worksheets, coding with cards and observing or evaluating the program outcomes, followed by debugging. Mathematical problem-solving was assessed by an adapted version of the Ward (1993) problem solving instrument, which assesses children's problem-solving skills across various questions, ranging from categorization and patterns to statistics. The results revealed that the children in the ER group improved significantly more in problem-solving than controls: The dimension of the effect was large (η^2= 0.17).

Similar findings are reported by Çakır et al., (2021), who evaluated the effect of a 32-hours ER intervention in which LEGO Education WeDo 2.0 was used to enhance the problem-solving skills of preschoolers. Children in the experimental group were asked to first build a robot, and subsequently to write a code for programming the robot through drag-and-drop block-based programming activities. After completing and executing the program, they had to verify the accuracy of the instructions given to the robot, reflecting on the functions of the different code blocks, and on the codes used for the activity. Children in the (active) control group were involved in joint book reading activities, reasoned about the stories read, and performed creativity and categorization activities. Children's problem-solving was measured by the Problem-Solving Skill Scale (PSSS, Oğuz & Köksal Akyol, 2015) assessing problem-solving applied to real-life problems. The results showed a significantly greater increase in problem-solving in the experimental compared to the active control group.

Only one study (Demir, 2021) tested the effects of unplugged coding activities on problem solving. Participants were 34, 14 to 18-year-old, students with mild intellectual disabilities. Demir examined the effects of unplugged coding activities on their problem solving. Special education classes, of approximately 4-6 students each, were randomly assigned to an experimental or control condition. The experimental intervention required the students to play structured games, based on unplugged activities, such as finding solutions to problems presented in stories or Tower of Hanoi problems. Problem-solving skills were assessed by asking to solve everyday like problems (e.g., washing dishes or making pasta problems). The results showed significant effects of the unplugged coding intervention.

Planning and Core EFs. Six studies considered in this systematic review assessed other EF skills, such as core EFs (inhibition, working memory and cognitive flexibility or switching), planning or fluid intelligence. They involved primarily younger children, from preschool to grade 2, and tested the effects of structured visual programming or ER interventions.

In their studies, Arfé and colleagues (Arfé et al., 2019 a, 2019b; Arfé et al., 2020) examined also the effects of *code-org* game-based coding program on first and second graders' cognitive inhibition and planning abilities, finding that the 8-hours code.org based intervention produced significant and large effects on the planning and cognitive inhibition skills of first and second graders. Cognitive inhibition and planning skills were assessed through standardized neuropsychological tests. Inhibition skills were assessed by the square and circle NEPSY-II subtests (Korkman et al., 2007) and a Numerical Stroop test (Batteria Italiana ADHD, BIA*,* Marzocchi et al., 2010). Planning skills were assessed by the Elithorn maze test (Gugliotta et al., 2009) and the Tower of London, ToL test (Luciana et al., 2009). The effect sizes ranged from d = 0.65 for response inhibition to $d = 1.27$ for planning. These results confirmed that even a relatively short structured virtual coding intervention (8 h of code.org-based activities) can boost children's cognitive inhibition and planning abilities.

Virtual coding interventions seem however less effective in stimulating cognitive flexibility. In a randomized trial, Özcan et al.'s (2021) compared the effects of a 10-week (20 hours) learn-to-code (virtual coding) program to two control instructional conditions: another STEM comparison treatment, based on mathematics, and a reading program control condition. One-hundred and seventy-four fourth graders from socioeconomically disadvantaged backgrounds were equally distributed and randomly assigned to the three conditions. Their fluid intelligence was assessed with the matrix reasoning task from the Wechsler Abbreviated Scale of Intelligence Measurement (Wechsler, 2011) at pre and posttest. The virtual coding program combined structured visual programming activities with Algo Digital (https://algodijital.com/) and code.org to project-based activities with Scratch. Although computational thinking scores improved significantly only for children in the learn-to-code treatment condition, children's fluid intelligence, a measure of cognitive flexibility (Colzato et al., 2006), improved equally in all groups, indicating that children's gains were unrelated to the intervention.

The few studies that have tested the effects of ER programs on children EFs have shown beneficial effects of ER on working memory and response inhibition skills in younger children (Di Lieto et al., 2020a, 2020b). Di Lieto et al. (2020a) examined the effects of a structured 20-h ER training on 5 and 6-year-old first graders' inhibition skills, WM, and cognitive flexibility. Cognitive inhibition was assessed by three standardized neuropsychological tests, the NEPSY-II circle and square subtest (i.e., Korkman et al., 2007), the Little Frogs subtest (i.e., BIA; Marzocchi et al., 2010), and Pippo Says test, a modified version of Simon-Says test assessing both inhibition and switching, that is cognitive flexibility (Marshall & Drew, 2014). Working memory was assessed by two visuospatial tasks: the Backward Corsi Block Tapping subtest (BVN test; Bisiacchi et al., 2005) and Matrix Path (BVS Corsi; Mammarella et al., 2008). Children were randomly assigned to an experimental (ER intervention, n = 96) or a control (wait list, n = 91) group condition, in which children participated in daily school activities and received the ER instructional program later. Children in the experimental group were introduced to programming through engaging coding activities with Bee-Bot, an interactive robot with a bee shape that can be programmed to execute movements using buttons on its back. After a familiarization phase with the Bee Bot, children were invited to solve complex visuospatial planning tasks with the robot to stimulate their working memory and inhibition skills. In the last sessions, task switching and inhibition tasks with Bee Bots were targeted. After the 20-h ER training, the intervention group showed significant improvements in WM and inhibition abilities with moderate effect size: $d = 0.63$ for visuo-spatial WM, and from $d = 0.43$ to $d = 0.69$ for inhibition skills, which is similar to the dimension of the effect reported for children of same grade level when the intervention involves virtual coding activities (Arfé et al., 2019a and Arfé et al., 2020). Insignificant effects were instead found for children's cognitive flexibility (Di Lieto et al., 2020a).

In a second study, Di Lieto et al. (2020 b) tested the efficacy of similar structured ER activities on the visuo-spatial WM and inhibition skills of 42 1st graders with special needs, including language disabilities, attention disabilities and cognitive or motor impairments. Like in the previous study, children were assigned to an experimental or wait list group condition. The 20-hours ER program was adapted to meet the motor, cognitive, and social needs of the children. The assessment instruments were the same as in the previous study. Again, the results showed an improvement in response inhibition for the children who received the training. The training effects were instead insignificant for WM.

Synthesis of the Research Findings. To summarize the findings of the studies, the systematic review revealed wide and positive effects of CT interventions on children's problem solving, complex EFs such as planning as well as on core EFs, such as cognitive inhibition and working memory skills. Remarkably, these significant effects are found for long interventions, lasting up to 32 hours (Çakır et al., 2021), as well as for short intervention programs of only 4 hours (Brown et al., 2008) and extend also to children with special needs (Di Lieto et al., 2020b; Demir, 2021) or disadvantaged backgrounds (Brown et al., 2008).

Effect sizes are typically larger for problem solving, both when problem solving skills are assessed by tasks similar to those proposed in the intervention programs (*near transfer*; effect sizes range from $d = 1.31$ to $d = 1.91$), and when they are assessed by different tasks, like PISA assessment tools (Akcaoglu and Koehler, 2014, $d = 1.05$) or self-report scales (Erol and Çırak, 2022, $\eta^2 = .26$). Notably, the studies are consistent in showing significant effects of CT interventions even in comparison to active control groups, assigned to other STEM or programming activities (e.g., Arfé et al., 2019a; Erol & Çırak, 2022).

Exceptions to these findings are represented by intervention studies focused on single components of CT, such as algorithm development (Oluk & Saltan, 2015), studies in which intervention effects were assessed by self-reports or problem-solving questionnaires instead of performance on cognitive tests or ad hoc problem solving tasks (see Çiftci & Bildiren, 2020; Çınar & Tüzün, 2020; Oluk & Saltan, 2015) or studies assessing cognitive flexibility (Özcan et al., 2021).

CT programs appear very effective even when EF skills are concerned. The dimension of the instructional effect is large for complex EFs such as planning (e.g., Arfé et al., 2020, d = 1.27) and moderate to large for core EF skills. Again, the only exception is represented by shifting or cognitive flexibility. However, evidence is still limited in this latter case, as only two of the studies examined in this systematic review assessed cognitive flexibility.

Does the cognitive effectiveness of CT vary with children's age?

As shown in Table 2, we divided the studies in two age-categories based on the age of their participants. The first category comprised older children, from grade 4 to grade 10, the second included preschoolers, first and second graders. The distribution of the studies across the two age-categories was balanced. Eleven of the studies considered in this systematic review tested CT programs addressed to older students, in one case (Demir, 2021) with mild intellectual disabilities. The remaining eight studies involved children from preschool to grade two. Based on our systematic analysis of the literature, CT interventions resulted equally effective for older and younger children.

Older age-group. The CT programs seem generally effective for the participants in the older age-group. Among the eleven studies including older participants only three did not report significant positive effects of the CT programs. One was the study by Oluk & Saltan (2015), in which, as anticipated, the insignificant effects could be attributed to the narrow nature of the CT intervention that focused on a single component of CT, that is algorithm development. The other two studies reporting non-significant effects, by Çınar and Tüzün (2020) and Özcan et al. (2021), proposed structured ER and visual programming interventions to 10th and 4th graders respectively. In the first study (Çınar & Tüzün, 2020), the students' programming skills improved, but their self-reported problem-solving abilities seemed not to be affected by the intervention. Also in the second study (Özcan et al., 2021) students' computational thinking skills improved but the effects did not generalize to fluid intelligence, that is cognitive flexibility. It is interesting to note that similar results are reported by Di Lieto et al. (2020a) who found that cognitive flexibility resulted difficult to improve also in younger (first grade) children (e.g., Di Lieto et al., 2020a).

Younger age-group. Also for the younger participants the studies reported significant positive effects of CT programs. The only exception is the study by Çiftci and Bildiren (2020), in which *far transfer* effects of a structured virtual coding intervention were assessed by a Problem-Solving Skill Scale (Aydoğan et al., 2012). The assessment tool used in this study could however explain the finding. The scale consisted of picture items representing various types of real-life problems. Children (4- and 5-year-old) had to understand the problem represented and find the best answer to it. The way problems were formulated and understood by the participant could have affected the results.

In synthesis, CT intervention programs yielded to significant and positive far transfer effects both in the older and younger groups. In the younger children significant transfer effects, from moderate to large were observed both for complex EF skills and core EFs, such as WM and cognitive inhibition skills (Arfé et al., 2019a; Arfé et al., 2019b; Arfé et al., 2020; Di Lieto et al.,2020a). For the older children, only complex EFs like problem solving and fluid intelligence were assessed, but again the effects were significant and effect sizes, when available, were large.

Which instructional modality (educational robotics/unplugged coding/virtual coding) is most effective?

Differences in the effectiveness of CT programs seem related to the nature (structured or unstructured, and comprehensive or not) of the intervention more than to its modality (virtual coding, ER, or unplugged coding). Both virtual coding (visual programming) and ER intervention programs resulted effective when they addressed the various components of CT, such as problem analysis, planning, evaluating and debugging. Conversely, CT interventions focused on just one component or on programming skills solely resulted less effective (Oluk & Saltan, 2015). Interventions were also effective both when they were specifically tailored to boost specific EF skills, like in Di Lieto et al. (2020a) and when they were more broadly targeting problem solving abilities (Arfé et al., 2019a; Arfé et al., 2020).

Game design or project development programs, in which students are allowed to develop their own projects, have been found effective in enhancing the problem-solving skills of older participants (Akcaoglu & Koehler, 2014; Erol & Çırak, 2022). These intervention programs were not addressed to younger children, who typically received more structured interventions, based on tasks with specific and predefined objectives. In these structured programs both visual programming and ER tools have been used, resulting equally effective. For instance, Arfé et al. (2019a; 2019b) and Di Lieto et al. (2020a), who proposed to 5-6-year-old children respectively virtual coding and ER activities found very similar effects on children's response inhibition skills: $d = 0.65$ and $d = 0.71$ (Arfé et al., 2019a and 2019b) and $d = 0.69$ (Di Lieto et al., 2020a).

It must be noted, however, that unstructured game-design or project-development programs and structured visual programming interventions, although proposed to children of different ages, yielded similar effects on students' problem solving or planning outcomes (Akcaoglu & Koehler, 2014; Arfé et al., 2019a; Arfé et al., 2019b). Akcaoglu and Koehler (2014) and Arfé et al., for instance, reported similar large effect sizes of project-development and virtual coding interventions on the problem-solving skills of students from 5th to 8th grade and 1st and 2nd graders respectively.

In synthesis, when comprehensive structured CT interventions, that involve scaffolding and practicing different CT components, are proposed to younger children their transfer effects can be similar to that of problem-solving unstructured programs addressed to older participants. This does not mean, though, that the effects of CT intervention at a given age can be independent from its structured/unstructured nature. We do not have, indeed, evidence in support of this hypothesis. Conversely, the comparison between virtual coding and ER interventions provides clear evidence of the equivalence of these two CT tools in boosting younger children's problem solving and EF skills.

1.4.3 Meta-analysis Results: Effects of CT/coding interventions on problem solving, planning and core EFs

The meta-analysis allowed to derive quantitative estimates of the cognitive effects of CT/coding interventions across EFs. The *omnibus* test suggests that at least one outcome differs from zero (χ_5 = 195.693, p < 0.001). Table 1.3 summarizes the results of the multivariate fixed-effect model. Figure 2 depicts the multivariate forest plot.

Table 1.3

Multivariate Fixed-Effect Model Summary

Outcome	d_{ppc2}	SE	95% CI	z	*p*
Cognitive Flexibility Acc.	0.118	0.096	[-0.07, 0.306]	1.227	0.22
Inhibition Acc.	0.168	0.057	[0.057, 0.279]	2.956	0.003
Planning Acc.	0.364	0.072	[0.222, 0.505]	5.047	< 0.001
Problem Solving	0.890	0.064	[0.764, 1.016]	13.816	< 0.001
Working Memory Acc.	0.199	0.079	[0.045, 0.353]	2.530	0.011

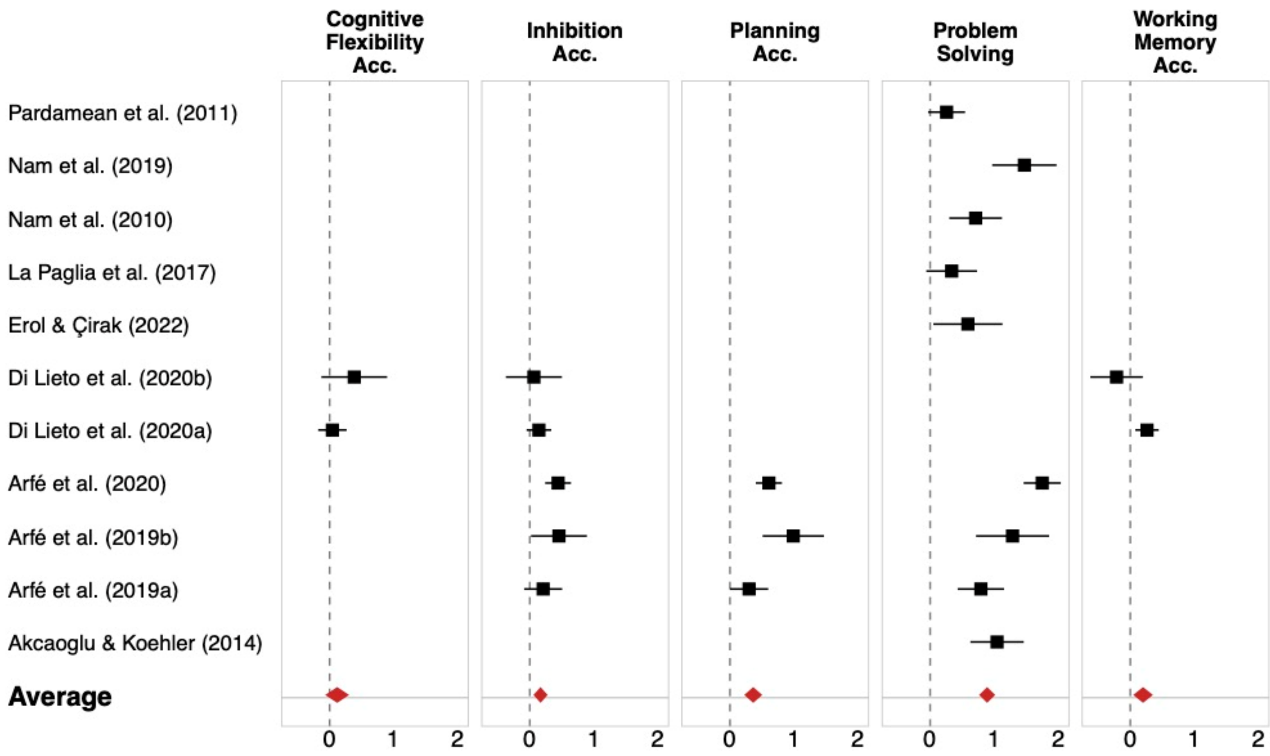
Omnibus Test $\chi_5 = 195.7$ $p < 0.001$; $\rho_{pre\text{-}post} = 0.7$, $\rho_{agg} = 0.5$, $\rho_{multi} = 0.5$

Note. Each outcome is summarized with the estimated mean effect (d_{ppc2}), the standard error, the 95% confidence interval, the *z* value, and the p-value.

The results of the meta-analytic study (Figure 1.2) show that except for cognitive flexibility (accuracy), all cognitive outcomes improved significantly after children performed coding or programming activities. Problem solving is associated with the highest effect ($d_{ppc2} = 0.89$), planning with a moderate effect ($d_{ppc2} = 0.36$), and response inhibition and WM, despite being statically significant, associated with lower effect size ($d_{ppc2} = 0.17$, $d_{ppc2} = 0.20$).

Figure 1.2

Multivariate Forest Plot

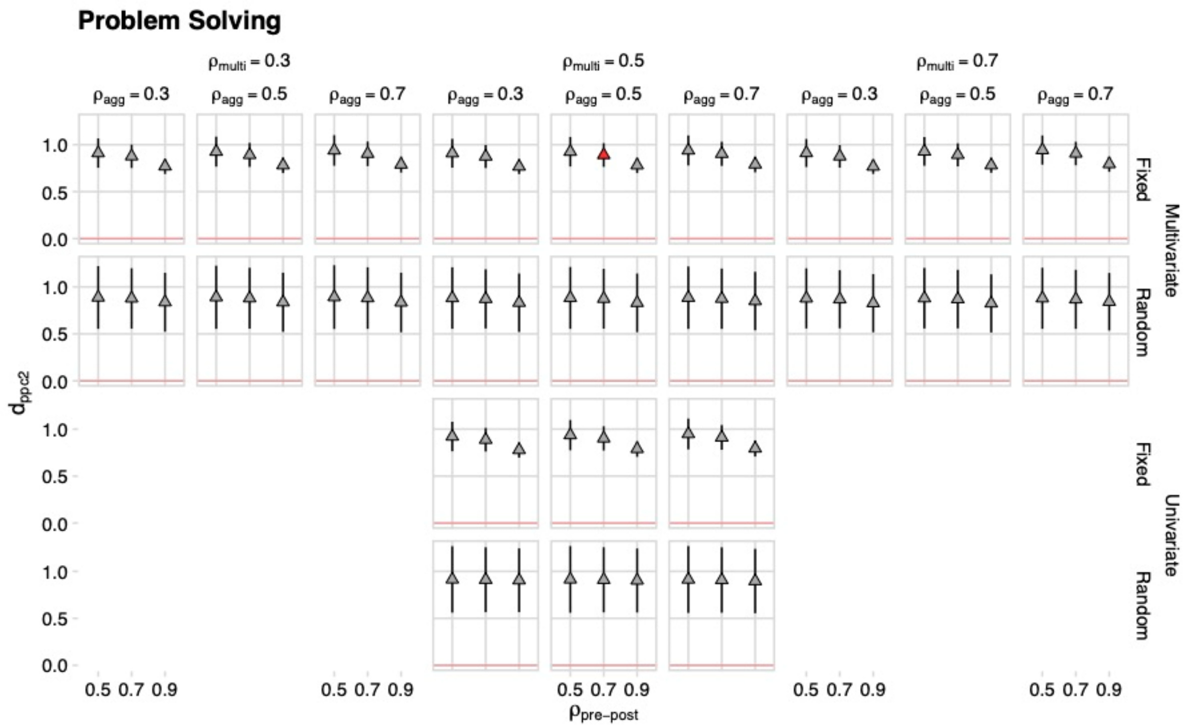


Note. Each individual effect is represented with the 95% confidence interval. The dotted line represents the null effect. For each outcome, the red diamonds depict the estimated average effect and the 95% confidence interval.

As explained in the previous sections, due to lack of information from published papers and the impossibility to have always access to raw data, we had to impute three correlations measures in order to compute the meta-analytic model. We also assessed the effect of imputing these correlations using a *multiverse* approach (Steegen et al., 2016). The main idea is to present a collection of different analytical possibilities. In the supplemental material, we present the effect of choosing a specific meta-analytic model (*fixed* vs. *random effect* and *univariate* vs. *multivariate*) varying also the imputed correlations. As an example, Figure 1.3 presents the multiverse analysis for problem-solving. The example shows how choosing a certain model and correlations value does not affect conclusions about problem-solving. This strengthens the main meta-analytic result.

Figure 1.3

Multiverse Analysis for Problem Solving



Note. The x-axis represents the pre–post ($\rho_{\text{pre-post}}$) correlation. On the top, there is the correlation between multiple effect sizes for the same outcome (ρ_{agg}) and the correlation between different outcomes (ρ_{multi}). The triangle and the bars represent the estimated effect and 95% CI. The red shape is the chosen combination for the meta-analysis. On the right are the different MA models. For the univariate model, we computed a *fixed* or *random effect* model ignoring the statistical dependence between outcomes.

Moderator analyses to statistically test the influence of age and intervention type were not possible, due to the reduced number of studies for each outcome.

Publication bias. Typically, the required number of studies for a publication bias assessment using funnel plot-based analyses is large (Furuya-Kanamori et al., 2020). Sterne and colleagues (2011) suggest however a minimum of 10 studies, which may increase though in presence of heterogeneity. Using selection models (see Hedges, 1984) the number of studies should be event higher (Jin et al., 2015). For these reasons we report the publication bias assessment using the Egger regression test (Egger et al., 1997) only for the *problem-solving* outcome that is associated with the highest number of effects (N = 9). We assessed the publication bias for the *univariate* fixed-effects model on *problem solving* with the chosen correlations' combination for the main meta-analysis model. The Egger regression is a meta-regression model estimating the relationship between sampling standard error and effect sizes. In the presence of publication bias (i.e., asymmetry in the funnel plot) the slope should be significantly different from zero. Furthermore, the intercept is usually interpreted as the estimated effect size with standard error close to zero. We computed the Egger test using the *regtest* function from the *metafor* package of R. The relationship between standard errors and effect sizes is not significantly different from zero ($z = 0.245, p = 0.807$) and the estimated *problem-solving* effect with zero standard error is 0.833 ($95\%\ CI = [0.268, 1.397]$). Overall, the Egger test suggests the absence of evidence for publication bias for the *problem-solving* outcome.

1.5 Discussion

The systematic review and meta-analysis aimed to address three key questions regarding the teaching of computational thinking: (1) which specific EFs are most influenced by it? (2) does the impact of CT programs differ based on children's age? and (3) which instructional modality (educational robotics, unplugged coding, or virtual coding) is most effective in enhancing EF skills in children and adolescents? The first research question was answered both by the systematic review and meta-analysis. The second and third research questions were exclusively tackled through the systematic review of the literature.

Unlike previous meta-analytic studies (Liao, 2000; Scherer et al., 2019), we exclusively focused on experimental trials. This approach allowed us to assess the cognitive effectiveness of CT interventions while accounting for potential confounding factors such as repeated testing and practice effects. While this decision improved the quality of our overall conclusions regarding the cognitive effects of CT, it also restricted the number of studies available for conducting a statistical analysis of the moderating effects of age and type of intervention. As a result, the conclusions on age and type of intervention effects are based on a qualitative analysis of the literature.

1.5.1 Which EFs are most influenced by the teaching of CT?

The review of the experimental studies showed that CT interventions are generally very effective in boosting children and adolescents' higher order and core EF skills. The largest effects were observed on children's problem solving and complex EFs such as planning, but significant positive effects emerge also for core EFs, like cognitive inhibition and working memory.

The meta-analysis provided statistical evidence to the qualitative conclusions of the systematic review, comparing the effectiveness of CT interventions across cognitive skills. Its results confirmed that problem-solving and planning were the cognitive skills that CT programs most strongly affected, indicating that children's inhibition and WM skills were influenced too by these interventions, albeit to a lesser extent.

Most studies reported *far transfer* effects to problem-solving tasks that are different from those trained by the intervention. Our meta-analysis provides thus further support for the transferability of the skills acquired through CT programs to situations that require problem-solving abilities, finding however a larger transfer effect (d_{ppc2} = .89) than in previous meta-analyses (Scherer et al., 2019), both in comparison with the *near* and *far transfer* effects (respectively, $\bar{g}$ =. 75 and $\bar{g}$ =. 47). Planning is the other EF that reflects most the cognitive effects of CT interventions (d_{ppc2} = .36), and in this case, the effects are similar to those reported for other higher order EFs (i.e., reasoning) in Scherer et al.'s study (g = .37).

The impact of CT interventions on children's problem solving skills is large both when children's problem solving skills are assessed by tasks similar to those proposed in the intervention programs, such as coding problems (*near transfer, e.g.,* Arfé et al., 2019a), and by very different tasks, like the problem solving tasks of PISA assessment or self-report scales (*far transfer,* e.g., Akcaoglu and Koehler, 2014; Erol and Çırak, 2021). In addition, these effects emerge also in studies with active control groups, that is when the effects of CT programs are compared to those of STEM or programming activities (e.g., Arfé et al., 2019; Erol & Çırak, 2022).

CT programs show also moderate *far transfer* effects on core EF skills such as cognitive inhibition and working memory, with effect sizes that are consistent across intervention studies employing different methods (Arfé et al., 2020; Di Lieto et al., 2020a). These transfer effect appear however weaker (d_{ppc2} = 0.17 for cognitive inhibition, d_{ppc2} = 0.20 for working memory) compared to those found for higher order EFs, such as planning and problem solving.

Planning and problem-solving are higher order EFs that closely relate to the kind of problem-solving activities involved in CT programs. Thus, they may better reflect improvements related to CT. As reported in other reviews and meta-analyses on the effectiveness of EF training programs (Diamond & Ling, 2016; Scionti et al., 2020), although the benefits of EFs training can transfer to untrained skills, the transfer effects appear to be typically narrow, and more pronounced for abilities similar to those trained. Thus, being more closely related to CT, planning skills can better mirror improvements in CT (Arfè et al., 2020), than working memory or inhibition skills (Arfé et al., 2019a; Di Lieto et al., 2020a).

On the other hand, foundational EFs such as inhibition and working memory may show less stability over time and be more influenced by temporary psychological and physiological states, like fatigue or stress, because they depend more on executive control processes than planning and problem solving. For planning and problem solving the acquisition of effective metacognitive strategies can be more critical (La Paglia et al., 2017). Other meta-analyses have shown that even when cognitive trainings are specifically focused on core EF skills, like WM and response inhibition, intervention effects can be small or non-significant, especially when far transfer effects are considered (Kassai et al., 2019; Melby-Lervåg & Hulme, 2013). Inhibitory control in particular seems resistant to EF interventions, showing non-significant effects to EF computer-based trainings (Cao et al., 2020), and small effect sizes ($g^+ = .18$) even when the training focuses on inhibition skills (*near transfer,* Kassai et al, 2019).

1.5.2 Does the Cognitive Effectiveness of CT Vary with Children's Age?

CT interventions seem equally effective for older and younger children. Beneficial effects of CT programs on problem solving are consistently reported across studies involving older (4th to 10th graders) or younger (preschoolers-2nd graders) participants who received comprehensive CT interventions targeting various components of CT, regardless of the specific type of intervention (virtual coding or ER; e.g., Erol & Çırak, 2022; La Paglia et al.,2017).

Older students participating in CT programs improved in performing problem-solving tasks (Akcaoglu & Koehler, 2014; Lai & Yang, 2011), in self-reported problem-solving (Erol & Çırak, 2022), and also developed metacognitive skills involved in problem solving (La Paglia et al., 2017).Among the few studies (three) that did not find significant positive effects of CT programs for these older participants, one (Oluk & Saltan, 2015) proposed a narrow intervention program, centered exclusively on algorithm development, and the other two (Çınar & Tüzün, 2020 and Özcan et al.,2021), involved the students in structured ER or visual programming activities.

For the younger children, significant transfer effects, from moderate to large, were observed both for complex EF skills (Akcaoglu & Koehler, 2014; Nam et al., 2019) and core EFs, such as WM and cognitive inhibition skills (Arfé et al., 2019a; Arfé et al., 2020; Di Lieto et al., 2020a), although for the latter effect sizes were small.

Given that all the studies assessing inhibition and working memory in this systematic review and meta-analysis focused exclusively on younger children, aged 5-7, age could account for the limited *far transfer* effects on these EFs. In our meta-analysis the moderating effects of age could not be tested owing to the small sample size. However, other recent meta-analytic studies that considered the effects of EF training programs on young children's (3 to 6-year-old, Scionti et al., 2020) or the transfer effects of programming across a larger age-span (from prekindergarten to college students, Scherer et al., 2019) found that age was not a significant moderator of the effects observed. Despite these meta-analytic findings, age related effects remain a pedagogically relevant research question that deserves more attention, especially if we consider the different plasticity of core EFs across child development (Scionti et al., 2020; McGuckian et al., 2023).

1.5.3 Which Instructional Modality (Educational Robotics/Virtual Coding) is most effective in enhancing Children and Adolescents' EF Skills?

Based on our systematic review the different effects of CT activities seemed not to be strictly related to the method or programming tool used (virtual coding, unplugged coding or ER), but rather to the structured or unstructured nature of the intervention and the age of the participants.

Structured versus unstructured interventions. A qualitative comparison between the effectiveness of structured and unstructured CT intervention programs was possible only for the older age group (grades 4 to 10). Game-design or project-development intervention programs, that consisted in open-ended problems, resulted much effective in enhancing the problem-solving skills and self-reported problem-solving abilities of these older learners (Erol & Çırak, 2022; Nam et al., 2010). Structured interventions, whether alone or combined with unstructured project development, appeared to be less effective (Çınar & Tüzün, 2020; Özcan et al., 2021), except when the intervention addressed students' mathematical problem solving (La Paglia et al. 2017) or was for students with intellectual disabilities (Demir, 2021).

Younger children received only structured interventions. For them, structured interventions, based logic games with a sole correct solution, proved effective both in boosting problem solving and EF skills (Arfé et al., 2020; Di Lieto et., 2020a). Among the structured programs that involved younger children, visual programming and ER tools were equally effective. The specific effects of virtual coding and tangible (ER and unplugged) coding interventions are discussed below.

Virtual coding. The results of the systematic review show that virtual coding interventions, when appropriately designed, result effective for students in a large age span, from grade 1 to 8. Twelve of the experimental studies examined in this systematic review tested the cognitive effects of virtual coding interventions, and nine of them (75%) reported significant positive effects on problem-solving and other EFs.

CT involves a complex set of cognitive abilities, such as memory, self-regulation, and planning, which are also involved in problem-solving (Frensch & Funke, 1995; Keen, 2011). These skills, as well as the development of higher order EF, including problem-solving skills, develop significantly from elementary to middle school (Brocki & Bohlin, 2004; De Luca et al., 2003; Luciana & Nelson, 2002; Luna et al., 2004), and further on during adolescence (Hooper et al., 2004; Unterrainer & Owen, 2006; Zelazo et al., 1997). For younger students, who are still developing foundational EF (Gathercole et al., 2004; Klenberg et al., 2001; Jonkman et al., 2003) structured virtual coding interventions enhance both problem-solving skills similar to those trained during the intervention (*near transfer*), and EFs less closely related to coding, such as planning and inhibition skills (*far transfer*). For late elementary school students and adolescents who have already developed the foundational EF skills (e.g., inhibitory control, working memory) and metacognitive abilities that are necessary to manage complex coding tasks (Best & Miller, 2010; Conklin et al., 2007; Gathercole et al., 2004; Klenberg et al., 2001; Jonkman et al., 2003), practicing with ill-defined, or unstructured, coding or programming problems, such as those involved in game-design or creative problem solving activities, results most effective. The finding that unstructured virtual coding activities seem to work better with older students should not be surprising. As Diamond et al (2016) have emphasized EFs need to be challenged, not just used, to promote improvement.

Educational robotics. Based on a concrete sensory-motor experience, ER or unplugged coding activities, are more often addressed to younger children, that is, preschoolers or first graders, or students with cognitive impairment (Demir 2021; Di Lieto, 2020a, 2020b; Nam et al., 2019). Out of the 19 studies included in this systematic review, only two experimental studies (Çınar & Tüzün, 2020; La Paglia et al., 2017) specifically examined the effects of ER interventions on late elementary school or middle school students, producing contrasting findings. La Paglia et al., (2017), found positive *far transfer* effects of ER on fifth graders' problem-solving skills. Çınar and Tüzün (2020) found no significant *far transfer* effects of ER to tenth graders' self-reported problem-solving.

In the younger, 5–6 year-old children, ER interventions seem to have general positive *far transfer* effects on EF skills, improving significantly children's problem-solving skills (Çakır et al., 2021; Nam et al., 2019), as well as core EFs (response inhibition and working memory, Di Lieto et al., 2020a; Di Lieto et al., 2020b). The benefits of ER seem however reduced for children with special educational needs (children with sensory, motor, or cognitive disabilities, or attention deficit hyperactivity disorder, and/or specific learning disorders, Di Lieto et al., 2020b). For these children, tangible, unplugged coding activities, seem more effective (Demir, 2021). A possible explanation is that although ER can sustain EF through concrete and tangible activities, it requires basic computer skills and memory and cognitive resources that students with intellectual disabilities may lack (Demir et al., 2021; Di Lieto et al., 2020b).

Since all studies focused on ER interventions also involved structured problem-solving activities, it is also difficult to determine whether their beneficial effects for younger children were due to the tangible (ER) or structured nature of the intervention. In fact, when narrowing our analysis to younger, 5-6 year-old children, structured virtual coding and ER activities seem to have equivalent *far transfer* effects on core EFs (Arfé et al., 2019; Arfé et al., 2020; Di Lieto et al., 2020a) and higher order EF skills, like problem solving (Arfé et al., 2019; Nam et al., 2019). These findings seem in contrast with those of Scherer et al. (2020), who report larger effects of instructional programs based on physicality, such as robotics ($\bar{g}$ = .72) than of virtual coding ($\bar{g}$ = .44), suggesting that the medium used in the instructional intervention can play a role. Direct comparisons between structured virtual coding and ER interventions are needed to test this hypothesis and could be a goal of future studies.

1.6 Conclusions

Over the last 20 years, computer scientists, experts in education, and psychologists have explored the cognitive effectiveness of CT-based activities or instructional programs, primarily in the domain of children's EFs (e.g., Arfé et al., 2020; Brown et al., 2008; Di Lieto et al., 2020a; Lai & Yang, 2011). However, there have been very few systematic reviews and meta-analyses that synthetized this literature (Liao, 2000; Liao & Bright, 1991; Scherer et al., 2019; Scherer et al., 2020). Systematic reviews and meta-analyses constitute complementary and fundamental tools of evidence-based practice, particularly when the focus of interest is on the evaluation of intervention effects, like in this study.

Overall, the systematic review and meta-analysis presented in this paper confirm that CT programs that most countries are recently integrating in their school curriculum (Lye & Koh, 2014), can be a powerful tool to boost and support the development of EFs, particularly higher order EFs like planning and problem solving, but also core EFs, like response inhibition and WM that underpin and predict both early and late academic achievements (Clark et al., 2010; Jacob & Parkinson, 2015; Spiegel et al., 2021).

Despite variation in outcomes across EFs, 14 of the 19 studies considered in this systematic review (74%) reported evidence of EF benefits. This is a higher percentage than that reported by Diamond & Ling (2016) for cognitive trainings (from 20% to 60%) and aerobic exercises (43%). While, it is remarkably similar to the effectiveness of other school programs, such as Montessori and Tools of the Mind (75%). As for these programs, an early integration of CT/coding in the school curriculum may be strategic for children's future academic accomplishments.

School embedded programs are comparatively more effective than cognitive training interventions targeting EFs (Diamond & Ling, 2016). When the training is part of children's daily school curriculum its activities become meaningful to children, and children may better perceive the importance and utility of the abilities learnt, their social value, and their transferability to other learning situations.

The finding of a beneficial effect of CT, especially on higher order EFs, is also important for another reason. Although a number of intervention studies have focused on improving EF (e.g., Passolunghi & Costa, 2016; Pozuelos et al., 2019; Schmitt et al., 2015), few have demonstrated effects on higher order EF (Diamond & Ling, 2016; Scionti et al, 2020). Enhancing these higher-order cognitive skills, also bears on a child's adaptive capacities to the social environment (Barkley, 2001; Huepe et al., 2011).

Our systematic review found that the effects of CT/coding did not seem to vary by the training method (virtual coding/unplugged/ER). Comparisons between training methods is however limited by the fact that different types of interventions have been typically addressed to different age-groups. The lack of direct comparisons between virtual coding and ER or structured and unstructured intervention conditions stands out as a prominent limitation in the existing literature. Another notable gap lies in the scarcity of studies examining the efficacy of CT activities in enhancing core executive functions (working memory, response inhibition, cognitive flexibility) in older students. CT programs may be particularly suitable to support the development of core EF skills even during late elementary schools and adolescence.

1.6.1 Limitations and Future Directions

In addition to the limitations inherent in the existing literature, it is important to acknowledge some limitations of the present meta-analysis. The first concerns the lack of complete data from published studies. For computing the appropriate meta-analytic model, we needed several correlation values often omitted in the papers selected. We addressed this problem by employing the multiverse-like sensitivity analysis that clearly shows when a certain imputed correlation has a relevant impact on the meta-analysis model.

A second limitation of this systematic review and meta-analysis is that by applying rigorous inclusion criteria, we could include a limited number of studies. For instance, as one e of the inclusion criteria for this systematic review was the peer-review publications, we did not consider the grey literature. Consequently, we were not able to robustly assess publication bias or consider important moderators, such as age and type of intervention, in the meta-analysis. With the growing number of experimental studies in the field of CT, future meta-analyses will be able to compare the effectiveness of different types of CT intervention, or how it varies with students' age. Finally, other variables could influence the effectiveness of a coding training and should be considered in future studies; examples are motivational factors like self-efficacy (Tsai, 2019), or gender (Montuori et al., 2022). These variables may have an important role in moderating children's response to coding interventions. The spread of experimental research in the area of CT studies, and the interdisciplinary collaboration between computer scientists and psychologists will soon allow to address also these factors.

EXPERIMENTAL STUDIES 2 AND 3

The systematic review and meta-analytic investigation conducted on the existing literature has highlighted a notable limitation in the current body of research concerning cognitive training (CT) interventions. The key findings provided by the systematic review and meta-analysis on the teaching of computational thinking (CT) and its impact on executive functions (EFs) revealed that CT interventions are generally effective in boosting higher-order and core EF skills in children and adolescents. However, intervention studies focused on preschoolers' population are lacking in the research field. Specifically, no studies have so far investigated the impact of CT interventions on the preschoolers' core EFs. Additionally, there is a lack of direct comparisons of age-related efficacy. The synthesis of available studies has revealed a substantial lack of empirical studies that allow for the comparison of the effects of CT interventions across different age levels.

Elucidating age-related nuances in the outcomes of CT interventions is paramount to advancing our understanding of the relationship between cognitive training and age-related effects of the CT interventions, and ultimately to informing more targeted and effective interventions across the lifespan.

The present PhD project aimed to address these research gaps by (1) investigating the effects of coding activities on executive functions and coding skills in preschool children and (2) testing the age-related effectiveness of plugged coding on children's executive functions and coding abilities.

Study 2, presented in Chapter 2, aimed to explore the effectiveness of tangible coding on the cognitive skills of development preschool children. Little is known about the effects of Educational Robotics (ER) interventions, alone or combined with unplugged coding activities, on preschoolers' CT and EF skills. In a cluster randomized controlled trial, we have assessed how gains of preschoolers in coding skills following interventions based on combinations of unplugged coding ER, transfer to plugged (computer-based) coding abilities and to EFs such as planning, response inhibition, and visuo-spatial skills. Forty-seven (47) preschoolers with no prior exposure to coding, were randomly assigned to an experimental (unplugged coding and ER, 2 classes) or control (standard school activities, 2 classes) instructional condition.

Study 3, presented in Chapter 3, aimed to assess children's response to coding intervention at different age levels (i.e., first and fourth grade). Past research has shown that children as young as 4-6 years can learn to code, and that being exposed to structured coding activities has also beneficial effects for their cognitive abilities, in particular two EFs, planning and response inhibition, which have been found to be strongly associated with coding. Our recent meta-analysis (Montuori et al., 2024) has shown that virtual coding interventions were also effective for older students (from 4th to 10th graders), with benefits observed in complex problem-solving tasks for late primary school students and adolescents. Few studies (Atmatzidou & Demetriadis, 2016; Jiang & Wong, 2022; Kyza et al., 2022; Papavlasopoulou et al., 2017; Rjike et al., 2018; Tsarava et al., 2022), mostly focused on older students (aged 10 to 18 years), have examined age-related differences in response to coding instruction. Most of them studies have shown that age affects the acquisition of CT, revealing that older students achieved higher coding skills than younger students trained by CT interventions (Jiang & Wong, 2022; Kyza et al., 2022; Rjike et al., 2018; Tsarava et al., 2022). However, results regarding the effect of age on CT training outcomes have not always been consistent. For instance, Jiang & Wong (2022) reported that younger children (fourth graders) showed the higher response and benefited more from a virtual and unplugged coding intervention than older children (i.e., fifth and sixth graders), while Atmatzidou & Demetriadis (2016) did not find any age-related differences in effectiveness. Moreover, none of these studies have explored whether age differences in learning to code translate also in differences in the cognitive benefits of coding. Past research has shown that children's EFs develop significantly between the age of 5-6 and the age of 9-10 years, and this may also affect children's response to coding interventions. Thus, age-related difference could be expected in response to coding intervention between children of these ages. Study 3 tested this hypothesis.

CHAPTER 2. THE EFFECTIVNESS OF CODING INTERVENTIONS ON PRESCHOOLERS' CODING AND COGNITIVE SKILLS [2]

2.1 Introduction

One in three 13-year-old students in Europe currently lacks the basic digital skills and problem-solving skills that are so much needed in modern societies (OECD, 2017; PISA 2015). Computational thinking (CT) learning programs address this need. A clear drive traverses' governments in Europe, USA, and other continents to integrate the teaching of CT in school curricula from early grades. Nonetheless, robust evidence is still lacking on what be the most effective way to introduce young children, totally novice to coding, to the concepts and processes of computational thinking. Although several studies are emerging in the field, researchers have been mainly concerned with testing the effectiveness of interventions addressed to primary school children or older students (e.g., Arfé et al., 2019; Brackmann et al., 2017; Sun et al., 2021). Computational thinking, which promotes a problem-solving attitude inspired by computer science, encourages students to break down complex problems into smaller, more manageable parts, to recognize recurring problem-and-solution patterns in them, and to develop algorithmic solutions to them that can be executed by an external agent. This aggregate of skills fosters a systematic and organized approach to problem-solving (Wing, 2006). Coding, which stimulates the skills to create, modify, and evaluate program text, fragments, symbols, and the familiarity with programming concepts and procedures, is one of the instruments of CT. Introducing

[2] The present chapter has been published in Education Sciences in the form of the following research paper: Montuori, C., Pozzan, G., Padova, C., Vardanega, T., Arfé, B. (2023). Combined Unplugged and Educational Robotics training to promote Computational Thinking and Cognitive Abilities in Preschoolers. *Education Sciences*, 13, 858. https://doi.org/10.3390/educsci13090858 Published under a CC BY LICENSE
invited by Education Sciences, special issue topic: "Robot Programming in Early Childhood and Primary Education."

CT from preschool years equips students from the onset of their education with skills and mindsets that are deemed crucial to operating in the modern world, such as digital skills, logical reasoning, and creativity (Shute, 2017). In addition to that, early exposure to CT lays a strong foundation not only for the development of other 21st century skills, like problem solving, but also aids future learning achievements in several disciplines, particularly Science Technology Engineering Mathematics (STEM) (Leonard et al., 2016). In Italy, the teaching of CT from preschool became compulsory with a law issued by the Ministry of Education (motion n. 1-00117, 12 March 2019). However, similarly to what happens in other European and extra-European countries, the teaching practices of CT in Italian preschool and primary schools are still far from being homogeneous, as many teachers lack the fundamental training and knowledge to achieve this goal. In fact, in Italy CT is currently absent from university curricula for primary school teachers in training.

According to literature, the tools used to teach coding elements of CT skills during the early school years are: (1) unplugged coding, that is, programming without the use of digital devices; (2) educational robotics (ER), where learners give executable instructions to a programmable robot in order for it to perform specific actions in a physical environment (Chen et al., 2017); and (3) plugged coding, which entails developing executable programs to instruct a digital sprite to achieve a goal in a constrained digital environment (Arfé et al., 2019; Fessakis et al., 2013; Kalelioğlu, 2015; Zhang et al., 2014). Although these tools are all used to introduce children to coding, they have different characteristics that make them more or less suitable for different age groups. Plugged practices make use of computing devices, while unplugged practices do not. Unplugged coding activities involve logic games, cards, strings, or physical movements that are used to represent and understand CT concepts, such as algorithms, without requiring abstract coding (Brackmann et al., 2017; Campbell & Walsh; 2017; Lee & Junoh, 2019). Although the efficacy of these tools for learning to code has been tested in some studies, very few experimental studies involved preschoolers, and none of them, to the best of our knowledge, tested the effects of *combined* unplugged and ER interventions. Moreover, none of these studies examined the transfer of ER or unplugged coding skills to children's cognitive abilities except for problem solving and visuo-spatial skills.

Whether acquired knowledge and skills can be transferred from one context or problem to another is a key question for educational and cognitive psychologists. In fact, the transfer of learning effects across contexts, tasks, and abilities lies at the very heart of education and in the concept of learning itself, involving the ability to the flexibly apply what has been learned (Barnett & Ceci, 2002). Transfer in learning means that learning in one context impacts learning and performance in other contexts. The transfer of abilities depends on the analogy and overlap between the contexts and problems in which the skills were gained and those presented later (Schunk, 2012). State-of-the-art research suggests distinguishing between near-transfer and far-transfer. Near-transfer is the transfer between the tasks or skills trained and similar or closely related tasks or skills; far-transfer is the transfer between dissimilar tasks or skills, and thus is considered more difficult (Perkins & Salomon, 1992). In the CT context, near-transfer effects are assessed by examining the effects of coding intervention to various guises of situations that all explicitly require programming skills (i.e., coding abilities). Far-transfer effects are instead assessed by examining transfer to tasks that require skills less closely or directly related to programming, such as cognitive abilities like response inhibition or planning, which may be used in coding but are assessed through tasks very different than those used during coding interventions (Arfé et al., 2019; Scherer et al., 2019).

Examining these far-transfer effects of coding is particularly important at the transition from preschool to primary school, as the test of far-transfer effects evaluate the general cognitive benefits of coding, that is the possibility that learning to coding may not only be useful to train 21st century digital skills, but also prove effective to foster children's cognitive arsenal. This is even more important during the preschool years, as the preschool period is characterized by greater "neural plasticity" in which the windows for brain development are all wide open and it is comparatively easy to lay down neural pathways for new skills (Brown & Jernigan, 2012; Kolb et al., 2017; Johnston, 2009). Gained knowledge about the effectiveness of coding in this time window would allow age- and cognitive developmentally appropriate programs for preschoolers to be incorporated into early teaching curricula with potentially cascading benefits for their learning.

2.1.1 Evidence about near-transfer and far-transfer effects of coding during preschool years.

Several studies conducted with preschoolers have so far demonstrated the feasibility of teaching coding through unplugged activities (Lee & Junoh, 2019) and ER (Bakala et al., 2021; Bers et al., 2014; Critten et al., 2022). However, a recent systematic review highlighted a limited number of research studies on the effectiveness of ER to develop CT skills of young learners (Ching & Hsu, 2023). Indeed, among the studies that have explored the learning of CT skills through ER (e.g., Relkin et al., 2021), few focus on preschoolers (Angeli & Valanides, 2020; Kazakoff et al., 2013; Fu et al., 2023; Roussou & Rangoussi, 2020). A recent meta-analysis by Li et al. (2022) highlighted that even among the experimental studies focused on the effectiveness of unplugged coding to promote CT skills (e.g., Brackmann et al., 2017; Jun et al., 2017; Zhan et al., 2022), none involved preschoolers (Li et al., 2022). Another limitation of the research field is that only a limited number of studies exploring the effects of CT interventions for younger learners are experimental studies proper. For instance, just one of the ER studies mentioned above is an experimental study (Fu et al., 2023). That study found a positive near-transfer effect of 12 hours of ER activities on the algorithmic skills of 42 preschoolers aged 5-6 years, with a medium to large effect size ($d = 0.77$). The sequencing abilities were however not significantly influenced by the ER intervention (Fu et al., 2023). Another experimental study demonstrated positive effects of unplugged coding on CT abilities (Zhan et al., 2022). However, the intervention addressed older participants (6-8 years).

Although these studies suggest positive effects of instructional interventions based on ER on preschoolers' coding abilities, their focus is on near-transfer effects only, that is on the effects of the interventions on coding itself. Additionally, the quantity of studies is still too small to inform truly evidence-based instructional practice.

Other experimental studies (e.g., Arfé et al., 2019; Di Lieto et al., 2020) have demonstrated that learning CT in educational settings since grade 1 leads also improves children's executive functions (EFs), which shows far-transfer effects of coding.

EF is an umbrella term that refers to a complex set of cognitive skills related to goal setting and the performance of goal-directed behaviors. Welsh & Pennington (1988) defined EF as the ability to maintain an appropriate problem-solving set for achieving a goal. This capacity encompasses core skills such as children's command over impulsive responses (i.e., inhibitory control), their ability to update their working memory (WM), and their skill to switch perspectives and shift attention between mental sets or tasks (i.e., cognitive flexibility, Diamond, 2013; Miyake et al., 2000; Thayer & Lane, 2000). These three cognitive functions (i.e., response inhibition, working memory, and shifting) are defined core EFs (Lehto et al., 2003; Miyake et al., 2000) from which higher-order executive functions, such as planning and problem-solving originate (Diamond, 2013; Pennington & Ozonoff, 1996; Zelazo et al., 1997, 2003). EFs skills underpin learning and academic performance across several domains (Clark et al., 2010; Jacob & Parkinson, 2015; Spiegel et al., 2021).

The preschool period that accompanies the transition to primary school is particularly sensitive for the development of EFs (Best & Miller, 2010; Vandenbroucke et al., 2017). Response inhibition skills develop rapidly in the early school years: inhibition is already present in the first year of life; it undergoes a considerable increase in the preschool period (Garon et al., 2008), and continues to develop throughout childhood and adolescence. Research has shown that children response inhibition improvement occurred from age years 3 to 4 (Carlson, 2005; Zelazo el al., 2003). Other findings revealed significant improvement from age years 3 to 6 (Klenberg et al., 2001). Also, planning skills seem to develop significantly in the first years of schooling (Mägi et al., 2016; Poutanen et al., 2016). It is therefore held that interventions aimed at boosting the development of EFs are particularly effective in time window of preschool (Scionti et al., 2020).

For example, researchers found that both plugged coding (Arfé et al., 2019, 2020) and ER (Di Lieto et al., 2020) significantly affect first graders' EFs (response inhibition, working memory, and planning). Whereas research on the deployment of coding instruction in preschool, primary, and secondary school is growing rapidly (Bati, 2021; Zhang & Nouri, 2019), the lack of experimental studies testing the effectiveness of these activities on preschoolers' cognitive development represents a significant gap in the research field. Just three experimental studies have tested the effectiveness of ER on preschoolers' cognitive development, with focus on problem-solving skills (Çakır et al., 2021; Nam et al., 2019) and inhibitory control task (Yang et al., 2022). Results showed that ER enhanced children's problem-solving skills, assessing that achievement through an ad-hoc problem-solving skills scale, where children were asked to make drawings representing the solution to a problem presented in a picture (Çakır et al., 2021), and mathematical problem-solving tasks (Nam et al., 2019). Those experiments reported that the problem-solving skills improved significantly, thereby achieving far-transfer effects. Conversely, a cluster randomized trial that involved a large sample (101 kindergartens, mean age = 64.78 months) showed that ER was not effective in boosting preschoolers' self-regulation (i.e., performance in an inhibitory control task) (Yang et al., 2022). The (active) control group was exposed to block-play activities that engaged inhibitory control skills by having children strategically determine which building blocks to use. The block-play intervention entailed constructing a tower with wooden blocks. The learning strategy encouraged children to use blocks with a goal-oriented mind, to think deeply before making moves, to solve problems, and to cooperate in groups. The findings reported by Yang and colleagues contrast with the results of a recent study

(Canbeldek & Isikoglu, 2023) about the far-transfer effects on preschoolers' self-regulation skills of a training intervention that combined unplugged coding, ER, and plugged coding. The cited study found significant far-transfer effects on inhibitory control assessed through the same self-regulation test used in Yang et colleagues' study (i.e., the Head-to-Toes Task, Cameron Ponitz et al., 2008) and on problem-solving skills. Effect sizes were not reported. Whereas the aims of study are innovative as they test the effects of *combined* training intervention, several experimental details are missing (modality and duration of each intervention), whose omission makes the instructional design obscure and not replicable. The plugged coding activities that were performed in that study were unstructured and employed Scratch, which may be unsuitable for preschoolers. Arguably, plugged coding requires more abstraction skills than unplugged coding, and ER favors a more concrete programming experience. Unplugged and ER are therefore especially suitable tangible tools for introducing preschoolers to coding.

Besides the above-mentioned effects on children's EF, ER and coding interventions also seem to have significant effects on children's visuo-spatial skills. Visuo-spatial skills, which involve the mental manipulation and understanding of visual stimuli in relation to objects and their positions, are involved in visual coding tasks. Unsurprisingly, several studies highlighted the correlation between visuo-spatial skills and programming performance (e.g., Parkinson & Cutts, 2019). Some authors found that students with stronger visuo-spatial skills exhibited greater programming aptitude and problem-solving abilities (e.g., Uttal et al., 2013). These findings suggest that spatial skills contribute to the comprehension of complex code structures, visualization of program execution, and debugging, all of which are crucial aspects of successful education to programming. However, since coding involves visuo-spatial skills, it is also possible that performing coding activities, especially unplugged coding, or ER in a physical environment, could improve children's visuo-spatial abilities.

Recent research (Brainin et al., 2022) examined whether coding activities in a tangible environment enhances preschoolers' visuo-spatial skills. The study involved 84 preschoolers aged 66.23 months divided in an experimental group (ER), an active control group (unplugged coding), and a passive control group. Children's visuo-spatial abilities were assessed before and after a series of 10 intervention sessions. Sub-tests from the Test of Visual Perceptual Skills (non-motor) revised (TVPS) battery (Gardner, 1996) were used to measure spatial relations and visual memory. Mental rotation skills were assessed by the spatial rotation test (Johnson & Meade, 1987). The test includes two example trial items and 16 test items. The items are presented on paper. For each item, a part of a square is presented, and on the right, four similar figures are presented. The children are asked to choose one out of the four figures that would complete the square. Although findings indicate that both unplugged coding and ER improved spatial-relations and mental-rotation skills, the effect size was higher in the group exposed to ER (eta-squared= .58) than in the unplugged coding (eta-squared= .14). This result suggests that ER interventions may be efficacious in enhancing visuo-spatial skills in preschoolers. The ER and unplugged interventions differed only in the type of tool used. The length and the CT concepts learned were the same for the two groups. The ER group used the programmable robot (i.e., *BeeBot*) to execute the coded programs, while the unplugged group was asked to carry out the same tasks with toys such as cars or dolls that the children had to move. No other studies that we know of have tested the effects of ER or unplugged coding on children's visuo-spatial skills.

In summary, although emerging research results show significant effects of ER and unplugged coding on preschoolers' cognitive skills, the relevant studies are still too limited, and most are focusing on one or two cognitive skills. During preschool years both visuo-spatial abilities and EFs undergo significant developmental changes. Hence, testing the effects of coding across these cognitive skills becomes important to design coding interventions aimed both at teaching CT and training cognitive abilities. Intense development of three EFs (response inhibition, working memory, and cognitive flexibility) occurs from 3 to 5 years of age (Best & Miller, 2010; Garon et al., 2008). Several intervention programs for empowering EFs are initiated as early as preschool age (Scionti et al., 2020). If learning to code were proven to improve the development of such EFs, it should also be included in preschool activities. It is therefore important to test the truth of this conjecture.

As noted above, prior research has demonstrated the efficacy of a tangible coding intervention via ER in first grade (Di Lieto et al., 2020). In the present study we aim at extending this research to younger children, testing the effects of a combined ER and unplugged coding intervention on the acquisition of coding skills as well as of three cognitive abilities: response inhibition, planning, and visuo-spatial skills.

2.1.2 The present study

In this exploratory study we conducted a cluster-randomized controlled trial on the effects of a combined intervention involving unplugged coding and ER on preschoolers' coding abilities, and the far transfer of the coding intervention effects to cognitive -EFs and visuo-spatial- abilities. We decided to plan a combined -unplugged coding and ER- intervention for several reasons. Firstly, although ER has the merits of being an interactive tool, of a tangible nature and therefore particularly suited to the sense-motor experience typical of pre-school age, ER is also a complex tool for younger children, especially for those without prior experience of coding. In addition to the learning demands of coding children with no knowledge of programmable robots need to acquire a certain amount of instrumental and command skills to manage the robot. All these stimuli make demands on younger children's memory and cognitive system, which induce children to focus on those demands rather than on programming concepts and processes. Thus, we adopted a more gradual approach to introduce coding to preschoolers, starting with propaedeutic activities and unplugged coding to introduce fundamental CT concepts first (e.g., sequences and algorithms) without the additional demands of learning to use new digital tools. Next, we introduced robots and robot programming as an extension and generalization of the unplugged coding experience. We conjectured that this would have allowed children to consolidate the new concepts gradually, before experiencing programmable robots. An additional problem in using robotics with young children is that although perspective taking is crucial for ER as children need to take the robot's perspective (i.e., visual perspective and spatial perspective) to understand which instructions to give it to achieve the given goal. For younger children, taking others' perspectives can be cognitively difficult, and unfortunately ER or interacting with robots in general do not seem to help in the understanding of directions from the robot's point of view, as confirmed by a recent study (Marinus et al., 2018). The authors of that study argue that although the activities with the programmable robot (i.e., *Cubetto*) were engaging and challenging for children, perspective taking proved complex for the preschoolers. Indeed, several facilitations were needed to support children in the task (e.g., giving instructions while sitting behind the robot, moving the mat with all the materials around to ensure that the child kept facing robot's tail, putting red and yellow stickers on each child's right and left hand respectively, since many 3-6-year-olds cannot yet tell right from left, Marinus et al., 2018). To date, just one recent study tested the feasibility of both unplugged coding and ER intervention to teach CT skills to preschoolers (Critten et al., 2022). This trial first introduced unplugged coding activities and then the programmable robots. The ER intervention consisted of programming *Beebot*, a bee-shaped robot that is programmed with directional commands via input pressing buttons placed on its back. The robot executes the instructions when child presses "play" button. The authors found that programming *Beebots* was not as easy as expected for young children, who incurred difficulties in using the map grids required to create algorithms to program the robot. In addition to the cognitive difficulties of their use, educational robots are also expensive and not affordable by all schools and teachers. Thus, some authors have stressed the importance of testing also the effectiveness of unplugged coding activities. Unplugged coding has some advantages over ER and could be a first step to introducing young children to coding, allowing children to acquire such basic skills as distinguishing right/left, experiencing perspective taking in a natural-tangible environment.

The present exploratory study addressed the following research questions:

Near-transfer effect: Can a combined – unplugged coding and ER intervention be effective in teaching 4-5-years-old preschoolers' coding skills and CT processes?

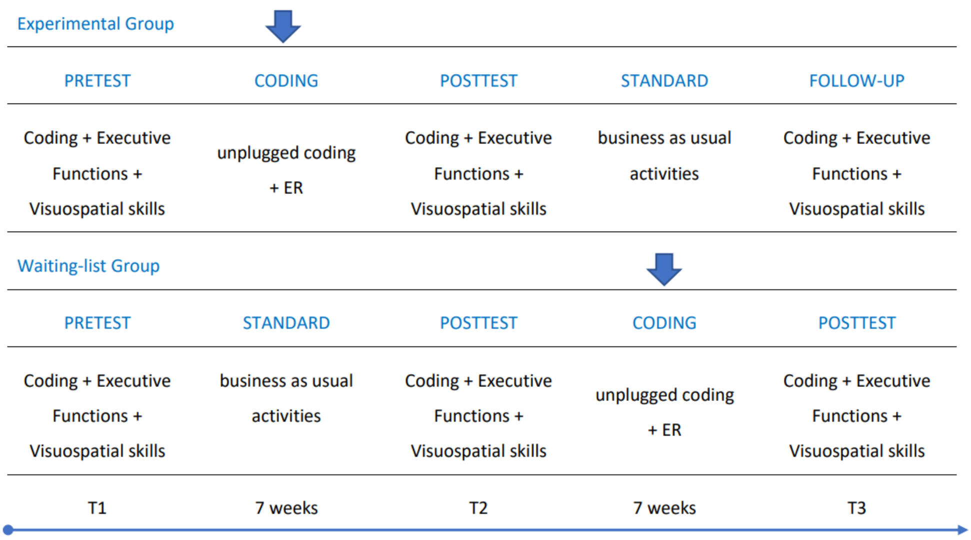
Far-transfer effects on cognitive skills: Do the positive effects of the combined – unplugged coding plus ER training transfer to 4-5-years-old preschoolers' response inhibition, planning, and visuo-spatial skills?

2.2 Method

2.2.1 Experimental design

The study was a cluster-randomized controlled trial (CONSORT guidelines, Campbell et al., 2004) with a standard pretest (T1) - intervention - posttest (T2) assessment for the experimental group, and business-as-usual activities for the control waiting list group between T1 (pretest) and T2 (posttest). This latter group received the intervention after T2. At T3 both the waiting-list and experimental groups were assessed again. For the waiting list group, T3 was the posttest, in which the effects of exposure to CT were tested against the effects of business-as-usual (standard) instructional school activities. For the experimental group, T3 was a follow-up measure of maintenance of the intervention effects. Four preschool classrooms were randomly assigned to the experimental (coding intervention) or control (business-as-usual) condition. After T1 (pretest), the experimental group, consisting of 2 preschool class groups, practiced with coding twice a week for seven weeks. The waiting list group, consisting of 2 preschool class groups from the same schools, performed business-as-usual activities for the same seven weeks, and practiced with coding after T2 (posttest). Such experimental design also allows the control group to benefit from potentially effective learning activities. For the children in the experimental group, follow-up data were collected at T3, two elapsed months after the posttest (see Figure 2.1).

Figure 2.1. Experimental design



2.2.2 Participants

Forty-seven 5-year-old preschool children from 4 class groups of the same school in northern Italy participated in the study. All children attended the last year of preschool. None of them had been exposed to coding before the intervention. The four class groups were randomly assigned to either the experimental or the waiting-list group.

The experimental/treatment group comprised 25 children (14 girls, 56%; 11 boys, 44%), 2 class groups, assigned to the treatment condition and participating in coding labs immediately after the pretest (T1). The waiting-list group included 22 children (10 girls, 43.5%; 13 boys, 56.5%), 2 class groups, assigned to business-as-usual activities and receiving the coding intervention only after the posttest (T2). Age ranged from 59 months to 70 months in the experimental/treatment group and from 58 to 70 in the waiting-list group. The mean age of the participants was 64.68 months (SD = 2.72) in the experimental group, and 64.82 months (SD = 3.84) in the waiting-list group. Children's socio-economic status (SES) was assessed by means of a socio-demographic questionnaire that parents returned along with the written informed consent to participate in the study. SES was estimated based on parents' education, on a scale from 0 (less than elementary school) to 4 (college), and occupation, from 1 (unemployed) to 4 (professional roles). A composite score was calculated as the sum of the highest education score and the highest occupation score obtained by either parent (Arfé et al., 2019), with maximum score 8. The mean SES was 7.04 (SD = 1.17) for the experimental group; and 7.23 (SD = 0.75) for the waiting-list. Familiarity with technology was also gauged by asking parents about children's daily use of digital devices (personal computers, smartphones, or tablets) in their home environment. A composite score was calculated as the sum of the (intensity of) use of those three digital devices. 20 children (80%) of the experimental group were not familiar with any type of mouse-controlling device; 3 children used the traditional handheld mouse (12%); only one child was familiar with the touchpad. In the waiting-list condition, 20 children (87%) were not familiar with any type of mouse and 2 children were able to use both types (8.7%).

The treatment and waiting-list groups were equivalent for age, $t\ (45) = -0.14$, $p = .89$, SES, $t\ (45) = .64$, $p = .52$, and familiarity with technology, $t\ (45) = -1.24$, $p = .22$. A chi-square analysis confirmed that the treatment group and the waiting-list group were also homogeneous for gender distribution, $\chi 2 = .75$, $p = .39$. Table 2.1 shows the demographic characteristics of the sample.

Table 2.1

Demographic characteristics of the sample. Means, standard deviations and t-test (t) of age, socio-economic status (SES) and familiarity with technology (Fam Tech).

Variable	Group		
	Waiting list (N = 22) M (SD)	Experimental (N = 25) M (SD)	t (DF)
Age (months)	64.82 (3.84)	64.68 (2.72)	.14 (45)
SES	7.23 (.75)	7.04 (1.17)	.64 (45)
Fam Tech	1.17 (.89)	1.48 (.82)	-1.24 (46)

* $p < .05$, ** $p \leq .01$, *** $p \leq .001$

2.2.3 Procedure and Materials

Instructional design. A combined -unplugged plus educational robotics- intervention was proposed to children to introduce them to coding. The training duration was seven weeks, 2 session each week (60 min), for a total of 14 sessions.

The intervention schedule and organization were discussed with the schoolteachers, who also collaborated in the implementation. To ensure that all children received the same activities and intervention dosages, we first recorded attendance on a logbook that was useful to identify absent children. Then, we involved them in extra-sessions of the coding laboratory to allow them to participate in the missed activities.

In order to maximize children's engagement, we developed a scenario and narrative setting for the training sessions: the story of a robot, named *Cubetto* who was navigating through space, attempting to reach Earth and, more specifically, to the school where the training took place. *Cubetto* was being aided by a group of bees, i.e., the *Bee-Bot* robots the children would play with during the training. Thus, each training session started with a message from the bees, in the form of a letter found in the school's letterbox: the message explained the session's activities and tied them to the overarching story. From a pedagogical point of view, we built the training sessions around three phases:

- Preparatory: activities aimed at ensuring that all children shared a common set of basic skills such as being able to distinguish left and right, basic pattern recognition, etc.
- Unplugged coding: activities that introduced the concept of code as a precise sequence of instructions executable by mechanical agents that, at this step, were embodied by the children themselves.
- Educational robotics: activities that introduced the *Bee-Bot* and *Cubetto* robots and consisted of activities during which children could program an actual fully-mechanical computing agent.

The following Sections describe the activities of each session in detail. For a summary of the whole training, see Table 2.2.

Table 2.2

Lessons plan

Coding sessions	Macro -step	Activities
Session 1	Preparatory	Games aimed at developing basic directional skills such as right-left discrimination.
Sessions 2-3	Preparatory	Reproducing "color-codes", i.e., sequences of colors, with construction bricks or colored dots on paper grids. Reproducing sequences of pictorial symbols.
Sessions 4-5	Unplugged coding	Understanding sequences of coded instructions and executing them to create pixel art.
Sessions 6-7	Unplugged coding	Solving navigational tasks on a child-sized map by taking the roles of programmer, robot, and tracer of execution.

Session 8	Unplugged coding	Solving navigational tasks on two-dimensional maps by programming sequences of instructions and executing them by moving a pawn.
Session 9	Educational robotics	Familiarization with the *BeeBot* robots.
Session 10	Educational robotics	Reading, understanding and mentally simulating pre-written sequences of instructions for the *BeeBot*.
Sessions 11-12	Educational robotics	Programming the *BeeBot* to solve navigational tasks.
Sessions 13-14	Educational robotics	Programming *Cubetto* to solve navigational tasks while inventing stories to justify its travels.
Closing session	Educational robotics	Metacognitive reflection on the goals of CT and the meaning of programming.

Session 1

- Goal: ensuring that all children were able to distinguish left-right directions.
- Narrative: an audio message from the bee characters directed the children to the school's letterbox, where they found the materials for the session.
- Activities: the first activity consisted of a game based on right-left discrimination exercises (e.g., instructors showed left hand, right foot, and the children were asked to move the corresponding body part and name the direction). In the second activity the children were presented with sheets of papers representing two windows and were asked to open the left/right window to find a hidden figure underneath. The third activity was a game in which the instructors asked the children, one at a time, to pick a fruit randomly placed on a

table and place it in a basket set either to the left or the right of the table. At the end of this session the instructors gave all children a bracelet to put on their left arm in order to reinforce and aid the recognition of directions during the next sessions.

Session 2 - 3

- Goal: pattern recognition and reproduction of a sequence of instructions.
- Narrative: another vocal message from the bees introduced the concept of code as a sequence that must be executed, in this case reproduced, exactly, step-by-step. The message also introduced the idea of debugging by stressing that errors in coding should not be discouraging and that they may and should be corrected.
- Activities: during the first activity, the children received sheets of paper representing sequences of colors and were asked to reproduce them with toy construction bricks. The bricks were placed in baskets positioned at a certain distance from the table the children were working on, so they had to carefully observe the paper sequence and remember the number of bricks they needed and their colors. Children who committed a mistake in retrieving the bricks, would recognize it while building the sequence and could then go back to the basket to fix it. The sequences of colors varied in difficulty starting with just two alternated colors and ending with sequences of all different colors. See Figure 2.2 (a) for an example of this activity.

Similarly, during the second activity, the children were asked to reproduce a “color-code” consisting of a sequence of dots, on a 4x3 grid drawn on a sheet of paper, with the possibility of having empty tiles (Figure 2.2 b). During the third activity the children reproduced on a 3x3 grid the sequence of colors shown on one face of a randomly shuffled Rubik’s cube. Finally, the fourth activity introduced codes composed of graphic symbols: the children were asked to memorize a sequence of four symbols - e.g., moon, star, square, circle - and to retrieve them from a different table, without referencing the target sequence. See Figure 2.3 for an example.

Figure 2.2. (a) A paper color sequence reproduced by a child with construction bricks.

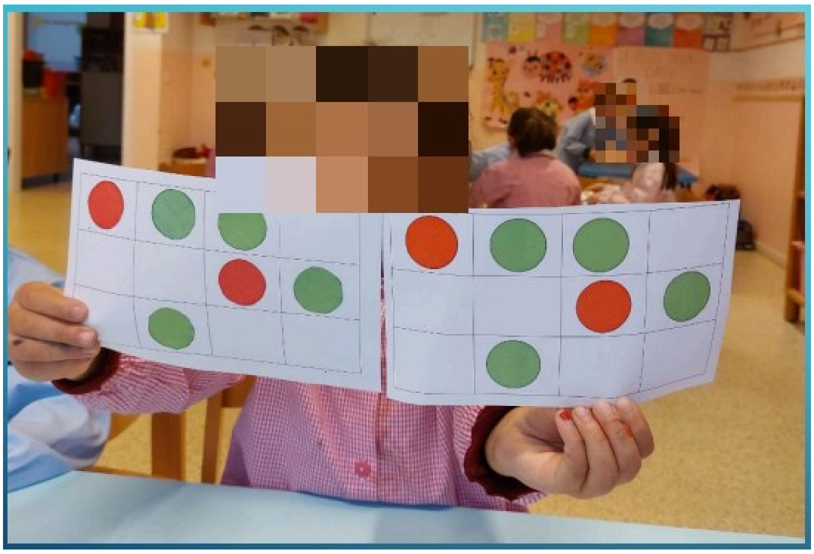


Figure 2.2. (b) Reproducing a color-code on a grid.

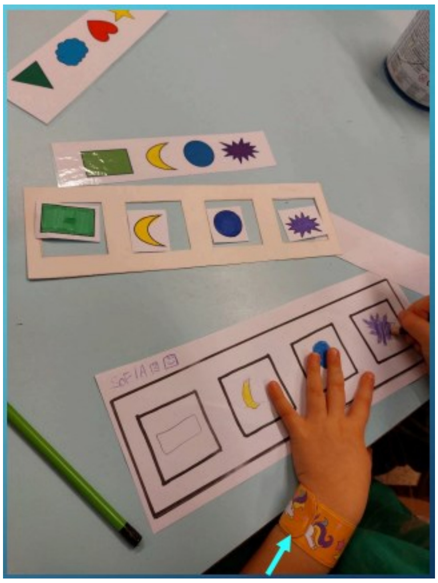


Figure 2.3. Reproducing a sequence of pictorial symbols.

Session 4 – 5

- Goal: understanding sequences of instructions given in a codified form and applying them to create pixel art.

- Narrative: a series of coded instructions was given by the bees to obtain a numerical code which would be used subsequently to access *Cubetto*'s spaceship.
- Activities: during the first activity, the children received (1) large sheets of paper representing a 12x12 grid; (2) colored paper tiles to set on the grids; (3) instructions on how to place the tiles on the grid, given in code form. Figure 2.4 shows an example of how these instructions were given: each row corresponds to a row of the 12x12 grid, the numbers indicate how many tiles of a given color should be placed on the row, the children should "execute" the instructions from top to bottom and from left to right. An important detail is that the instructors did not simply explain how to read the coded instructions. Instead, they encouraged the children to reach the conclusion by themselves and helped them by asking questions and giving clues e.g., pointing out the fact that the coded instructions had the same number of rows as the paper grids. After having understood how to execute the coded instructions, the children started composing the pixel art on the paper grid. Figure 2.5 shows an example of this activity. The second activity of this session was very similar to the first one and served to reinforce the notions to be learned. Figure 2.6 shows the codified instructions and final result of this activity.

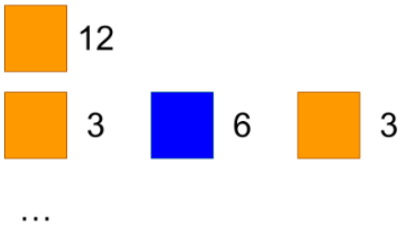


Figure 2.4. Example of coded instructions used to create pixel art.

Figure 2.5. Children executing coded instructions to create pixel art.

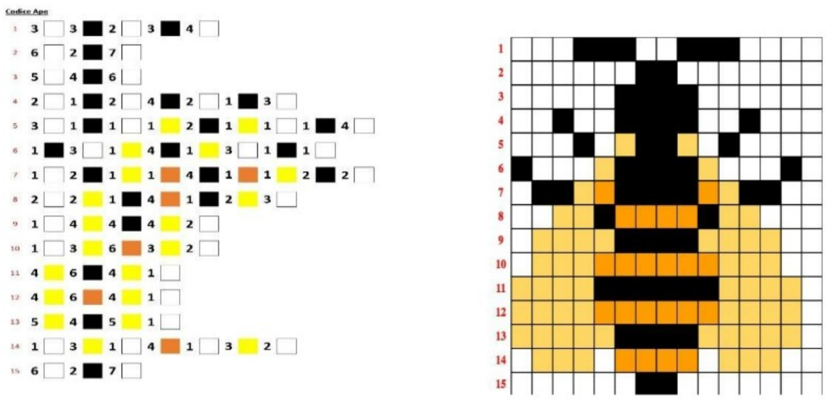


Figure 2.6. Example of a more complex pixel art activity.

Session 6 – 7

- Goal: introduction of the navigational instructions - used to program *BeeBot* and *Cubetto* robots - and direct experience of the roles of programmer, robot, execution tracer.
- Narrative: the bees tell the children that they will learn to use code to travel around the solar system; their goal will be reaching the sun without hitting any planet.
- Activities: the activities of these sessions took place on a large grid drawn on the floor by using tape. Each tile of the grid was large enough for a child to stand in it comfortably. The instructors used symbols and drawings to mark some of the tiles of the grid: a starting position, a goal position (the sun) and some obstacle tiles (planets). Moreover, the instructors explained to the children the instructions used to interact with this navigational task. These instructions were represented as arrows drawn on sheets of paper and had a direct correspondence to those used to program *BeeBot* and *Cubetto*: “move forward/backward” one tile according to the direction the robot is facing and “turn left/right” ninety degrees, without changing tile. During the activity, the children took turns in impersonating different roles:
    - Programmer: tasked with composing a sequence of instructions to guide the robot from the start position to the goal position, without hitting obstacles (planets).

- Robot: tasked with physically executing the instructions given by the programmer, exactly as told, even if this meant “hitting” an obstacle, i.e., walking on a planet tile.
- Tracer: tasked with marking any tile the robot walked on by placing colored breadcrumbs. These breadcrumbs were used to debug programmers’ sequences of instructions. For example, in case of an error, the children could associate each instruction with a breadcrumb and find out which specific instruction caused the robot to hit an obstacle.

Every child was asked to observe the whole process and help capture mistakes. The activities started with very simple tasks, e.g., straight, short paths from the starting position to the goal, and became increasingly more complex by adding more turns and obstacles. See Figure 2.7 for an example of one of these activities.

Figure 2.7. Children attempting to solve a navigational task taking the roles of programmer, robot, and tracer.

Session 8

- Goal: solving navigational tasks by creating sequences of instructions and moving a pawn to execute them.
- Activities: this session moved the navigational games to a smaller scale and tasked the children with solving navigational puzzles represented on sheets of paper by programming a sequence of instructions and then moving a pawn to execute them.

Figure 2.8 shows an example of this activity.

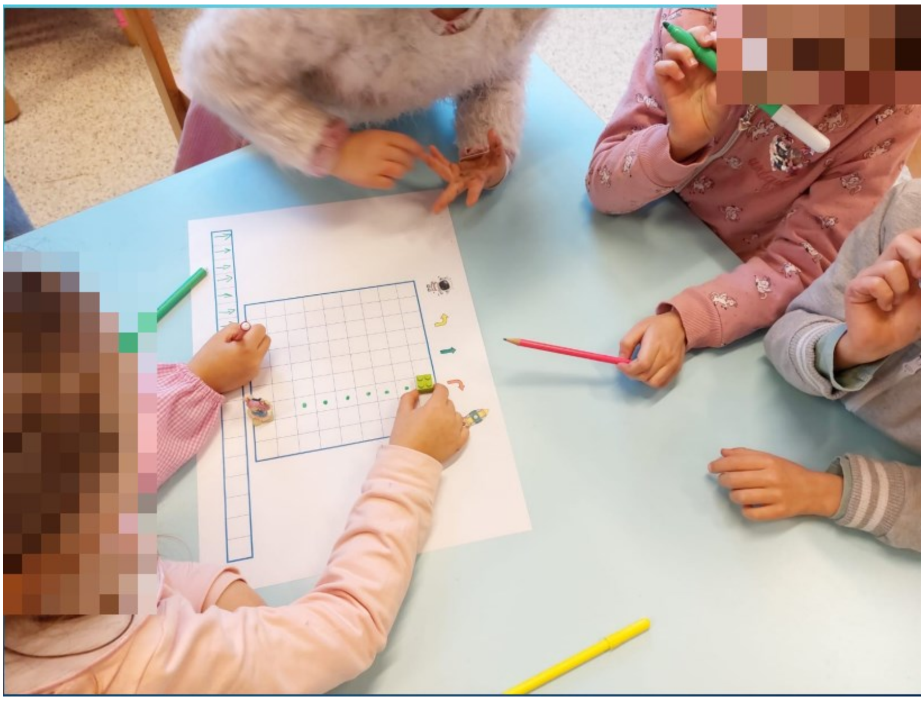


Figure 2.8. *Navigational puzzles solved by programming and executing sequences of instructions by moving a pawn on a 2-d grid.*

Session 9

- Goal: introduction of the *BeeBot* robots.
- Narrative: this session marked the arrival of the spaceship (Figure 2.9); the children used the code obtained at the end of Session 4 to open the spaceship, where they finally found the *BeeBot* robots.
- Activities: this session took a more free-form approach and focused on letting the children interact with the *BeeBot* robots and get a basic grasp of their functioning.

Figure 2.9. *The arrival of the spaceship with the padlock that the children opened with the number code obtained during the pixel art session.*

Session 10

- Goal: understanding pre-written sequences of instructions and mentally simulating their execution.
- Activities: during this session the children were tasked with reading different sequences of code and then anticipating where they would lead the *BeeBot*, given a starting position and direction on a 6x4 tile map. The instructors helped the children by asking relevant questions and pointing out important details such as the starting direction of the robot, the number of instructions in the sequence, etc. For each sequence of instructions, the children placed a marker on the tile they believed the robot would end up in after the execution of the instructions. Then, taking turns, they input the instructions to the robot, watched the execution and verified if they had given the correct answer. The sequences of instructions started simple and short and grew in complexity and length: 2 steps forward, 2 steps with a turn in between, etc. See Figure 2.10.

Figure 2.10. *Children attempting to understand pre-written sequences of instructions and mentally simulate their execution.*

Session 11-12

- Goal: programming *BeeBot* to solve navigational puzzles.
- Activities: during these sessions, the children took on different roles while solving navigational tasks that required programming sequences of instructions with the goal of moving *BeeBot* from a starting position to a goal position on a tile-based map.

The instructors presented each task by placing a *BeeBot* on a starting tile and a marker on a goal tile (Figure 2.11). The children took turns in:

1. writing a sequence of instructions that would take the robot to the goal tile, given its initial position and direction;
2. inputting this sequence of instructions to the *BeeBot*;
3. verifying each other's code e.g., by pointing out possible mistakes.

The tasks were presented in order of increasing difficulty, determined by the length and number of steps of the solutions and the presence of obstacles on the maps.

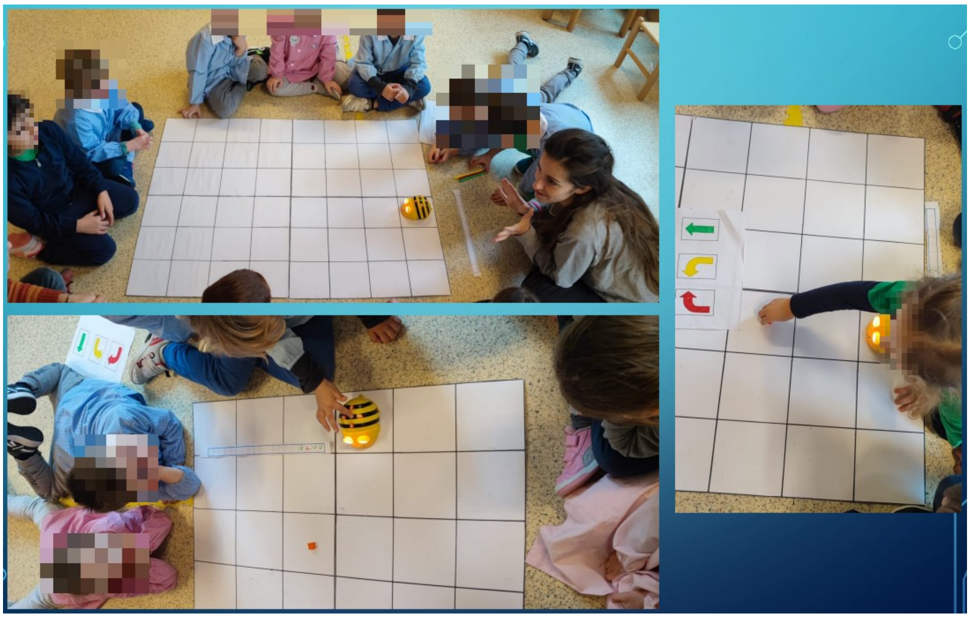


Figure 2.11. Children attempting to program *Beebot*.

Session 13-14

- Goal: programming *Cubetto* to navigate through a tile-based map representing different situations, and inventing stories to justify its roaming.
- Narrative: these sessions were the culmination of the overarching story of the training activities; the robot *Cubetto* finally reached Earth and the children's school; the children would now accompany it in its travels. See Figure 2.12.
- Activities: during these final sessions the children took turns in solving simple navigational tasks by programming the robot *Cubetto*. To engage all children, even while waiting their turn to interact with the robot, the instructors encouraged them to invent stories to justify the roaming of *Cubetto* on a tile-based map in which each tile

represented a particular terrain or destination e.g., the sea, mountains, desert, etc.

Figure 2.12. Activity with the robot *Cubetto*.

Pretest and posttest assessment

At the pretest and posttest, children performed individually 4 coding problems from CoThi platform (see section below). Moreover, two standardized neurocognitive tests were used (one to assess children's response inhibition, the other to assess planning), and a visuo-spatial task. The use of tasks for executive functions and visuo-spatial skills aimed at ascertaining far-transfer effects of the intervention on children's cognitive abilities. In the following sections each test is described in detail.

The assessment of coding and cognitive skills was done individually for each child, and each participant was supervised by the first author of this study and by trainees and master's students in developmental psychology.

All tasks were administered in a classroom setting, as the school had provided a dedicated space for this research project in which children could participate in the assessment phases without being distracted by noises.

Coding skills. Before the pretest, both the experimental and the waiting-list group familiarized with the CoThi platform and the use of tablets with touchscreen by performing an example trial, assisted by the first author of this paper and trained collaborators (trainee students). The pretest started after this familiarization phase. The coding test consisted of four exercises in which the children were asked to program a sequence of instructions to solve two-dimensional navigational puzzles: guiding a bee sprite through a forest to reach a goal position represented by a star.

The puzzles consisted of tile-based maps composed of path tiles traversable by the sprite and wall tiles creating obstacles. Figure 2.13 (a) shows an example of such a map: the white tiles represent paths, the green tiles covered in trees and bushes represent walls. The instructors showed and explained each type of tile to the children before starting the test. To solve the navigational puzzles the children had to create a sequence of instructions by dragging-and-dropping code blocks that implement the exact same commands of the Bee-bot: "move forward/backward" one tile towards the direction the bee is facing, and "turn left/right" ninety degrees on the spot, without changing tile. For example, the puzzle of Figure 2.13 (a) would be solved by the sequence: "move forward, move forward, turn left, move forward, move forward" shown in Figure 2.13 (b). After composing their sequence of instructions, the children would click on a button labelled "Execute" to see whether their coded program solved the problem. The platform executed the sequence of instructions of the children without showing its effects on the map: in case of errors, it would display visual feedback on the screen and the instructors would explain to the child that the program needed some fixing. We decided to avoid showing the execution on the map to (1) reproduce pen and paper unplugged coding digitally and (2) avoid trial and error approaches. The children had up to three attempts to solve each exercise.

We designed the four exercises in a scale of increasing difficulty based on the length of the path and the number and type of turns it contained four exercises of increasing difficulty:

(1) a straight three-tile-long path from the bee to the goal.

(2) a four-tile-long path with a single turn.

(3) a six-tile-long c-shaped path with two turns in the same direction.

(4) a six-tile-long path with three turns in different directions.

This test environment is akin to the unplugged coding and ER activities conducted in the latter training sessions in the sense that it consists of navigational puzzles that must be solved with directional instructions. However, in these tests, the children no longer find themselves physically immersed in the navigational task: they cannot move around the map, change perspective etc. but must decode and understand the 2-d representation they are given and solve it no longer by pressing physical buttons or placing cardboard arrows on a sequence, but *indirectly*, by dragging-and-dropping digital code blocks.

Moreover, not being able to see the actual execution of the program but only receiving feedback stating that there is an error somewhere in the code creates a significantly superior challenge for the children. In particular, in order to debug their programs, they need to mentally simulate the instructions to find out whether they miscalculated a distance or made a wrong turn.

The CoThi platform recorded the children's coding performance which was scored for:

- Coding planning time, in seconds: the time from the moment the child received the task instructions to the moment s/he drags-and-drops the first digital code block.

- Coding accuracy: a score of 3 was given if the child successfully solved the item at first attempt; 2 on solving it at the second attempt; 1 on solving it at the third attempt; 0 otherwise.

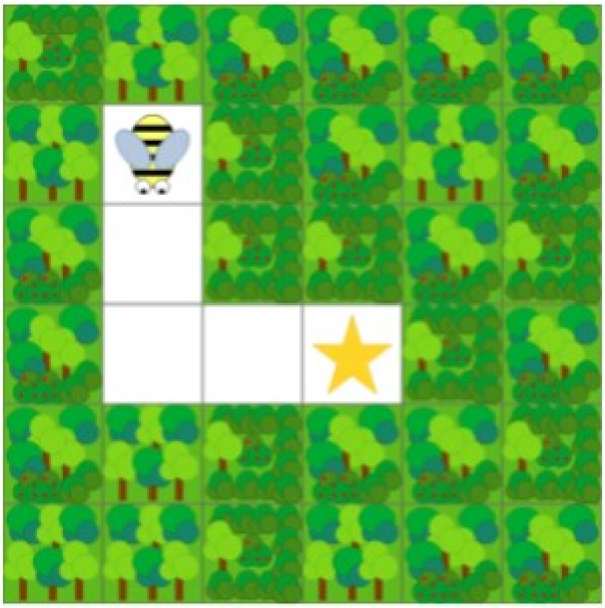


Figure 2.13. (**a**) *2-d tile-based map for a coding test exercise, the goal is guiding the bee to the start, the white tiles are traversable, the tiles containing trees and bushes are obstacles.*

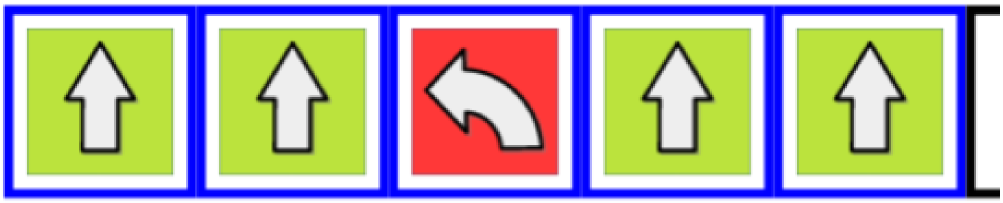


Figure 2.13. (**b**) *sequence of instructions that solves the navigational puzzle of Figure 13 (b).*

Executive functions: Response inhibition. The *NEPSY inhibition (squares/circles) subtest* of *NEPSY-II* (Korkman et al., 2007) was used to assess response inhibition skills at the three assessment times (i.e., pretest, posttest, and delayed posttest). See Figure 1. Test-retest reliability indices for the age range 5–6 years is r = 0.79 for inhibition time and r = 0.77 for inhibition errors (Brooks et el., 2009).

The *squares/circles inhibition subtest* of *NEPSY-II* consists in naming aloud printed figures (circles and squares) displayed on five rows of eight figures each, saying “circle” for squares and “square” for circles. To respond accurately, children must inhibit their automatic response (i.e., naming “circle” when seeing a circle and vice versa). The task is standardized for children aged 3 to 16. Children’s execution time is recorded, and children’s performance was scored for:

- Inhibition time, in seconds: the total time to complete the task.
- Inhibition errors: the number of errors and self-corrections made by the child in performing the task.

Executive functions: Planning. The *Tower of London, ToL* was used to assess planning (Luciana et al., 2009). We used a version of the *ToL* standardized for children aged 4-13 (Fancello et al., 2013). The *ToL* test is used to measure children’s planning and problem-solving skills. The test manual does not report indices of concurrent validity with other measures of EFs. Arfé and colleagues (2020) report a test-retest reliability indices, i.e., r = 0.57 for accuracy scores and r = 0.71 for planning times. The concurrent validity of the test was calculated by considering the correlation between the performance on the *ToL* and *Elithorn* tests for the age group of 6-year-old children.

In order to take the test, children interact with a structure composed of a base from which three pegs of different heights rise. The test subjects must move three beads of different colors (red, blue, and green) to reproduce a target configuration; the moves must follow a set of rules:

(1) the child must move only one bead at a time; (2) at any one time at most one bead can be placed on the shorter peg, two on the middle peg, three on the longer peg; (3) the operation must be strictly sequential, i.e. a bead removed from a peg must be inserted on another peg before removing any more beads; (4) a maximum number of moves is allowed on each trial; (5) each task must be

solved within 60 seconds counted from the moment the examinee receives the visual stimulus of the target configuration. The whole test includes 12 target configurations of increasing difficulty (the number of allowed moves, starting from 2 and arriving at a maximum of 5). Each task starts with the *ToL* in the same initial configuration.

Children's planning skills were scored for:

- Planning time, in seconds, from when the trial is shown to the child until s/he makes the first move, pulling the first ball off the stick.
- Planning accuracy: one point was awarded if the child performed the trial correctly in 1 minute without breaking any rule; 0 otherwise.

Visuospatial skills: mental rotation. The *Primary Mental Ability, PMA subtest* was used to assess visuo-spatial skills (Thurstone & Thurstone, 1965). We used a version of the *PMA* standardized for children aged 5-17. The *Spatial Relations PMA subtest* used to measure children's visuo-spatial and mental rotation skills was the K-1 level subtest standardized for children aged 5-7. The subtest has good reliability (r = 0.83). The task includes four example trial items and 27 trial items. The items are presented on paper. For each item, a part of a square is presented, and on the right, four similar figures are presented. The child is asked to choose one out of the four figures that would complete the square. Before beginning the test, making sure the child knows what shape a square is essential. Once the child has performed and understood the example trials, the experimenter proposes to continue the activity by performing the next figures. The maximum unfolding time is 6 minutes.

Child's visuo-spatial skills were scored for visuo-spatial accuracy: one point was awarded for each trial correctly solved in 6 minutes; 0 otherwise.

2.3 Results

Seven outliers were identified (with scores in dependent variables > 2.5 SD) and deleted from subsequent analyses resulting in a final sample size of 40 ($n = 22$ for the training group and $n = 18$ for the waiting-list group).

As children's scores were not normally distributed, we used non-parametric tests in the data analysis. Between-group differences at the pre-test (T1) were preliminarily explored by a Mann-Whitney U test, which confirmed that the two groups (experimental and waiting-list) did not differ significantly in any dependent measure: coding accuracy ($U = 167.50$, $z = -1.03$, $p = .30$, $r = .16$), coding planning time ($U = 189$, $z = -.24$, $p = .81$, $r = .04$), response inhibition errors ($U = 127.50$, $z = -1.95$, $p = .05$, $r = .31$), response inhibition time ($U = 157$, $z = -1.11$, $p = .26$, $r = .18$), planning accuracy ($U = 179.50$, $z = -.54$, $p = .59$, $r = .09$), planning time ($U = 196$, $z = -.34$, $p = .73$, $r = .05$), visuo-spatial accuracy ($U = 174$, $z = -.92$, $p = .36$, $r = .14$).

A Wilcoxon Signed Rank test was used to test the differences within each group between the pretest (T1) and the posttest (T2), when time the experimental group was exposed to the coding training; and between the posttest (T2) and the delayed posttest (T3), when the waiting-list group received the same coding training. In the following, we report the results of the Wilcoxon Signed Rank tests for each dependent measure. Effect sizes (r) estimation was computed in R (R Core Team, 2021). Statistical significance was adjusted for multiple comparison to 0.007 (Bonferroni's formula, 0.05/7).

Table 2.3 reports the groups' mean and standard deviation at pretest, posttest, and delayed posttest, and the statistical comparison between T1 and T2 and T2 and T3.

Table 2.3

Means and standard deviations (SD) at pretest (T1), posttest (T2), and delayed posttest (T3) for each dependent variable, and statistical comparison between T1 and T2 and T2 and T3.

Variables	Group	Mean (SD)		z value	ES (r)
		T1	T2		
Coding Accuracy	Waiting list	.61 (.92)	.84 (.83)	1.00	0.24
	Experimental	.36 (.73)	3.38 (2.25)	3.84*	0.87
Coding planning time	Waiting list	9.17 (4.47)	6.46 (2.71)	2.33	0.55
	Experimental	12.07 (16.18)	9.04 (5.09)	0.33	0.07
Inhibition errors	Waiting list	1.94 (1.66)	2.68 (2.26)	1.53	0.35
	Experimental	3.64 (3.22)	2.71 (1.98)	1.04	0.23
Inhibition time	Waiting list	52.11 (11.20)	48.07 (10.44)	2.42	0.57
	Experimental	57.34 (15.26)	48.21 (7.10)	2.45	0.54
Planning accuracy	Waiting list	2.17 (1.04)	3.89 (3.99)	1.75	0.32
	Experimental	1.95 (1.05)	3.57 (2.68)	2.54	0.55
Planning time	Waiting list	4.46 (2.24)	4.24 (1.41)	0.81	0.19
	Experimental	4.32 (1.16)	5.06 (2.24)	2.52	0.55
Visuospatial skills	Waiting list	12.21 (3.92)	14.21 (4.02)	1.92	0.43
	Experimental	11.09 (3.96)	15.57 (4.76)	3.09*	0.69
Variables	**Group**	**Mean (SD)**		**z value**	**ES (r)**
		T2	T3		
Coding Accuracy	Waiting list	.84 (.83)	2.58 (2.32)	2.99*	0.71
	Experimental	3.38 (2.25)	2.14 (1.42)	2.58	0.61
Coding planning time	Waiting list	6.46 (2.71)	6.49 (3.38)	0.16	0.04
	Experimental	9.04 (5.09)	5.42 (2.35)	2.42	0.53
Inhibition errors	Waiting list	2.68 (2.26)	2.37 (1.61)	0.67	0.14
	Experimental	2.71 (1.98)	3.18 (3.32)	0.10	0.05
Inhibition time	Waiting list	48.07 (10.44)	45.26 (9.26)	1.21	0.28
	Experimental	48.21 (7.10)	44.13 (10.13)	1.30	0.28
Planning accuracy	Waiting list	3.89 (3.99)	5.58 (3.91)	1.89	0.36
	Experimental	3.57 (2.68)	4.73 (3.43)	1.07	0.40
Planning time	Waiting list	4.24 (1.41)	4.20 (1.29)	1.09	0.25
	Experimental	5.06 (2.24)	4.09 (1.14)	0.50	0.41
Visuospatial skills	Waiting list	14.21 (4.02)	15.95 (3.81)	1.95	0.45
	Experimental	15.57 (4.76)	14.64 (4.26)	1.07	0.24

Note. z value = Wilcoxon signed rank test; ES = effect size; *$p < .007$, statistical significance was adjusted for multiple comparison to 0.007 (Bonferroni's formula, 0.05/7).

2.3.1 Coding skills

Between T1 and T2, a statistically significant improvement in coding accuracy was found for the experimental group, $z = 3.84$, $p = 0.000$, with a large effect size ($r = 0.87$) but not for the waiting list group. Planning time on the coding tasks did not differ significantly between T1 and T2 both for any of the two groups.

Between T2 and T3, a statistically significant improvement in coding accuracy was found for the waiting-list group, $z = 2.99$, $p = 0.003$, after this group received the intervention. The effect size was moderate ($r = 0.71$). Coding accuracy did not instead vary significantly between T2 and T3 for the experimental group, which maintained the posttest coding performance. Planning time in coding did not differ significantly between the two times in any of the two groups.

2.3.2 Response inhibition

No statistically significant differences were found between T1 and T2, or between T2 and T3, in response inhibition errors and inhibition time in both groups.

2.3.3 Planning skills

In the experimental group, differences in planning accuracy between T1 and T2, approached statistical significance, $z = 2.54$, $p = 0.01$, after Bonferroni corrections. The effect size was moderate ($r = 0.55$). For the same group, differences in planning time between T1 and T2 were not statistically significant: $z = 2.52$, $p = 0.02$. For the waiting-list group no statistically significant differences emerged in planning.

No statistically significant differences in planning accuracy and planning time were found between T2 and T3, for any of the two groups.

2.3.4 Visuo-spatial skills

Between T1 and T2, the experimental group improved significantly in visuo-spatial accuracy: $z = 3.09$, $p = 0.002$. The effect size was moderate ($r = 0.69$). No statistically significant differences were found in visuo-spatial accuracy for the waiting-list groups.

Between T2 and T3, any of the two groups showed statistically significant differences in visuo-spatial accuracy. The results are represented in boxplots: Figure 2.15, Figure 2.16, and Figure 2.17.

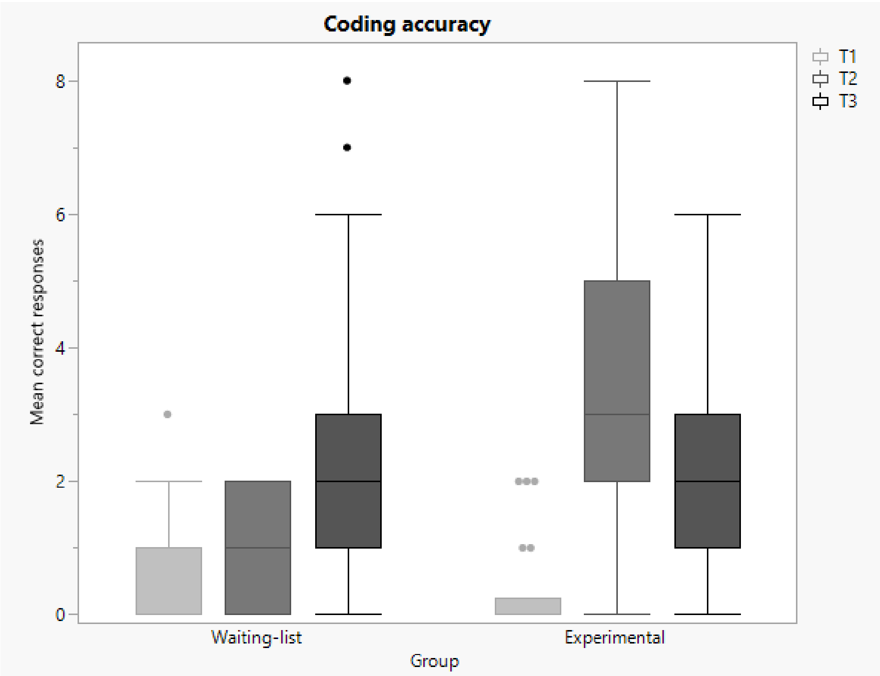


Figure 2.15. *Groups' accuracy on coding tasks at the pretest (T1), posttest (T2), and delayed posttest (T3).*

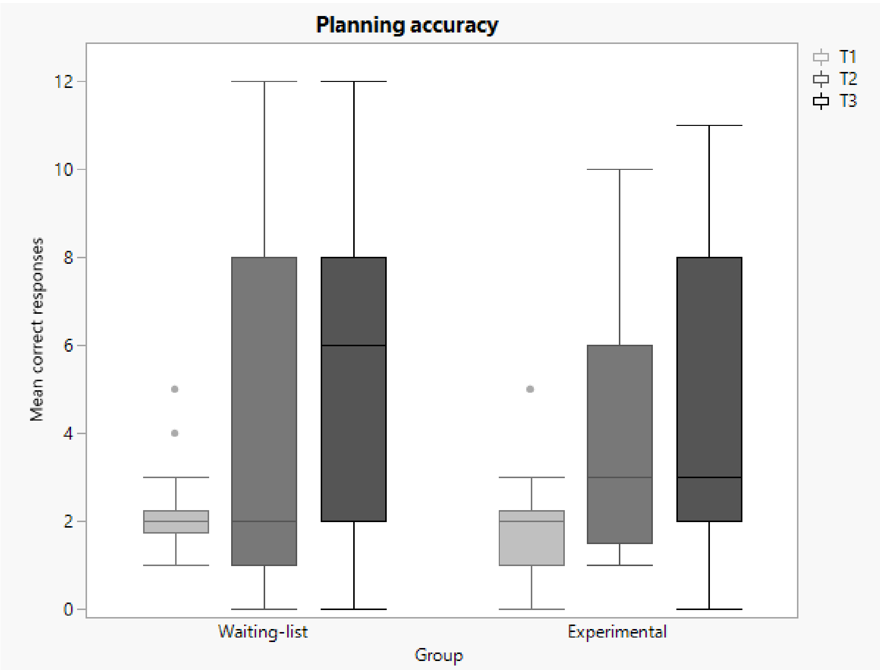


Figure 2.16. *Groups' accuracy on planning task at the pretest (T1), posttest (T2), and delayed posttest (T3).*

Figure 2.17. *Groups' accuracy on visuo-spatial task at the pretest (T1), posttest (T2), and delayed posttest (T3).*

2.4 Discussion

This study investigated the effects of a 7-week combined -unplugged coding and ER- intervention aimed at developing young children's (5-year-olds) computational thinking, response inhibition, planning, and visuo-spatial skills. A stepped wedge cluster randomized trial design was used to test the effects of the intervention, with the experimental and waiting-list control group receiving the intervention at different times (the former between T1 and T2; the latter between T2 and T3). Results suggested that, compared to a waiting-list group, children in the intervention -unplugged coding and ER- group significantly improved their coding abilities and visuo-spatial skills between T1 and T2. With the coding training between T2 and T3, also the waiting-list control group improved in coding, showing at T3 levels of performance equivalent to those of the experimental group. These findings confirm previous evidence of near-transfer effects of ER interventions on coding skills from an early age (Fu et al., 2023). In addition, our findings suggest also far-transfer effects of the intervention, to visuo-spatial skills, and, partly, to planning abilities. Conversely, only response inhibition seemed to be not influenced by the unplugged-ER training, confirming previous evidence from a meta-analysis that highlights the resistance of response inhibition to far-transfer effects (Kassai et al., 2019).

2.4.1. Training effects on CT skills

Our first research objective was to determine the extent to which preschoolers' computational thinking skills could be improved by an instructional program combining unplugged coding sessions and ER activities. Compared to the waiting-list group, children in the experimental group demonstrated improvements in coding skills, with a large effect size ($r = 0.87$), and at T3, after the training, also the waiting-list control group improved significantly in coding, with a moderate effect size ($r = 0.71$). It is important to remark that coding abilities were assessed by plugged coding tasks, and thus children's gains in coding reflected children's ability to transfer the coding skills trained during tangible (unplugged and ER) intervention in which agents were physical objects or other children in a physical environment to a dimension where coding problems were plugged, and thus more abstract, and did not allow to concretely experience in a physical setting the program outcomes. Notably, children were never trained for the virtual coding test, as teachers were unaware of the specific requirements of our computational thinking assessment. For these reasons, we interpret our findings as evidence that the intervention effectively favored generalization and transfer of computational thinking skills. These results confirm those of other recent studies on tangible interventions in preschool (Fu et al. 2023; Roussou & Rangoussi, 2020; Zhan et al., 2022), which showed a general effectiveness of ER training (Fu et al. 2023) and unplugged coding (Zhan et al., 2022) in developing early children's coding skills. Fu and colleagues (2023) found for instance a positive near-transfer effect of -12 hours ER activities on the algorithmic skills of 42 preschoolers at the age 5-6 years with a slightly smaller effect size (d = 0.77) than we found. The consistency of the findings suggests that these types of tangible coding may be particularly suitable for preschoolers to learn computational thinking processes. In the present study we introduced children to coding through tangible activities. Others have recommended to start with concrete representations involving unplugged (Lee & Junoh, 2019), hands-on practices that allow children to physically move things around instead of having to mentally simulate every state of the execution of a program. At this age level, learning takes place essentially based on sensorimotor experience. Starting from Marinus et al.'s (2018) and Critten et al.'s (2022) studies on the feasibility of teaching computational thinking to preschoolers in a tangible environment, we structured the activities aimed at fostering generalization. This was done by introducing the fundamental computational thinking concepts first (e.g., sequences and algorithms) through unplugged coding activities, which were those closer to the child's everyday experience, and gradually shifting to ER. We conjectured that this would allow children to consolidate the new computational thinking concepts gradually, before experiencing programmable robots. Robot programming was introduced as an extension and generalization of the unplugged coding experience. We also reasoned about the additional issue in using robotics with young children, since programming a robot means also taking the robot's perspective to understand which instruction to give or which program steps are needed. Previous research (Critten et al., 2022; Marinus et al., 2018) highlighted that perspective taking can be cognitively difficult for younger children, and this may limit the child's experience of ER, being particularly difficult for preschoolers understanding directions from the robot's point of view. As it will be discussed below, our combined -unplugged and ER- training seemed however effective in training children's perspective taking, as demonstrated by the posttest performance of the groups, which after the training were able to solve plugged coding tasks that greatly relied on perspective taking and performed better also on visual rotation tasks. Improvements in CT skills were not specific to a set-task of highly familiar stimuli, as computational thinking skills were tested on untrained coding tasks.

2.4.2. Cognitive abilities

Our second research goal was to investigate the extent to which a combined -unplugged and ER- intervention would improve children's visuo-spatial skills and EFs. The results revealed that the children exposed to the coding intervention between T1 and T2 improved significantly in visuo-spatial task with a moderate-large effect size ($r = 0.69$). Conversely, the waiting list-group did not equally improve in their visuo-spatial abilities at T3. It is possible that the limited effect of the training for this second group could be related to the period in which the last posttest assessment was carried out. The intervention indeed ended at the completion of the school year when children were very tired and fatigued. We conjecture that this condition might have influenced the children's posttest performance (T3). The experimental group, in contrast, received the coding intervention between January and March and carried out the posttest (T2) in April. The positive effect of coding on these children's visuo-spatial skills has been observed earlier (Brainin et al., 2022; Di Lieto et al., 2020) and our findings provide further confirmatory evidence in this direction. Importantly, in this exploratory study we assessed complex visuo-spatial skills, such as mental rotation and spatial relations. The previous study of Brainin et al. (2022) assessed these complex skills too and found that the improvements in complex visuo-spatial abilities observed for the active control group (trained with unplugged coding) were inferior to those showed by the experimental group trained with ER. Our results provide the first empirical evidence that learning coding in a combined -unplugged and ER- environment positively affects also these complex visuo-spatial abilities.

Although the focus of our coding intervention was primarily aimed at developing computational thinking skills, we also explored transfer effects to two EFs (response inhibition and planning) that had resulted sensitive to coding interventions in prior studies: planning and response inhibition (Arfè et al., 2019, 2020, Di Lieto et al., 2020). Preschool years is a critical period for the development of brain regions that subserve EFs abilities and, as such, might constitute a crucial developmental window to target the malleability of EFs (Brown & Jernigan, 2012). Hence, interventions aimed at enhancing EFs can be especially effective within this timespan. Previous research found a positive far-transfer effect of computational thinking on response inhibition and working memory in first graders who had followed 10-week ER laboratories (Di Lieto et al., 2020). Response inhibition was positively affected by ER, with a moderate-large effect size (d = 0.69). Conversely, in our study we did not find significant effect on response inhibition skill. Other research suggests contrasting findings about the effectiveness of coding intervention on response inhibition skills (Yang et al., 2022; Canbeldek & Isikoglu, 2023). The trial by Yang and colleagues (2022) for example tested the effectiveness of ER intervention on 101 kindergartens' performance in an inhibitory control task-HTKS, finding no significant effect. Conversely, Canbeldek and Isikoglu (2023) showed the effects of combined -ER, unplugged, and virtual coding- training on improving eighty preschoolers' self-regulation skills. Our results confirm Yang and colleagues' findings and are consistent with the results of Kassai and colleagues (2019) recent meta-analysis about the near and far-transfer effects on children's EFs skills. In fact, also this meta-analysis reports insignificant far-transfer effects of EFs intervention on inhibitory control, working memory, or flexibility. Performance on the EFs components that were trained did significantly improve, however, these gains did not transfer to the untrained components (Kassai et al., 2019). This suggest that in order to have a far-transfer effect on EFs at this early age, requires interventions targeting exactly the specific component of EFs skills to be trained. This is also in line with the results of Di Lieto et al.'s trial. The ER intervention by Di Lieto and colleagues (2020) aimed at enhancing EFs, 16-session were planned and structured so as to improve specific EFs. They found significant effects on visuo-spatial working memory. Another difference between this and our study was that the overall duration of the training was longer than in our trial: 20 ER sessions of 60 minutes each. It may be, our combined -unplugged coding and ER-intervention, which was primarily aimed at teaching basic computational thinking concepts, was less apt at scaffolding EFs in these young children. Past studies demonstrated that learning coding can be effective both for the development of coding skills and EFs in six-year-old children (Arfé et al., 2019, 2020). At this phase of cognitive development one year difference is significant and can explain differences in outcomes. Moreover, as anticipated the duration of our training could have been insufficient to consolidate children's cognitive gains. In our 14-session training, we devoted 3 sessions to activities aimed at developing directional skills (differentiating left from right) and reproducing sequences of colors or pictorial symbols; 5 sessions to unplugged coding, which were not directly aimed at training EFs. One session focused on the familiarization with the *BeeBot* robots, and 5 sessions were focused on ER. Following prior research (Critten et al., 2022; Metin, 2022) in which preschoolers were first exposed to unplugged coding and then to *BeeBot* and *Cubetto* robots, we hypothesized that generalization would be facilitated by shifting from unplugged to ER activities. It may instead be that shifting between tools and types of activity in this short time was insufficient for consolidating EFs, and particularly inhibition skills. The lack of significant effects of the training on children's EF might be also related to the insufficient power of the study. Although significant effects were found for coding and visuo-spatial skills, the effects only approached significance for planning. It could be that inter-individual variability in this high order EF was still large at this age, as literature suggests (Anderson & Reidy, 2012).

2.5 Limitations and future directions

The current study has some limitations which should be acknowledged.

Firstly, the main limitation is that this study cannot determine which tool (unplugged coding or ER) is most effective for fostering the development of computational thinking and enhancing of cognitive abilities such as response inhibition, planning, and visuo-spatial skills. The present study tested the effectiveness of a combined - unplugged coding and ER- intervention and not the intervention modalities independently. Although the findings of this exploratory study suggest the efficacy of these activities combined, such a conclusion should be taken with some caution. The current study only provides partial evidence that the specific combination of two coding tools can promote preschoolers' coding and cognitive abilities. We do not determine whether an intervention designed on one of these tools alone would have been more effective than the other.

As in previous studies (e.g., Arfé et al., 2019; Unterrainer et al., 2013), we assessed response inhibition and planning by standard EFs tasks. We have used one task for each cognitive ability we considered. We could have used more than one task for each executive function to increasing the reliability of EFs assessment. However, significant issue occurs when evaluating EFs, especially when young children are involved. The final phase of the present exploratory study took place at the end of the school year, a moment in which children might have been tired and cognitively fatigued. Performance during the follow-up assessments may therefore have been affected by factors related to the time of the tasks administration.

Lastly, it is important to mention that our sample may not entirely represent the Italian population of preschoolers. Indeed, the sample size was not large, and the children involved in our research project were from middle-high socio-economic status.

These considerations suggest the need for further studies to test, on one hand, the effects of intervention duration and dosage, and on the other, to compare different instructional programs or the differential contribution of intervention components like unplugged coding and ER and their combination. Indeed, due to the lack of intervention studies, little is known about which coding tools or programs (e.g., unplugged coding or ER) can be most effective in developing coding ability and boosting children's cognitive skills. Future research could fill these gaps, conducting a randomized controlled trial to test the effectiveness of different coding tool, such as (1) combined unplugged coding and ER; (2) unplugged coding; (3) ER; with a waiting-list control condition.

Another future direction would be to verify the effectiveness of tangible coding on the development of coding skills and cognitive functions in children from disadvantaged socio-economic level.

2.6 Conclusions

Although the interest in coding has increased rapidly in the last years (Bakala et al., 2021; Tsarava et al., 2022), research focused on the cognitive effects of these activities is limited. Although recent studies suggest a positive effect of coding on the development of first graders' EFs, no studies yet have examined these effects in preschoolers. The present exploratory study is a first attempt to fill this gap.

The randomized controlled trial reported in this paper is the first finding about the effectiveness of a combined -unplugged and ER- training on the development of preschoolers' coding and cognitive abilities. The study shows that learning the computational thinking concepts by tangible coding during the last year of preschool not only significantly improves children's skills to solve coding problems (near-transfer effect), but it may also have some far-transfer effects on cognitive functions, such as visuo-spatial skills. In the current study, the benefits of coding learning have been observed in the period of transition from preschool to the first year of schooling, which has been shown to be a particularly sensitive time window for the cognitive development.

As we highlighted in the introduction of this study, evidence about coding efficacy is needed to orient instructional decisions regarding which tools and programs are best to introduce young children to coding activities in preschool. Although the teaching of CT is compulsory, we still know very little about what programs work best and, consequently, also recommendations for instructional practice are lacking. The present study also contributes to this direction, designing and testing an instructional program for teaching coding from an early age.

CHAPTER 3. ASSESSING THE IMPACT OF AGE ON THE COGNITIVE EFFECTIVNESS OF CODING INTERVENTION [3]

3.1 Introduction

Although current research supports the idea that coding/CT should be introduced in the school curriculum from an early age (e.g., Bers et al., 2019), this conviction is not strictly grounded on cross-sectional or comparative evidence which considers age-related differences in children's ability to learn to code. For instance, only few studies have compared the response to coding instruction in children of different age or grade level (e.g., Atmatzidou & Demetriadis, 2016; Kyza et al., 2022). Since coding and CT involve a complex set of cognitive skills, it is possible that older children who have more mature cognitive skills could benefit and learn more than younger children from coding/CT activities (Robledo-Castro et al., 2023).

3.1.1 Studies Examining Age-Related Differences in Coding and CT

Tsarava et al. (2022) tested the CT skills of 8-10 year-old third and fourth grade elementary school children and found positive and significant correlations, that is the older had higher performance, between performance on CT tasks and children's age and grade level. Their analyses also showed that children's verbal reasoning and visuospatial reasoning contributed to predict children's performance in CT tasks.

Kyza et al. (2022) administered a 6-hour (collaborative) CT/coding intervention with ScratchJr software to two cohorts of students, the first of 6 to 9-year-old and the second of 10 to 12-year-old children and found significant differences between the two cohorts in the complexity and quality of children's projects. The children in the older cohort, age 10-12, reached higher levels of CT than the

---

[3] The paper based on this study is currently in preparation.

younger group. The authors attributed these differences to children's cognitive development, assuming that a certain level of cognitive maturation was necessary to benefit most from the intervention.

Similar findings were obtained by Rjike et al. (2018), who compared the CT skills and perceived task difficulty of 200 primary school children who were introduced to the computational concepts of abstraction and decomposition. The participants, aged between 6 and 12 years, were distributed in three age groups, 6-8-years old, 8-10-years-old, and 10-12-years old. Although the authors did not find significant differences between the younger and older children in their perceived task difficulty, cognitive load or task flow, age-related differences emerged in children's CT performance. Older students performed better in the abstraction task than the 8-10-years old students and the youngest -6-8-years old- children; the 8-10-years old students performed better than the 6-8-years old children. In the decomposition task, the oldest group (10-12 year-olds) performed better than the two younger groups (8-10 years and 6-8 years). Based on these findings, the authors concluded that only from 9-10 years children had the basic skills to fully benefit from the introduction of CT concepts like abstraction and decomposition.

Papavlasopoulou et al. (2017) used eye-tracking data to examine differences between 8-12 and 13-17 year-old students performing coding tasks after an introductory 5-sessions coding workshop based on a constructivist approach. Participants were engaged in interactive and game design activities. The learning gains of the two age groups were scored based on pre- and posttest coding tests. The results showed that the older group learned more from the workshop than the younger. The qualitative analysis of students' performance through eye-tracking methodology showed that the greater learning gains observed in the older group were associated to their eye-movement (coding) behaviors: the older group focused more on scripts, commands, and outcomes than the younger group, showing greater attention toward games execution and monitoring than the younger participants, who focused more on sprites, that is surface aspects of the games.

Opposite are the age-related effects observed in another recent study by Jiang and Wong (2022). The authors tested the response to a 5-week 5-lessons CT intervention of two groups of four grade (ages 9-11) and six grade (ages 12-13) students, finding that although the older group showed a higher performance than the younger group on CT tasks before the intervention, the younger (four grade) students showed the greater response to the intervention, benefitting more from it, except for generalization skills. Jiang and Wong (2022) speculated that the greater improvement of the younger-age group was explained by the developmental phase the children were going through or their readiness to acquire CT concepts like conditionals or pattern recognition.

Finally, Atmatzidou and Demetriadis (2016) conducted 11-week robotic learning activities with older students from junior high (15 years) to vocational high school (18 years), with no prior experience of robotics. Overall, the authors did not observe age-related differences in response to the intervention, except for algorithmic thinking and generalization, which improved more for the older age-group.

Previous studies, thus, suggest that sensitivity to coding or CT intervention varies with children's age, and that this is likely associated to cognitive development factors, such as children's capacity to engage in logical reasoning (Lai & Yang, 2011) or abstract patterns (Novack et al., 2014) which can be considered high-order cognitive skills. Based on the studies reported, from the age of 9 or 10 children seem to benefit most from coding/CT interventions. At this age, they may show some high-order cognitive like planning or reasoning abilities at an emerging level, and this could explain their greater sensitivity to coding interventions.

3.1.2 Age Differences in Executive Functioning

Executive functions (EFs) have been defined as the cognitive abilities necessary to maintain an organized problem-solving approach to achieve a goal. Executive functioning is a complex skill set that includes several cognitive processes (e.g., mental representation of tasks and outcomes, Welsh and Pennington, 1988) and allow to perform independent, purposeful, and adaptive behavior (Lezak, 1983). The core EFs include the inhibition of impulsive responses, the maintenance and updating of information in working memory (WM), and cognitive flexibility (Anderson, 2002; Gioia et al., 2000; Miyake et al., 2000). These core EFs serve as foundational components from which higher-order executive functions, such as planning, problem-solving, and reasoning evolve (Diamond & Ling, 2016; Miyake et al., 2000; Thayer & Lane, 2000). Both core and higher-order executive functions significantly contribute to children's cognitive development, influencing their abilities to succeed in several cognitive tasks and problem-solving scenarios (Diamond & Lee, 2011).

EFs' developmental trajectory depends on the development of the brain, specifically the prefrontal cortex (De Luca & Leventer, 2008), as well as environmental stimulation (Hughes, 2011). The three core EFs (i.e., inhibitory control, working memory, and cognitive flexibility) start to develop in the first year of life, with the first signs of EFs evident by 8-9 months of age (Diamond et al., 2006). A significant first growth spurt is observed between the ages of 2 and 3, with further improvements occurring around age 8 (Vandenbroucke et al., 2017).

Computational thinking, along with its components, is a problem-solving process that involves executive functions. Practicing CT skills engages executive processes. Specifically, during a coding task, children may employ executive functions such as inhibition and planning. Inhibition allows the child to refrain from coding impulsively and instead allows the child to reason about the task. Planning allows the strategic organization of steps necessary for formulating the algorithm to solve the problem. It is thus not surprising that recent research bring evidence in support of a causal link between the teaching of CT and the improvement of EFs skills in children (Arfé et al., 2029, 2020; Di Lieto et al., 2020).

Inhibitory control is the ability to maintain concentration by disregarding distractions and exhibiting resistance to the impulse of producing a specific response, opting instead to generate an alternative response (Diamond, 2006). Research has shown that it develops gradually over time, with improvements during the preschool and early school years (Best & Miller, 2010). Some inhibitory control skills are already present in the first year of life (Diamond, 2006) and undergo an intensive increase in the preschool period (Garon et al., 2008). Research has shown that children inhibitory control improvement occurred gradually from age 3 through age 4 (Carlson, 2005; Zelazo el al., 2003), with an intensive improvement in children aged 6 (Klenberg et al., 2001). Hence, one significantly developmental change in the ability to inhibit the impulsive responses occurs during the first year of primary school (Diamond, 2013). However, the inhibitory control skills continue to develop throughout childhood and adolescence. Jonkman et al. (2003) and Casey et al. (1997) found significant decreases in commission errors on inhibition tasks from age 9. During middle childhood, inhibitory control continues to develop, with evidence suggesting that children aged 12 show improvements in their ability to inhibit irrelevant information and regulate their behavior (Diamond, 2013). It is then during adolescence that the more complex inhibition skills inhibitory undergo significant maturation. Studies found a substantial improvement on inhibition tasks both in reaction time and accuracy measures in adolescent aged from 14 to 16 (Huizinga et al., 2006). Inhibitory control was higher in young adulthood compared to adolescence, and it declined from 35-years-old (Ferguson et al., 2021).

Working Memory involves the ability to retain and manipulate information in the mind over a short period without relying on external aids or cues (Diamond, 2006; Huzinga et al., 2006). The development of working memory proceeds gradually, starting with basic perceptual and sensorimotor functions and culminating with the physiological maturation of widespread neural networks that integrate complex processing demands inherent to working memory tasks (Luciana & Nelson, 2002).

Research has shown that children as young as 9 to 12 months have the ability to update the content of their working memory (Bell & Cuevas, 2016; Diamond, 2006). The ability to keep multiple information in mind and manipulate them in the working memory requires however a more extended period to develop. The performance on WM tasks depends on their complexity and, thus, the executive demands of the task, with fewer demanding tasks being mastered earlier in development. As children develop the specific function of WM, they become able to master increasingly complex tasks (Luciana et al., 2005). A first significant growth spurt is observed in preschool period, at age of 4, with a particular improvement in the ability to update working memory (Garon et al., 2008). During childhood, working memory undergoes further important maturation. Evidence suggests that by age 6, the executive processes of working memory develop, allowing for the use of complex WM tasks that require coordination of WM subcomponents. A significant increase in working memory abilities occurs across the ages of 9 – 17 (Best & Miller, 2010; Conklin et al., 2007).

Children aged 8 to 11 show significant additional improvements in their capacity to maintain and manipulate information over time (Gathercole et al., 2004), with further gains in complex WM tasks (i.e., requiring a greater degree of processing such as the maintenance and manipulation of information without reliance on external aids or cues) through adolescence.

Cognitive flexibility is the third core EF and refers to the ability to switch flexibly from one task or set of mental processes to another (Miyake et al., 2000). In early childhood, cognitive flexibility is relatively limited and gradually emerges between the age 3 and 4. Some research, in which switching was assessed by the Dimensional Change Card Sort (DCCS) task, demonstrated that cognitive flexibility begins to develop later than other core EFs, only around the age of 3 (Diamond, 2013; Smidt et al., 2004; Zelazo et al., 1996). Between the ages of 4 and 7, children increase their cognitive flexibility further, demonstrating better task-switching abilities and mental set shifting (Zelazo et al., 2013). Davidson et al. (2006) found improvement from age 4 through adolescence (e.g., Huizinga et al., 2006). Indeed, cognitive flexibility continues to develop, with evidence suggesting that children aged 8 to 12 have greater ability to flexibly adjust their thinking and problem-solving strategies, to adapt to changing situations, and consider multiple perspectives (Davidson et al., 2006).

Planning is a crucial higher-order EF, which plays a key role in other higher-order cognitive processes such as problem-solving, decision-making, and goal-directed behavior. It refers to the cognitive ability involved in formulating, organizing, and implementing a sequence of actions or strategies to achieve a specific goal. Planning encompasses the ability to anticipate future steps, set priorities, and allocate resources efficiently. Children who show well-developed planning skills can create and follow a systematic plan, consider potential obstacles, and adapt their strategies as needed. The development of core executive EFs clearly precedes and lays the foundation for the maturation of higher-order EFs (Best & Miller, 2010; Diamond, 2013). The hierarchical nature of EFs development indeed explains the dependence of higher-order executive functions on the core abilities.

The emergence of planning skills starts between the ages of 5 and 6 (Usai et al., 2014). From the ages 6 to 9, planning skills undergo substantial development (McGuckian et al., 2023), followed by a second remarkable improvement in planning between the age of 9 and 15 years, that is related to the maturation of prefrontal regions (Luciana et al., 2009). However, the most intensive planning developmental modifications occur during adolescence (Luciana et al., 2009).

3.1.3 The Effects of Coding on children's Executive Functions

Recent research has shown how the benefits of teaching CT and coding can also extend to other cognitive functions. Practicing complex problem-solving abilities, CT can indeed stimulate the development of those cognitive skills that are involved in problem-solving, such as planning (Arfé et al., 2020), reasoning (Scherer et al., 2019), metacognitive skills (Scherer et al., 2019; Robledo-Castro et al., 2023) as well as of more elementary executive functions like response inhibition (Arfé et al. 2019; Arfé et al., 2020; Di Lieto et al., 2020) and working memory (Di Lieto et al., 2020).

Recent research investigating the effectiveness of coding instruction on enhancing executive functions skills in young students (i.e., first graders) suggests a positive impact of coding on such EFs. Two experimental studies by Arfé and colleagues (2019, 2020) implemented a plugged-in coding curriculum for first-grade students. The results revealed significant improvements in response inhibition and planning abilities. These findings are consistent with the study conducted by Di Lieto and colleagues (2020), which tracked first-graders engaged in educational robotics training and found significant improvements in their working memory and inhibition EFs.

A recent RCT study (Robledo-Castro et al., 2023) showed that CT trainings may have a significant effect on 10-11 years old children's EFs. Specifically, these effects were observed in tasks assessing WM skills, which include metamemory abilities, as well as metacognitive monitoring and control related to the anterior prefrontal cortex. Furthermore, the study found improvements in tasks related to visuospatial planning ability, sequential planning, and mental flexibility, which are associated with the dorsolateral cortex. In contrast, significant effects were not found on Stroop inhibition and motor control tasks, which assess EFs related to the orbitofrontal area. The same research group demonstrated in a recent second study (Robledo-Castro et al., 2023) that children aged 10-11, who were trained with CT activities, showed improvements on visuospatial working memory, response inhibition, and sequential planning. The participants received an 8-week CT program consisting of 16 sessions held twice a week, which aimed to enhance cognitive skills through both plugged and unplugged activities. The training activities were both individual and collaborative and consisted of three structured units in each session: Conceptualization, involving explanations through videos, games, or real-life examples; Unplugged group activities, incorporating games and exercises to introduce computational thinking principles without electronic devices; and Plugged-in activities that involved children learning to code using laptops and a block-based language through the MakeCode platform.

As noted earlier, few studies (e.g., Atmatzidou & Demetriadis, 2016; Kyza et al., 2022; Rjike et al., 2018), mostly focused on older students (aged 10 to 18 years), have examined age-related differences in CT abilities in response to coding instruction and none of them has explored whether age differences in learning to code translate also in differences in cognitive gains with coding.

Our study aimed to extend the findings of the previous studies with a focus on the age-related transfer effects in response to a coding intervention. The main goal was to investigate the cognitive benefits of executive functions (EFs) for both first and fourth graders. By targeting distinct age groups, we aimed to provide insights into how a coding intervention may benefit individuals at different developmental stages. This approach allowed us to identify potential variations in the transferability and impact of coding skills on EFs, providing a more comprehensive understanding of the intervention's efficacy across different ages. Our research hypothesis stems from previous studies (Arfè et al. 2019, 2020; Di Lieto et al., 2020), which suggest that first graders may benefit from coding activities, with large improvements in their coding and EFs skills.

Previous research has suggested that older students with higher cognitive abilities may benefit more from CT activities in the acquisition of coding skills. However, recent studies have shown that these advantages can be seen as early as in young children, particularly at the beginning of primary school. This indicates that the greatest benefits may be achieved during a period of heightened cognitive plasticity.

The present study addressed the following research questions:

(1) Do age-related differences manifest in the acquisition of coding abilities between first and fourth graders after a virtual coding intervention?

(2) Do age-related differences manifest in the impact of a virtual coding intervention on executive functions (EFs) skills, specifically inhibition and planning, among first and fourth graders?

3.2 Method

3.2.1 Experimental design

The effects of the exposure of experimental group to CT were tested against the effects of business-as-usual (standard) instructional STEM activities in a cluster-randomized controlled trial (CONSORT guidelines, Campbell et al., 2004), with random allocation of classrooms (clusters) to the treatment (coding intervention) or control (standard STEM) condition. The study involved pre- (T1) and post-testing (T2) of cognitive (response inhibition and planning) and CT skills. Between the two tests, children belonging to the treatment group received a coding intervention; children in the control group continued with business-as-usual STEM activities. The coding intervention consisted in 8-hours of coding labs in which children solved coding games from Code.org under the supervision of expert instructors. Control activities consisted of 8-hours of math and technology in which coding was not practiced. Figure 3.1 provides a synopsis of it.

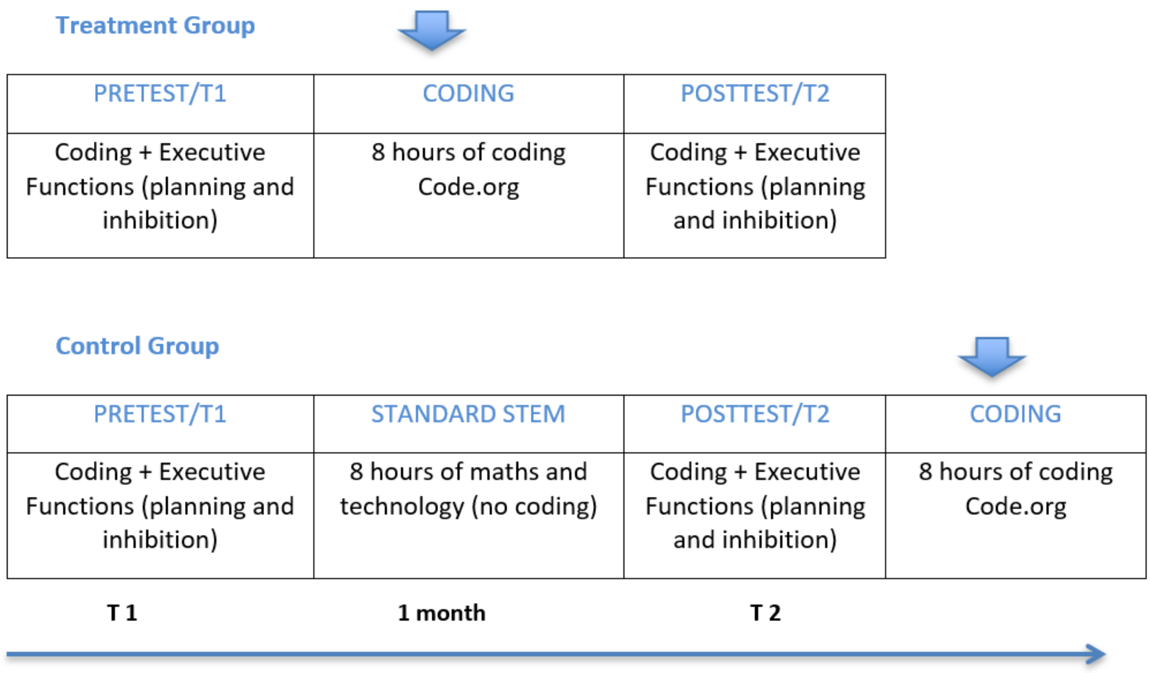


Figure 3.1. *Experiment design.*

3.2.2 Participants

Four-hundred thirty-seven primary school children, attending the first year or the fourth year of primary school, from 5 schools and 25 class groups from different socio-economic status (SES) areas of northern Italy participated in the study. None of the children had previously been exposed to coding. Within each age group (first or fourth grade) children were assigned to a treatment and a control group.

The first-grade group comprised 273 children (134 girls, 49%) in total: 128 children (54 girls, 42%), 8 classrooms, assigned to the treatment condition and participating in coding labs immediately after the pretest (T1) and 145 children (80 girls, 55%), 8 classrooms, the control group, assigned to standard STEM activities (math and technology) and receiving the coding intervention only after the posttest (T2). The mean age of the participants was 6.02 ($SD = 0.41$) in the experimental group, and 5.97 in the control group ($SD = 0.30$). Children's socio-economic status, SES, was assessed by means of a socio-demographic questionnaire that their parents returned along with the written informed consent to participate in the study. SES was estimated based on parents' education, on a scale from 0 (less than elementary school) to 4 (college), and occupation, from 1 (unemployed) to 4 (professional roles). A composite score was calculated as the sum of the highest education score and the highest occupation score obtained by either parent (Arfé et al., 2019), with maximum score 8. The mean SES of the experimental group was 5.53 ($SD = 1.54$); that of the control group was 5.73 ($SD = 1.64$). Familiarity with technology was gauged by asking parents about children's daily use of personal computer, smartphone, or tablet devices in their home environment. A composite score was calculated as the sum of the use of three digital devices. The mean of the experimental group was 1.70 ($SD = 0.86$); that of the control group was 1.61 ($SD = 0.87$).

The first-grade treatment and control groups were equivalent on age, $t(271) = -1.35$, $p = .18$, SES, $t(271) = -1.02$, $p = .31$, and familiarity with technology, $t(267) = -0.84$, $p = .40$.

The fourth-grade group comprised 164 children (71 girls, 43%): 94 children (44 girls, 47%), 5 classrooms, assigned to the treatment, or coding, condition between T1 and T2, and 70 children (27 girls, 39%), 4 classrooms, assigned to the control, or standard STEM condition, receiving the coding intervention after the posttest (T2). Mean age was 8.95 (SD = 0.27) in the treatment group, and 8.93 (SD = 0.26) in the control group. Mean SES was 5.52 (SD = 1.72) for the treatment and 4.93 (SD = 1.67) for the control group. The fourth-grade treatment and control groups were equivalent on age, t (162) = -.44, p = .66. Differences in SES and familiarity with technology were significant, respectively t (159) = -2.20, p = .03, and t (160) = -2.73, p = .01.

First graders and fourth graders did not differ significantly on SES, t (69) = -1.01, p = .31, familiarity with digital devices, t (69) = -1.01, p = .31, or gender distribution, χ^2 = 1.38, p = .24.

Demographic data are reported separately for first graders and fourth graders in Table 3.1.

Table 3.1

Demographic characteristics of the Control and Experimental Group for first grade and fourth grade.

	First grade (n= 273)			**Fourth grade (n= 164)**		
	Control	**Experimental**		**Control**	**Experimental**	
	M (SD)	**M (SD)**	**t (df)**	**M (SD)**	**M (SD)**	**t (df)**
Age	5.97 (.30)	6.02 (.41)	-1.35 (271)	8.93 (.26)	8.95 (.27)	-.44 (162)
SES	5.73 (1.64)	5.53 (1.54)	1.02 (267)	4.93 (1.67)	5.53 (1.72)	-2.20 (159) *
FamTech	1.61 (.87)	1.70 (.86)	-.84 (267)	1.52 (.76)	1.87 (.84)	-2.73 (160) **
Gender						
	n (%)	**n (%)**	χ^2	**n (%)**	**n (%)**	χ^2
Female	80 (55.2%)	54 (42.2%)	4.59*	27 (38.6%)	44 (46.8%)	1.11
Male	65 (44.8%)	74 (57.8%)		43 (61.4%)	50 (53.2%)	

Note. * *p* < .05; ** *p* ≤ .01; *** *p* ≤ .001; SES = Socio-economic status; FamTech = Familiarity with technology.

3.2.3 Procedure and Materials

Instructional design

The Italian version of the Code.org platform provided by the "Programma il futuro" initiative (https://programmailfuturo.it/) was used to introduce children to coding. Code.org is a nonprofit organization with the goal of broadening access to CS activities globally. Its web platform offers a variety of CT courses and activities geared towards users of all grades. In Code.org children are faced with CT games or problems consisting of giving instructions to a sprite (angry bird, a bee, a zombie) who must achieve a predefined goal (e.g., reach a target through a maze). Instructions are given by sequencing visual code blocks, corresponding to commands, whose execution aims at the objective. Figure 3.2 shows an example of game interface in Code.org. The upper left corner of each game's interface contains the exercise map which shows a sprite (a character that the users control with code), a goal destination or a series of intermediate goals (e.g. in Figure 1 users must guide the bee to first retrieve nectar from the flower and then produce honey on the honeycomb) and a series of cells that can be traversable or not (this puts some constraint on the range of allowed movements). The rest of the interface contains (1) an instruction panel, on the top page, which presents a brief text that states the goals of the exercise and provides informative feedbacks in case of need; (2) a toolbox (labeled Blocks in Figure 3.3) which holds the code blocks available to solve the task; (3) a workspace area in which users can compose their programs by moving the visual code blocks from the toolbox area and anchor them to the first block in the workspace area. Running the program children can test its efficacy. A system of non-verbal (e.g., the bee crashing on the hedges) and verbal informative feedback (e.g., helps appearing on the top page) enables children to self-monitor their progresses on the screen and correct programming errors. While children progress through the platform, task difficulty progressively increases too.

Figure 3.2. *Interface of an exercise from Code.org*

In the present study the lesson plan was designed to cause children to switch as much as possible between computing functions (sequencing, debugging, and loops) or scenarios (e.g., Artist, Maze), so that the consequent task novelty forced children to retain a problem-solving approach to the coding tasks at hand.

Children of the two age-groups (1st and 4th graders) received the same amount of instruction, participating in eight coding laboratories of 60 minutes each. Each laboratory involved the solution of 5 to 8 coding problems. The first author of this study led the lab activities with the assistance of a post-graduate student and three master students. Table 3.2 reports the full lessons plan for the first and fourth graders.

The type of code games selected to train first and fourth graders differed. Course 1 of Programma il futuro (https://programmailfuturo.it/come/lezioni-tecnologiche/corso-1) was used as our participants, 1st graders who were enrolled at the beginning of the school year, were beginning readers and completely novices to Code.org. Following Arfé et colleagues (2019; 2020) all children worked alone at their computer in a computer room. Course 2 and course 3 of Programma il futuro was used to train fourth graders (see Table 3.2).

Overview of CS concepts present in Code.org's exercises. Code.org's exercises use a block-based programming language based on Google Blockly, a library from Google for building beginner-friendly block-based programming languages. Blockly's code blocks are expressive enough to represent a variety of complex algorithmic concepts (loops, conditionals, functions, etc.), which we summarize in the next paragraphs.

Loops

A loop command identifies a section of code (a sequence of computational steps) that may be repeated according to:

- A specific termination condition which may or may not occur (this is usually called a while loop, as in "while this condition holds, do this").

- A specific number of iterations have occurred (this is usually called a for loop, as in "repeat this action for n times").

It should be noted that loops can contain other loops (in other words, loops can be nested); this is also true for conditionals and other control structures. For a visual summary and example cfr. Figure 3.3.

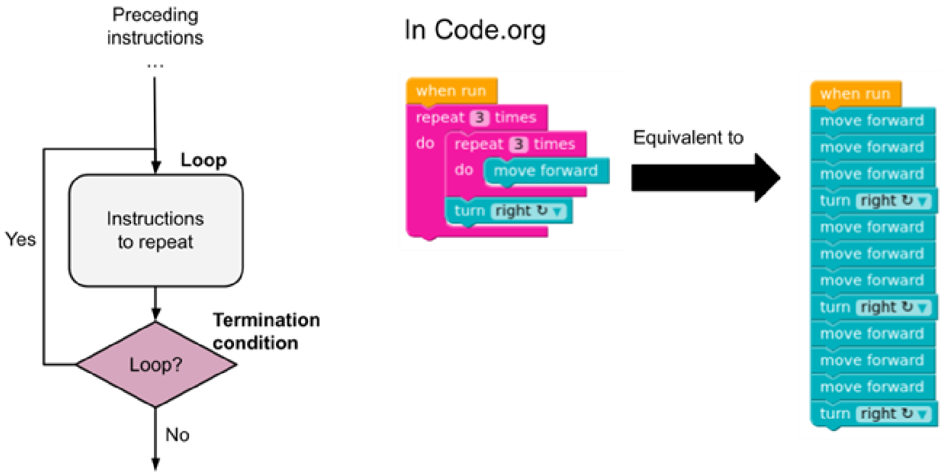


Figure 3.3. *Loop command schematization and example from Code.org*

Conditionals

A conditional command creates a branching in the flow of execution of a program. In other words, after a conditional instruction, the program could continue along one path or another, depending on a specific condition. The branching does not have to be binary: conditional instructions can be sequenced to create an arbitrary number of possibilities (as in "if condition 1 occurs, follow path 1; if condition 2 occurs, follow path 2; if condition 3 occurs, … and so on"). An example and visual summary of this kind of command is presented in Figure 3.4.

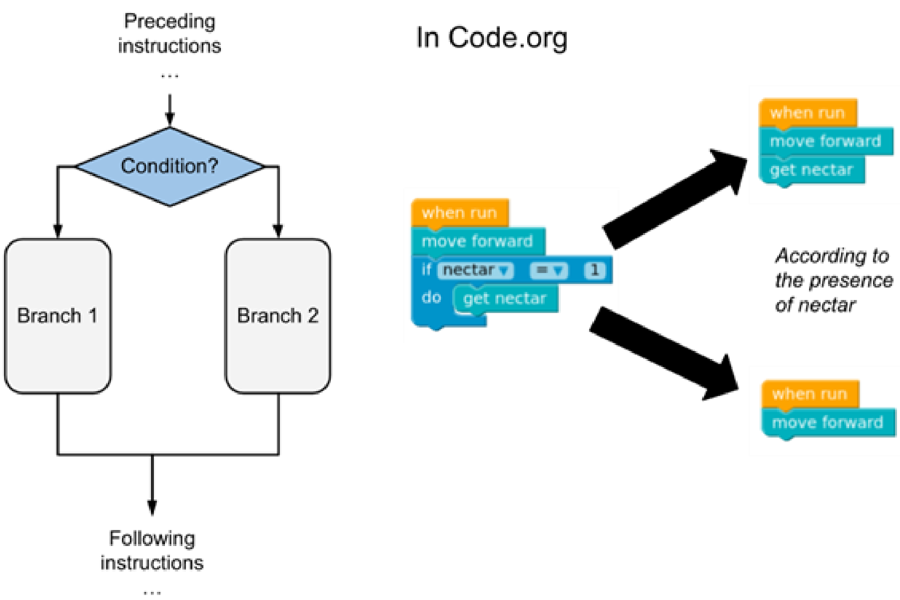


Figure 3.4. *Conditional command schematization and example from Code.org*

Functions

A function is a piece of code that can be referenced by name in other parts of the code. When a running program calls a function, it saves its current position in the sequence of instructions, jumps to the body of the function (i.e., its sequence of operations), executes it, and finally it jumps back to its original position (actually, it jumps to the position after the invocation of the function). Functions can be parameterized, i.e., they can receive some input data and thus provide a generalized solution for a class of problems (e.g., a function could compute the mean of a series of numbers of arbitrary length).

Figure 3.5 shows a schematization of the function construct and an example from Code.org; note that, in the schematization, the color of the arrows determines their order: the blue arrow entering the function body is followed by the blue arrow that exits the body, and so on.

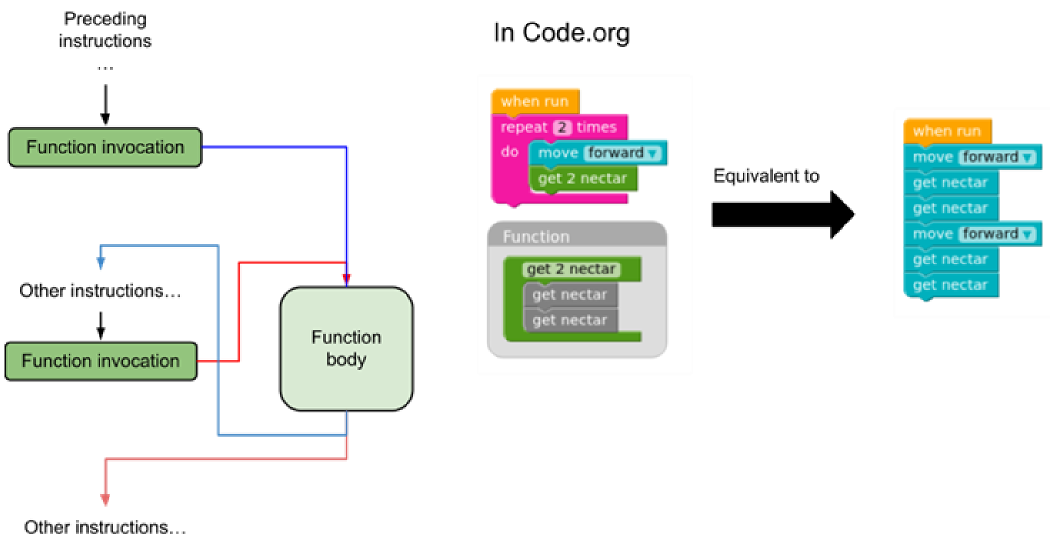


Figure 3.5. *Function command schematization and example from Code.org*

Difference in CT activities for first- and fourth- graders.

The coding exercises varied between first graders and fourth graders due to their age levels. However, the intervention structure, duration, and functions trained remained consistent for both grades, except for the conditionals. First-grade students were only provided with visual blocks whose function is represented by images and icons, such as the arrows on the blocks shown in Figure 3.2. This limited the complexity of the effects that can be executed by these blocks, as it is difficult to express them without words. In contrast, fourth-grade students are provided with blocks that have functions expressed with words, such as 'move forward' and 'turn left', allowing them to access more complex functions, including conditionals.

Moreover, sprite movement for first-graders is always absolute: the available instructions are (move) north, south, east, west; it is interesting to note that this means that, depending on the state of the sprite (i.e. the direction it is facing), a code block could represent more than a single action (e.g. for a sprite facing south the command “move west” actually includes the two actions “turn right” and “move forward”). Fourth graders are provided with blocks of code that represent relative movement (i.e., “turn right/left”, “move forward”). Thus, to some degree, they need to apply visuospatial skills in order to correctly plan the sprites’ movements.

Fidelity check.

Direct monitoring and a checklist were used to ensure fidelity to the lesson plan and consistency of procedures across class groups. The checklist consisted of two yes/no questions and fifteen questions on a five points Likert scale. The checklist assessed whether all key lesson components were performed by the experimenter during the training. Training components included (a) performing all planned coding trials, (b) reading aloud task instructions to the class, (c) encouraging the child to focus on task, and (d) ensuring classroom discussion of game solutions. Master students, blind to the research questions of the study, randomly observed 30% of the lessons recording on the checklist Adherence to the lesson plans and procedures ranged across classes from 98% to 100%.

Table 3.2

Lessons plan

Coding sessions First grade	Course 1	Trial number	Content
Session 1	Lesson 3	1, 2, 3, 4, 5, 6	Jigsaw: Drag and Drop
	Lesson 4	2, 5, 6, 7	Maze: Sequence
Session 2	Lesson 4	8, 10	Maze: Sequence
	Lesson 5	3, 4, 5, 6, 7	Maze: Debugging
Session 3	Lesson 5	8, 9,10	Maze: Debugging
	Lesson 8	4, 5, 6, 7, 8	Artist: Sequence
Session 4	Lesson 8	9, 10, 11	Artist: Sequence
	Lesson 10	4, 5, 6, 7, 8	Artist: Shapes
Session 5	Lesson 13	1, 2, 3, 4, 5, 6, 7	Maze: Loops
Session 6	Lesson 13	8, 9, 10, 11, 12	Maze: Loops

Session 7	Lesson 14	3, 5, 6, 7, 8, 9	Bee: Loops
Session 8	Lesson 18	2, 4, 5, 6, 7	Artist: Loops
Closing session	Classroom discussion	What have we learned?	Metacognitive reflection on the goals of computational thinking and the meaning of programming
Coding sessions Fourth grade	**Course 2**	**Trial number**	**Content**
Session 1	Lesson 3	2, 3, 6, 9	Maze: Sequence
Session 2	Lesson 6	3, 6, 9, 10	Maze: Loops
Session 3	Lesson 8	1, 3, 6, 8	Bee: Loops
Session 4	Lesson 10	2, 4, 6	Bee: Loops, Debugging
Session 5	Lesson 13	2, 4, 6, 7	Bee: Loops, Conditionals
	Course 3		
Session 6	Lesson 2	8, 9, 10	Maze: Loops
Session 7	Lesson 6	1, 2, 3	Bee: Loops, Functions
Session 8	Lesson 7	1, 4, 5	Bee: Loops, Conditionals
Closing session	Classroom discussion	What have we learned?	Metacognitive reflection on the goals of computational thinking and the meaning of programming

3.2.4 Pretest and posttest assessment

At the pretest and posttest, children performed cognitive inhibition and planning tasks from standardized tests (Fancello et al., 2013; Korkman et al., 2007; BIA, Marzocchi et al., 2010). In addition, they performed individually four coding problems from Code.org. While the cognitive tests were the same for both age groups, the CT tests and coding training activities were differently suited for different age groups and stages of cognitive development, so we chose to adapt our activities accordingly. The coding exercises were specifically designed for 1st and 4th graders, the number and type of exercises was similar, and the exercises were taken from code.org courses for these age levels.

In addition, three standardized neurocognitive tests were used, two to assess children's response inhibition, and one to assess planning. The use of two tasks for inhibition dimension aimed at verifying whether potential positive effects on inhibition were evident both across dimensions and across tasks within each dimension. In the following each test is described in detail.

Cognitive skills: response inhibition. The NEPPSY inhibition (squares/circles) subtest of NEPSY-II (Korkman et al., 2007), and the Numerical Stroop test of the Batteria Italiana ADHD (BIA, Marzocchi et al., 2010) were used to assess response inhibition skills pre- and post the 8-h coding course.

The squares/circles inhibition subtest of NEPSY-II (Korkman et al., 2007) consists in naming aloud printed figures (circles and squares) displayed on five rows (of eight figures each), saying "circle" for squares and "square" for circles. To respond accurately the child must inhibit her automatic response (i.e., naming "circle" when seeing a circle viceversa). The task is standardized for children aged 3 to 16 and has good reliability. Reliability as reported by the manual is r = 0.79 for inhibition time and r = 0.77 for inhibition errors.

Children's execution time is recorded, and children's performance is scored for:

1) Inhibition time, in seconds: the total time to complete the task.

2) Accuracy: the number of errors and self-corrections made by the child in performing the task.

The Numerical Stroop test of the Batteria Italiana ADHD (BIA, Marzocchi et al., 2010) is a classical stroop task: Digits from 1 to 5 (e.g., the digit 5) are displayed on a table, and repeated in each cell n times (e.g., the digit 5 reproduced 3 times). The child is instructed to screen the table from left to right and say as quickly and accurately as possible how many digits (e.g., how many 5s) are shown in the cell (in the example, "three digits"). In order to perform the task efficiently the child must supress her automatic digit recognition (i.e., inhibit the automatic response "5"). The task is standardized for children aged 6 to 11. Children's execution time is recorded, and the following scores are computed:

1) Inhibition time, in seconds: the time to complete the task.

2) Accuracy: number of errors and self-corrections.

Cognitive skills: planning. The Tower of London, ToL, was used to assess planning (Luciana et al., 2009). We used a version of the ToL standardized for children aged 4-13 (Fancello et al., 2013). To take the test, children interact with a structure composed of a base with three pegs of different height. The examinee must move three beads of different colours (red, blue and green) from one to the other pegs to reproduce a target configuration; the moves must follow a set of rules: (1) moving only one bead at a time; (2) at any time one bead maximum can be placed on the shorter peg, two on the middle peg, a maximum of three on the longer peg; (3) moves must be sequential: when a bead is removed from a peg, it must be inserted on another peg before removing any more beads; (4) a maximum number of moves is allowed on each trial; (5) each task must be solved in under 60 seconds (counted from the moment the examinee receives the visual stimulus of the target configuration) (Fancello et al., 2013).

The full test includes 12 target configurations of increasing difficulty (and number of allowed moves, starting from 2 and arriving at a maximum of 5). Each task starts with the ToL in the same initial configuration. Only one attempt per trial was allowed and all 12 trials were presented. The scores for this test were:

1) Planning time, in seconds, from when the trial is shown to the child until when s/he makes the first move, pulling out the first ball off the stick.

2) Planning accuracy: one point was awarded if the child performed the trial correctly in 1 minute, without breaking any rule; 0 otherwise.

Indices of concurrent validity with other EFs measures are not provided by the test manual. Test-retest reliability indices, such as r = 0.57 for accuracy scores and r = 0.71 for planning times, are reported by Arfé and colleagues (2020). The concurrent validity of the test was determined for a group of children aged six in the same study (Arfé et al., 2020) by considering the correlation between the results of the ToL and Elithorn tests.

Coding skills. Both the experimental and the control group familiarized first with the Code.org platform and the use of the mouse in drag-and-drop trials. The first graders performed the first trial of lessons 4, 5, 8, 14 from Course 1, the fourth graders performed the trials 2 (lesson 3), 3 (lesson 6), 2 (lesson 10), 4 (lesson 13) from Course 2, assisted by a trained examiner (a master student or the author of this thesis). The pretest started after this familiarization phase. In it, the first graders were asked to solve trials 9 (lesson 4), 2 (lesson 5), 3 (lesson 8), 4 (lesson 14) from Course 1, the fourth graders were asked to solve trials 5 (lesson 3), 8 (lesson 6), 4 (lesson 10), 5 (lesson 13) from Course 2, working autonomously at a personal computer. The children's performance was scored for:

1) Time spent planning, in seconds, from the moment the child received the task instructions to the moment s/he moved the first block. A chronometer and -where the computer room resources allowed- a free screen-capture software were used to record planning times.

2) Planning accuracy: a score of 2 was given if the child successfully solved the item at first attempt; 1 on solving it at the second attempt; 0 otherwise.

3.3 Results

3.3.1 Preliminary analysis at pretest

Preliminary t-test analyses were run to explore between-age-groups (first graders and fourth graders) differences in the dependent (EF and coding) variables' pre-test scores. Statistically significant differences between the age groups at the pre-test were found for accuracy on the coding task $t(430) = 9.34$, $p < .001$, and the ToL, $t(431) = -4.68$, $p < .001$. The first-grade group showed a better pre-test performance than the fourth grade in the coding task, while the fourth graders showed a better pre-test performance for planning task on the ToL. Subsequently, for each grade level, pretest differences between the randomly assigned experimental and control group were tested.

Preliminary t-test analyses on first graders revealed that the experimental and control group differed on pre-test planning accuracy (ToL), $t(267) = -2.19$, $p < .05$. The experimental group showed a better pre-test performance ($M=5.09$) than the control group ($M=4.32$). Statistically significant differences between the experimental and control groups at the pre-test were also found for inhibition errors on the Stroop task, $t(267) = -2.02$, $p < .05$. The experimental group made more inhibition errors ($M=8.31$) than the control group ($M=6.68$) at the pre-test.

Preliminary test on fourth graders revealed that the experimental group and the control group did not differ on any dependent measure at the pre-test.

3.3.2 Data analysis

A multilevel analysis was conducted, following the recommendation to use multilevel data analysis (Peugh, 2010) when participants are nested in class groups. Multilevel models are recommended when multiple observations are nested within clusters (e.g., classrooms) or individuals, as with repeated measures (Peugh, 2010).

Planning time and accuracy on coding tasks, planning time and accuracy on the ToL task, and inhibition time and errors on the NEPPSY-II and the numerical Stroop tasks were the dependent measures. Random effects at class and subject levels were first assessed. We used interclass correlation (ICC) statistics (see Table 3.3 A) to test independence assumption. A value of ICC greater than 0 indicates a deviation from the independence assumption, suggesting that scores among students in the same classrooms are correlated (Peugh, 2010).

The fixed effects included the intercept, time, group (experimental or control), grade, socioeconomic status (SES), and interactions such as time x group, time x grade, and time x group x grade. After assessing the variance attributable to random factors, the effects of time, group, grade, and the interactions (time x group, time x grade, time x group x grade) were tested. Planned comparisons were run to further examine the effects of time (pre-post) at grade within each group (experimental and control). Effect sizes were estimated using Cohen's d (1988).

Multilevel analyses were performed to test the interaction between Time (pretest, posttest), Group (experimental, control), and Grade level (first grade, fourth grade). We used grade level to test for the impact of the learning experience for two different age range, and thus to control for age-related effects. SES was used as the covariate for all models to control for SES-related differences.

Table 3.3, divided into two sections as Table 3.3 A and Table 3.3 B, presents the results of the multilevel analysis. Table 3.3 A reports random effects at the classroom and student levels. The findings reveal variance at the student level across all measures, ranging from 7 to 61% as indicated by ICC values. The analysis also shows that the variance attributed to classroom effects range from 5 to 48%. This suggests that the primary source of variability in our dependent measures arises from individual differences among students. Table 3.3 B provides a detailed account of the fixed effects.

In the following, we report the results of the multilevel for each dependent measure (planning time and accuracy at coding tasks, and planning time and accuracy, response inhibition time, and errors at standardized tasks). Multilevel analyses were followed by planned comparisons to further examine pre-post comparisons for each group and each grade. These analyses were run with a descriptive and exploratory aim.

Table 3.3 A

Random Effects: Variance in Coding and Cognitive (Planning and Inhibition) Skills Due to Random Classroom and Subjects' Effects.

			Coding			
	Accuracy			**Planning time**		
Effect	B	SE	%	B	SE	%
Intercept subject	.84	.14		14.02	6.97	
Intercept classroom	.59	.22		74.43	20.80	
Residual	1.96	.13		127.40	8.66	
ICC subject			25			7
ICC classroom			17			35
			Tower of London			
	Accuracy			**Planning time**		
Effect	B	SE	%	B	SE	%
Intercept subject	1.71	.26		.88	.20	
Intercept classroom	1.91	.65		3.59	1.15	
Residual	3.29	.23		2.96	.20	
ICC subject			25			12
ICC classroom			28			48
			Nepsy-II			
	Inhibition errors			**Inhibition time**		
Effect	B	SE	%	B	SE	%
Intercept subject	2.17	.41		63.74	6.11	
Intercept classroom	.44	.23		3.18	2.54	
Residual	5.79	.40		41.41	2.84	
ICC subject			26			59
ICC classroom			5			3
			Stroop			
	Inhibition errors			**Inhibition time**		
Effect	B	SE	%	B	SE	%
Intercept subject	6.73	1.14		1144.94	102.02	
Intercept classroom	1.95	.89		180.35	82.49	
Residual	15.36	1.05		559.11	38.34	
ICC subject			28			61
ICC classroom			8			10

Note. Planning time and inhibition time are expressed in seconds.

Table 3.3 B

*Fixed effects: Variance in coding and cognitive (planning and inhibition) skills controlling for SES, Time, Grade, Group and Time*Grade, Time*Group, Time*Grade*Group effects.*

	Coding					
	Accuracy			**Planning time**		
Effect	B	SE	t (df)	B	SE	t (df)
Intercept	4.65	.46	10.11 (44.83)***	10.82	4.83	2.47 (42.31)*
SES	.13	.05	2.78 (424.67)**	.15	.30	.52 (418.21)
Time	-3.95	.21	-19.13 (425.44)***	5.91	1.67	3.55 (433.19)***
Grade	.62	.50	1.25 (24.01)	-2.95	5.19	-.57 (31.73)
Group	-3.28	.58	-5.64 (23.4)***	3.96	6.10	.65 (31.63)
Time*grade	1.16	.27	4.28 (425.44)***	11.43	2.19	5.23 (433.19)***
Time*group	3.22	.32	10.19 (427.05)***	-2.46	2.55	-.96 (434.98)
Time*grade*group	-1.20	.40	-3.02 (426.45)**	-11.565	3.21	-3.63 (434.32)***

	Planning – Tower of London					
	Accuracy			**Planning time**		
Effect	B	SE	t (df)	B	SE	t (df)
Intercept	7.43	.75	9.97 (35.48)	2.75	.92	2.99 (27.03)**
SES	.22	.06	3.57 (418.63)***	.07	.05	1.27 (412.03)
Time	-2.51	.27	-9.36 (424.68)	-.20	.26	-.80 (427.17)
Grade	-.49	.85	-.58 (22.90)	2.54	1.11	2.28 (22.15)*
Group	-1.67	.98	-1.69 (22.80)	.06	1.31	.05 (22.10)
Time*grade	-.85	.35	-2.42 (423.83)*	-.02	.33	-.06 (426.25)
Time*group	1.55	.41	3.77 (425.06)***	.29	.39	.74 (427.58)
Time*grade*group	.23	.52	.45 (424.17)	.63	.49	1.28 (426.62)

	Inhibition – Nepsy II					
	Inhibition errors			**Inhibition time**		
Effect	B	SE	t (df)	B	SE	t (df)
Intercept	3.04	.58	5.24 (81.68)***	34.27	2.08	16.46 (85.43)***
SES	-.35	.07	-4.80 (389.91)***	-.62	.29	-2.15 (331.91)*
Time	1.35	.36	3.79 (426.80)***	3.44	.95	3.61 (425.94)***
Grade	.69	.54	1.28 (31.11)	13.09	1.75	7.5 (26.59)***
Group	1.11	.64	1.74 (30.79)	-.20	2.04	-.01 (26.40)
Time*grade	1.13	.47	2.42 (425.83)*	4.92	1.25	3.94 (425.33)***
Time*group	-1.32	.54	-2.43 (427.14)*	3.29	2.57	1.28 (26.92)
Time*grade*group	-.17	.69	-.24 (426.20)	-2.05	1.83	-1.12 (425.58)

	Inhibition – Stroop					
	Inhibition errors			**Inhibition time**		
Effect	B	SE	t (df)	B	SE	t (df)
Intercept	4.33	1.05	4.11 (65.04)***	118.55	10.08	11.76 (62.90)***
SES	-.33	.12	-2.63 (411.19)**	-2.87	1.23	-2.34 (409.75)*
Time	2.65	.58	4.56 (426.89)***	15.41	3.50	4.40 (425.76)***
Grade	.39	1.03	.38 (26.48)	65.72	9.56	6.86 (23.23)***
Group	1.77	1.21	1.46 (26.22)	3.46	11.23	.31 (23.01)
Time*grade	2.95	.76	3.87 (426.01)***	22.89	4.59	4.99 (425.34)***
Time*group	-1.83	.89	-2.06 (427.29)**	1.36	5.36	.26 (426.00)
Time*grade*group	-2.18	1.12	-1.95 (426.37)	-14.57	6.73	-2.16 (425.45)*

Note. * $p < .05$; ** $p \leq .01$; *** $p \leq .001$; B = Beta; SE = Standard error.

3.3.3 Effects of learning to code on coding skills

Accuracy coding. The covariate SES was significant ($B = 0.12$, $p < .05$). The main factors Time and Group were also significant (respectively, $B = -3.95$, $p < .001$; $B = -3.28$, $p < .001$). The two-way interactions Time x Grade and Time x Group were significant (respectively, $B = 1.16$, $p < .001$; $B = 3.22$, $p < .001$). Also, the three-way interaction Time x Grade x Group was significant ($B = -1.20$, $p < .05$). The interaction was examined by planned comparisons, reported in Table 3.4. Planned comparisons showed that in grade one accuracy on the coding task was significantly greater at the posttest both in experimental group and control group. However, the effect size was larger for the experimental group, $t(425) = -15.87$, $p < .001$, $d = -1.539$, than for the control, $t(425) = -4.63$, $p < .001$, $d = -.449$. The experimental group exhibited a large effect size, while the control group exhibited a medium effect size. Also in grade four, the performance was significantly greater for experimental group at the posttest. Again, the experimental group showed a significantly larger difference between pretest and posttest compared to the control group. The experimental group had a large effects size compared with the small effect size of the control group ($d = -1.855$, for the experimental and $d = -.293$ for the control). Figure 3.6 displays changes in performance between the pretest and posttest in first and fourth graders for coding accuracy.

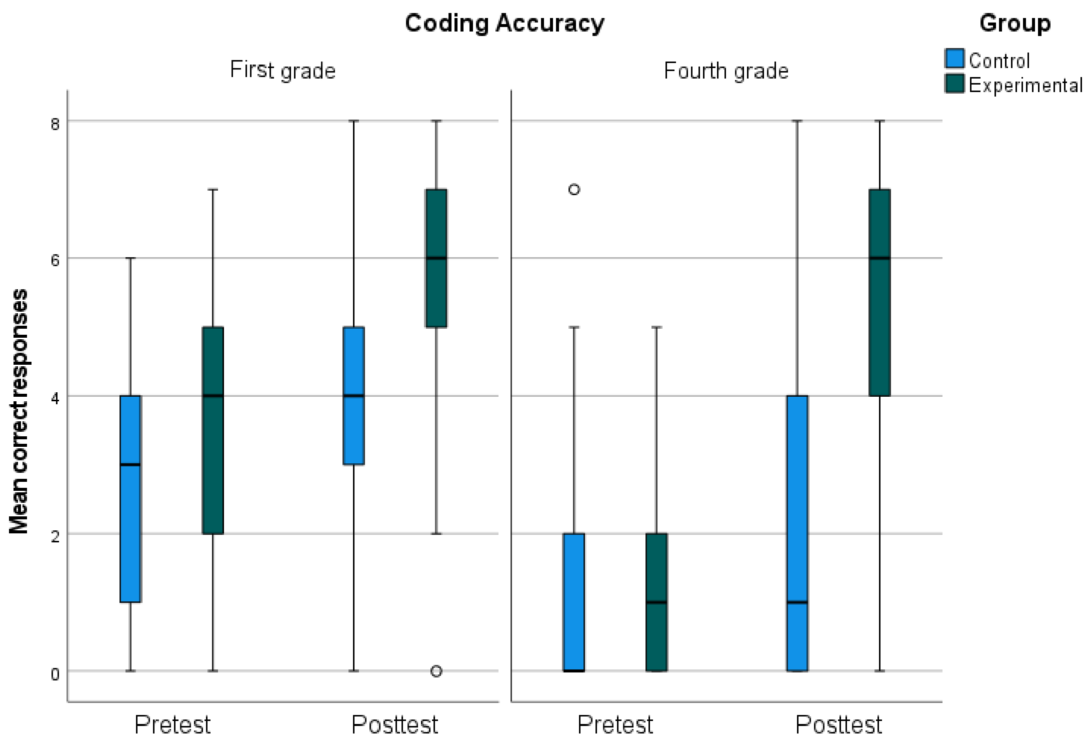


Figure 3.6. *Groups' accuracy on coding tasks at the pretest and posttest (SES covariate).*

Planning time coding. The main factor Time was significant (B = 5.90, $p < .001$). The two-way interaction Time x Grade was significant (B = 11.43, $p < .001$). The three-way interaction Time x Grade x Group was also significant (B = -11.65, $p < .001$). Planned comparisons showed significant differences at the posttest between experimental and control group. Specifically, in grade one's experimental group planning time in the coding task decreased significantly between the pre and posttest, while the pre-posttest differences was no significant in the control group. Additionally, the coding planning time in first graders' experimental group decreased more than in the grade four's experimental group, $t(433) = 12.24$, $p < .001$, $d = 1.18$ (first graders), $t(433) = 3.55$, $p < .05$, $d = 0.34$ (fourth graders).

3.3.3 Effects of learning to code on executive functions

EFs: Accuracy planning. The covariate SES was significant (B = 0.22, $p < .001$). The main factor Time was also significant (B = -2.51, $p < .001$). The two-way interactions Time x Grade and Time x Group were significant (respectively, B = -.85, $p < .05$; B = 1.55, $p < .001$). The three-way interaction Time x Grade x Group was nonsignificant. Planned comparisons showed that in grade one accuracy in planning task was significantly greater at the posttest in both groups, but the effect size was larger for the experimental group, $t(423) = -14.78$, $p < .001$, $d = -1.44$, versus $t(423) = -7.37$, $p < .001$, $d = -0.72$, for the control group. Similar effects were found for the fourth graders, with larger pre-posttest differences for the experimental group, $t(425) = -9.36$, $p < .001$, $d = -0.91$, versus $t(425) = -3.11$, $p < .05$, $d = -0.30$ for the control group, indicating that also at this grade level, the students benefitted from the intervention. In both first grade and fourth grade the effects of the intervention were large. Figure 3.7 displays changes in performance between the pretest and posttest in first and fourth graders for planning accuracy.

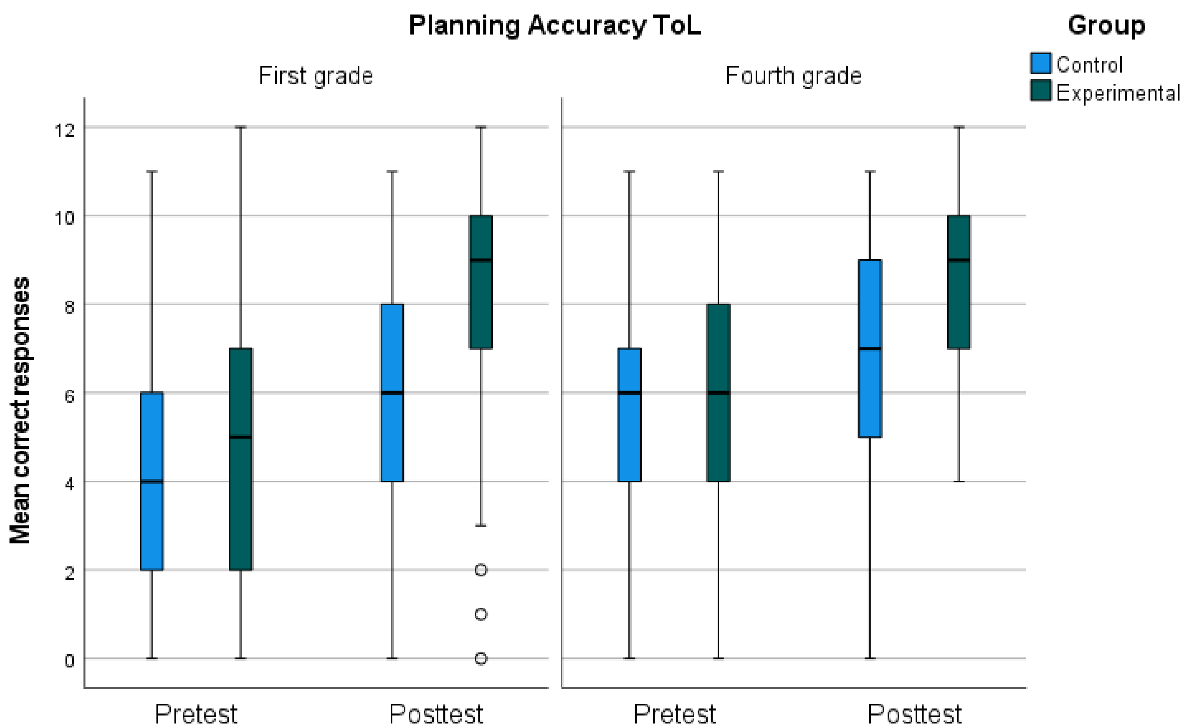


Figure 3.7. *Groups' accuracy on planning task at the pretest and posttest (SES covariate).*

EFs: Planning time. The main factor Grade was significant (B = 2.54, $p < .05$). The two-way interactions Time x Grade and Time x Group were nonsignificant. Also, the three-way interaction Time x Grade x Group was nonsignificant.

EFs: Inhibition. On the NEPSY-II, inhibition subtest, inhibition errors decreased significantly with time. The main factor Time was significant (B = 1.35, $p < .001$). The covariate SES was also significant (B = -0.35, $p < .001$). The two-way interactions Time x Grade and Time x Group were significant (respectively, B = 1.13, $p < .05$; B = -1.32, $p < .05$). The three-way interaction Time x Grade x Group was nonsignificant. Planned comparisons showed that grade one inhibition errors decreased significantly at the posttest in both groups. However, the difference between pretest and posttest was larger for the experimental group; the dimension of the effect for the control group was small: ($t(425) = 8.21$, $p < .001$, $d = 0.80$, for the experimental, and $t(425) = 3.48$, $p < .05$, $d = 0.34$, for the control). Planned comparisons showed that in grade four inhibition errors decreased significantly at the posttest only for the experimental group, $t(427) = 3.79$, $p < .001$, $d = 0.38$. The significant two-way interactions indicate that also at this grade level, the students benefitted from the intervention.

On the NEPSY-II task, inhibition time decreased significantly with time for both grades and groups. The main factors Time and Grade were significant (respectively, B = 3.44, $p < .001$; B = 13.09, $p < .001$). The covariate SES was also significant (B = -0.62, $p < .05$). The two-way interaction Time x Grade was also significant (B = 4.92, $p < .001$), while the interaction Time x Group was nonsignificant. The three-way interaction Time x Grade x Group was also nonsignificant.

The exploratory planned comparisons showed that in grade one inhibition time decreased significantly at the posttest in both experimental group and control group. However, the dimension of the effect was slightly different between the experimental and control groups: (t (424) = 10.36, p

$< .001$, $d = 1.00$, for the experimental, and $t(424) = 8.86$, $p < .001$, $d = 0.86$, for the control). Also in grade four inhibition time decreased significantly with time for both group, $t(426) = 3.61$, $p < .001$, $d = 0.35$ for the experimental group, and $t(426) = 3.53$, $p < .05$, $d = 0.34$ for the control.

On the Stroop task (errors), the main factor Time was significant ($B = 2.65$, $p < .001$). The two-way interactions Time x Grade and Time x Group were significant (respectively, $B = 2.95$, $p < .001$; $B = -1.83$, $p < .05$). The three-way interaction Time x Grade x Group approached significance ($B = -2.18$, $p = .05$). Planned comparisons showed that inhibition errors were significantly lower at the posttest in both groups, but the effect size was larger for the experimental group, $t(437) = -12.53$, $p < .001$, $d = -0.60$, versus $t(419) = -4.19$, $p < .001$, $d = -0.20$ for the control groups.

Exploratory planned comparisons showed that in grade one inhibition errors were significantly lower at the posttest in both groups, but the effect size was larger for the experimental group, $t(425) = 11.37$, $p < .001$, $d = 1.10$, versus $t(425) = 3.41$, $p < .05$, $d = 0.33$ for the control group. The exploratory planned comparisons for the fourth graders showed a pre-posttest significant difference in inhibition errors only for the experimental group, $t(427) = 4.56$, $p < .001$, $d = 0.44$, indicating that at this grade level, only the experimental group students benefitted from the intervention.

For inhibition time (Stroop task) the main factors Time and Grade were significant (respectively, $B = 15.40$, $p < .001$; $B = 65.72$, $p < .001$). The two-way interaction Time x Grade was significant ($B = 22.88$, $p < .001$); the three-way interaction Time x Grade x Group was also significant ($B = -14.56$, $p < .05$). Planned comparisons showed that in grade one inhibition time decreased significantly at the posttest in both groups. However, the dimension of the effect varied between the experimental and control groups: ($t(425) = 12.90$, $p < .001$, $d = 1.25$, for the experimental, and $t(425) = 8.94$, $p < .001$, $d = 0.86$, for the control). In grade four inhibition time decreased significantly with time for both group, $t(426) = 4.40$, $p < .001$, $d = 0.43$ for the experimental group, and $t(426) = 4.14$, $p < .001$, $d = 0.40$ for the control. The dimension of the effect was lower than for first graders and similar for the experimental and control group. Figure 3.8 displays changes in performance (Stroop task accuracy) between the pretest and posttest in first and fourth graders.

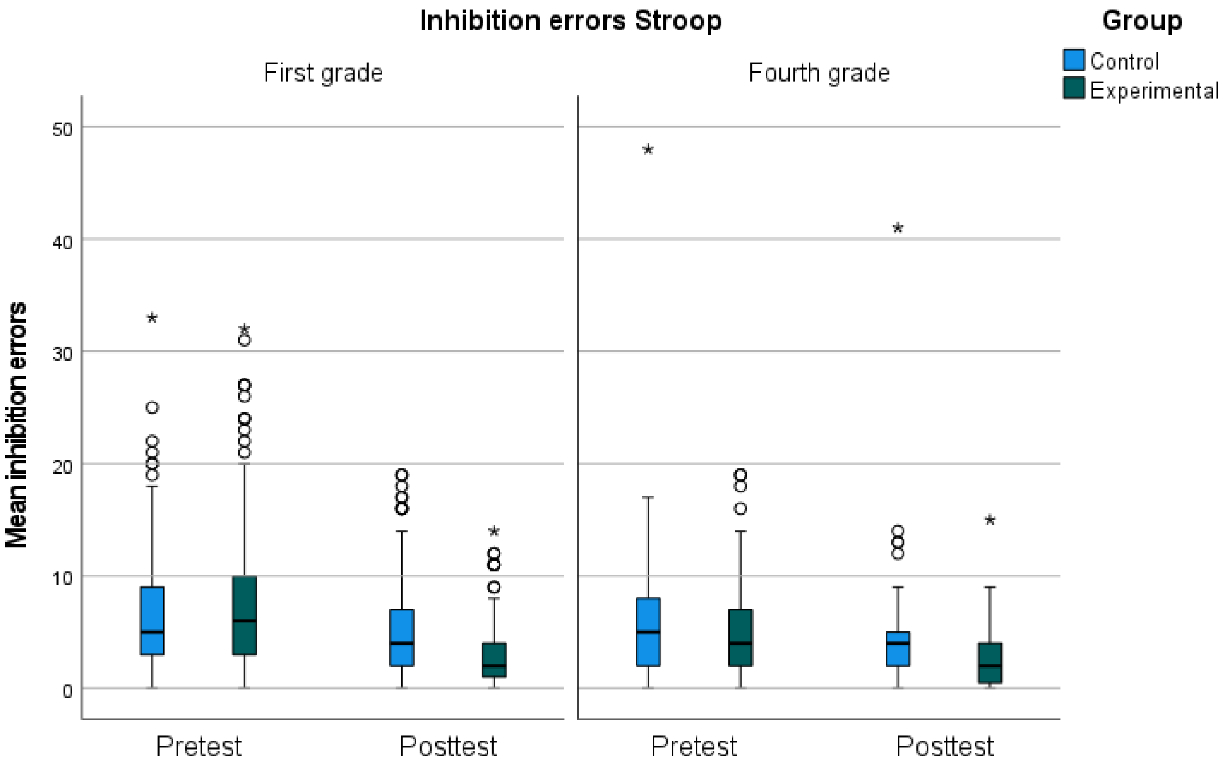


Figure 3.8. *Groups' inhibition errors on Stroop inhibition task at the pretest and posttest (SES covariate).*

Table 3.4

Planned comparisons: Mean Differences between T2 (posttest) and T1 (pretest) and Effect Sizes (Cohen's d) for the Control and Experimental Group for first and fourth grade.

Variable	Grade	Control group				Experimental group			
		Mean Diff. (T1-T2)	SE Diff. (T1-T2)	t (DF)	Cohen's d	Mean Diff. (T1-T2)	SE Diff. (T1-T2)	t (DF)	Cohen's d
Coding									
Accuracy	First	-.77	.17	-4.63 (425)***	-.45	-2.79	.18	-15.87 (425)***	-1.54
	Fourth	-.73	.24	3.03 (428)**	-.29	3.95	.21	-19.13 (425)***	-1.86
Planning time	First	3.23	1.34	2.41 (433)	.23	17.34	1.42	12.24 (433)***	1.18
	Fourth	3.45	1.93	1.79 (436)	.17	5.91	1.66	3.55 (433)**	.34
Tower of London									
Accuracy	First	-1.56	.22	-7.37 (423)***	-.72	-3.36	-.23	-14.78 (423)***	-1.44
	Fourth	-.96	.31	-3.11 (425)**	-.30	-2.51	.27	-9.36 (425)***	-.91
Planning time	First	.69	.20	3.38 (425)**	.33	-.22	.22	-1.03 (425)	-.10
	Fourth	.08	.29	.28 (428)	.03	-.20	.26	-.80 (427)	-.08
Nepsy-II									
Inhibition Errors	First	.99	.29	3.48 (425)**	.34	2.48	.30	8.21 (425)***	.80
	Fourth	.03	.41	.07 (427)	.01	1.35	.36	3.79 (427)***	.37
Inhibition time	First	6.76	.76	8.86 (424)***	.86	8.36	.81	10.36 (424)***	1.01
	Fourth	3.90	1.10	3.53 (426)**	.34	3.44	.95	3.61 (426)***	.35
Stroop									
Inhibition Errors	First	1.59	.47	3.41 (425)**	.33	5.59	.49	11.37 (425)***	1.10
	Fourth	.82	.67	1.22 (428)	.12	2.65	.58	4.56 (427)***	.44
Inhibition time	First	25.01	1.81	8.94 (425)***	.87	38.29	2.97	12.90 (425)***	1.25
	Fourth	16.77	4.05	4.14 (426)***	.40	15.41	3.50	4.40 (426)***	.43

Note: Mean Diff (T2-T1) = Mean differences between posttest (T2) and pretest (T1); SE Diff. (T2-T1) = Standard Error differences between T2 and T1; * $p < .05$, ** $p \leq .01$, *** $p \leq .001$

3.4 Discussion

The purpose of this study was to assess the effect of an 8-week plugged-in coding intervention on the development of executive functions and CT abilities of first and fourth graders. As discussed earlier, there is a dearth of research regarding age-related effects in response to coding instruction. A few studies suggest that older students have greater benefits -in coding abilities- than younger students from CT interventions (e.g., Atmatzidou & Demetriadis, 2016; Jiang & Wong, 2022). The authors of those studies speculated that this trend could be attributed to cognitive development and increased problem-solving abilities. The problem-solving skills of older students may enable them to engage more effectively with and benefit from practicing CT activities. If coding activities require significant cognitive resources, then the benefits of such activities may also vary with age, depending on the child's cognitive development. Previous experimental studies provide evidence of a causal relationship between the teaching of CT and the improvement of EF skills also in younger children (e.g., Arfé et al., 2019; Arfé et al., 2020; Di Lieto et al., 2020). However, to date no studies have compared the effects of CT intervention across ages.

The main findings of this study lead to conclude that CT interventions are effective across grade levels, improving coding and executive functions.

3.4.1 Coding abilities

The CT intervention was essentially effective for children of both grade levels, even though the pre-post difference was slightly higher for the experimental group of fourth graders exposed to the training. However, the experimental group of first graders also improved their coding skills at the posttest, with a large effect size. Our results differ thus from those of Kyza et al. (2022) and Rjike et al. (2018), who concluded that the acquisition of CT concepts only occurs around ages 9-10. In contrast to these studies, we found substantial improvements in coding accuracy not only in the fourth graders but also in first graders, and a greater effect of the intervention on first graders' coding planning time. The present study found that first-grade students who received CT intervention showed a greater decrease in planning time (d=1.10) compared to fourth-grade students (d=0.34). The significant decrease in first graders' planning time is also associated with an increase in accuracy when solving coding exercises, confirming that CT intervention can be very effective from this early age (see Arfé et al., 2020; Di Lieto et al., 2020).

3.4.2 Planning abilities

Similarly, also on the ToL planning task, both the first graders' experimental group and fourth graders' experimental groups improved their planning accuracy with the intervention. This finding aligns with those of previous studies (Arfé et al., 2019, 2020; Robledo-Castro et al., 2023) and indicates that the effectiveness of the CT intervention for boosting planning skills was consistent across grade levels. Since CT activities like the ToL involve non-verbal planning, these effects could be either interpreted as near or far transfer effects. All experimental groups performed significantly better than the control groups at the post-test. This suggests that are no age-related differences for planning abilities in response to coding intervention. Despite the complexity of the Tower of London (ToL) task, which could potentially present greater difficulty for first graders attributed to the significant involvement of working memory in the sequencing of the plan during problem resolution, the first graders exhibited a good performance with an effect size nearly equivalent to that of fourth graders. Considering the substantial developmental spurt of the planning skills at the age 9 (Luciana et al., 2009; McGuckian et al., 2023), it would have been expected that fourth graders would have shown greater improvement in planning skills compared to first graders, where planning is just starting to develop. The results of our study suggest that in one month of CT intervention, planning skills can undergo a significant increase in children, both at age 6 and at age 9. This is a key finding, considering that planning represents a higher-order EF skill, which involves core EF. Therefore, improvements in planning may potentially transfer to core EF skills. The effects of the intervention on children's response inhibition skills appear to support this interpretation.

3.4.3 Inhibition

The transfer of coding effects to first graders' inhibition skills has been observed earlier (Arfé et al., 2019; Arfé et al., 2020; Di Lieto et al., 2020). The results of our study confirm these findings, as we observed a significant improvement in response inhibition tasks among children who received the coding intervention between pretest and posttest. This improvement was evidenced by a decrease in inhibition errors across tasks (i.e., the subtest of the NEPSY-II and the Stroop) and grade levels, with a moderate to large effect size. In the first graders, both the experimental and control groups showed greater accuracy on the inhibition tasks at the posttest. However, the experimental group showed significantly larger gains (with a large effect size) compared to the control group (with small effect size). As expected, the decrease in inhibition errors in the control group aligns with the typical learning patterns of response inhibition, which can be attributed to the familiarity effect with the task. The positive effects of the training on children's inhibition skills were expected. It is well-established that the most substantial development of core executive functions, particularly inhibitory control, occurs during the preschool years, spanning from 3 to 6 years (referencing Carlson, 2005; Scionti et al., 2020; Traverso et al., 2015), as well as during the transition to primary school (Garon et al., 2008; Macdonald et al., 2014; Zelazo et al., 2003). This may explain the benefit derived by the CT intervention.

In the fourth graders, only the experimental group showed a significant decrease in inhibition errors at the posttest, across the two tasks, but with a smaller effect size compared to the first graders. The significant decrease in inhibition errors across the two tasks (i.e., NEPSY-II subtest and Stroop) suggest that the CT intervention strongly influenced the children's inhibitory control.

With respect to the inhibition time on the NEPSY-II task, similarly to the inhibition errors, the effectiveness of the CT intervention was consistent across grade levels. In the Stoop task the results were confirmed by the significance of the three-way interaction which revealed that the decrease in inhibition time differs for grade levels and groups. In fact, the planned comparison revealed that the intervention had a greater impact on first graders than on fourth graders, as evidenced by the larger effect size which was almost triple than that of the fourth graders group. In the fourth graders the pre-posttest difference was not differ across the experimental and the control group. The finding that early primary school children, following exposure to an 8-hour coding intervention, enhanced speed in inhibiting automatic responses is a promising indication of the intervention's impact on EFs. This improvement is not limited to the NEPSY-II subtest but extends also to the more complex tasks as the Stroop, emphasizing the robustness of the intervention's influence.

This study thus confirms the effectiveness of CT intervention in boosting first graders' inhibition skills (Arfé et al., 2019, 2020; Di Lieto et al., 2020) and extend these results to older children, demonstrating that CT interventions can have strong effects even at later stage of schooling and cognitive development.

Overall, the results of the study revealed wide benefits of CT activities, with similar effectiveness observed across grade levels, indicating that the response to the intervention does not vary with age. This finding highlights the need for introducing early coding interventions and continuing them throughout all years of primary school.

3.5 Limitations and Future Directions

The study has several limitations that should be acknowledged.

First, despite our sample size is large, this sample may not fully represent the diversity of early primary school populations. Socioeconomic and cultural factors can influence the response to coding interventions, and caution is needed when extrapolating the findings to a broader range of demographic groups. Especially, the sample of fourth graders was less differentiated by socioeconomic status than the sample of first graders. In fact, preliminary analyses showed statistically significant differences in SES between the experimental and control groups of the fourth graders, which we accounted for by using SES as a covariate. Furthermore, the influence of socioeconomic status on accuracy coding emphasizes the importance of considering broader contextual factors in educational interventions. The significance of the SES covariate underscores the importance of considering socioeconomic factors when examining the effects of interventions on executive functioning. The link between SES and EFs skill is consistent with existing literature (Blair & Raver, 2016), emphasizing the multifaceted nature of cognitive development and the need to consider social variables when interpreting intervention outcomes.

Second, our research specifically targeted primary school children and the groups were not optimally matched at the pretest. However, we employed multilevel models as robust tools to address this limitation. It is crucial to recognize that despite the robustness of multilevel models, they may not completely eliminate the effects of confounding variables. The initial mismatch between groups could lead to biased estimates, which may affect the generalizability of our findings.

Third, regarding the tool used to assess both the coding and the EFs' skills, it is important to acknowledge some limitations that may influence our findings. While our study aimed to assess coding skills, the limited number of items in the assessment may not adequately capture the full variability among participants. A more extensive set of items, strategically designed to encompass various aspects of computational thinking processes like sequencing, debugging, and functions, would have provided a more nuanced and comprehensive evaluation of participants' coding abilities. The additional coding items could have been tailored to address specific CT components, allowing for a more detailed analysis of first graders and fourth graders' skills in each area. For example, including items that focus on the ability to logically sequence code and to identify and correct errors (debugging) would contribute to a more thorough understanding of participants' coding skills.

Regarding the planning assessment, we could not administer a second planning test in addition to the ToL task due to the length of the individual assessment sessions, which lasted approximately thirty to fifty minutes for each child. Although the ToL task is a standardized test that provides valuable insights about the planning dimension, it is always recommended to employ multiple tests for the same dimension when assessing EFs.

Fourth, although CT interventions have shown promise in enhancing EF skills, their broader impact on other cognitive processes remains unclear. It is crucial to consider the limited generalizability of the effects of EF interventions to domains beyond those directly targeted by the training. Although cognitive training may improve EF skills, it is unclear whether these improvements translate into meaningful changes in children's quality of life. For example, it is uncertain whether improved EF skills lead to better academic performance, social interactions, or emotional regulation. Additionally, it is important to investigate whether these enhancements facilitate learning processes or adaptation to various educational settings. Furthermore, it is necessary to investigate whether enhancing EF skills can provide a compensatory mechanism for children at risk of neurodevelopmental disorders. Can improved EF skills alleviate the effects of challenges such as learning disabilities, attention deficits, or behavioral disorders? It is essential to assess the potential compensatory effects of EF interventions to develop targeted interventions that address the multiple needs of children in educational settings.

These considerations about the findings of our study suggest the need for further research.

By comparing our coding intervention results with those of other interventions aimed at enhancing cognitive function, researchers can identify similarities and differences in cognitive outcomes. These comparisons could help determine whether coding interventions provide unique benefits or share similarities with interventions designed for other cognitive domains. Understanding these relationships can guide the development of tailored interventions and inform practitioners and educators about the transferability of skills gained in coding interventions to broader cognitive functions.

Additionally, a comparative analysis would contribute to the growing body of knowledge on the generalizability of interventions targeting specific cognitive functions. This information is essential for improving intervention strategies, identifying effective practices, and maximizing the impact of cognitive enhancement programs.

Furthermore, it is also necessary to consider the broader impact on children's educational outcomes and overall well-being. Future research should include measures that evaluate the transferability of EF improvements to several life contexts, such as academic achievement, social functioning, and emotional regulation. Future studies should consider stratifying the sample based on participants' levels of executive function performance, distinguishing between low and high EF performers. This approach would yield valuable data to highlight the potential benefits of CT training activities, particularly in contexts aimed at rehabilitation or intervention for children at risk for neurodevelopmental disorders.

CHAPTER 4. OVERALL DISCUSSION

This PhD project provided an overview of the computational thinking (CT) intervention and its impact on the cognitive skills of children and adolescents. In the previous chapters, the theoretical framework, the systematic review of the literature in the field, and two intervention studies have been presented and discussed. The next paragraphs will highlight the main key findings arising from the investigations conducted within the overall PhD project.

The aim of the *Study 1* was to take stock of existing literature in the field. A systematic review and meta-analytic study on the cognitive effects of the development of computational thinking in children and young people was carried out to support the literature update of the present research project. The main goal of this meta-analytic study was to synthesize the existing randomized controlled trials (RCT) that have examined the effectiveness of CT interventions on the cognitive abilities of children from 4 to 16 years. *Study 1* reported an in-depth review of these experimental studies in which the cognitive effects of CT interventions were assessed in children and adolescence.

As the meta-analytic study shows, CT interventions were found to be generally effective in boosting higher-order EFs skills in children and adolescents. The largest effects were observed in students' problem-solving and complex EFs such as planning, although positive effects were also found for core EFs like cognitive inhibition and working memory, with a lower effect size.

CT interventions were found to be equally effective for older and younger children, with beneficial effects on problem-solving consistently reported across age groups. Older students improved in problem-solving tasks, self-reported problem-solving, and metacognitive skills, while younger children showed significant transfer effects for complex and core EF skills.

The effectiveness of CT activities was not strictly related to the method or programming tool used but was influenced by the structured or unstructured nature of the intervention. Virtual coding interventions were effective across a broad age span, with positive effects on problem-solving and core EFs observed in children from first to eight grades. Educational Robotics (ER) interventions showed positive far transfer effects on EF skills, especially in younger children, but the effects varied based on the specific ER activities and the age group.

The systematic review and meta-analysis shed light on the profound impact of CT interventions on executive functions, while also highlighting some gaps in the research field: (1) there was a lack of studies investigating the cognitive effects of CT interventions on preschoolers' core executive functions and (2) research was insufficient to statistically test the effects of age and intervention type through moderator analyses. Further research was needed to determine:

(1) whether the various instructional methods used to teach CT skills—such as plugged-in coding, unplugged coding, and educational robotics (ER)—have different efficacy;

(2) whether age differences in learning to code translate also in differences in the cognitive benefits of coding.

The second study (*Study 2*) presented in this thesis was a cluster randomized controlled trial and focused on the effectiveness of a 7-week combined unplugged coding and ER intervention on coding, response inhibition, planning, and visuo-spatial skills in preschool children. The intervention significantly improved coding abilities and visuo-spatial skills, with evidence of both near-transfer effects on coding skills and far-transfer effects on visuo-spatial skills. While improvements in coding and visuo-spatial skills were observed, the study did not find significant effects on response inhibition, suggesting variability in the transfer effects of CT interventions.

Study 3 moved from the existing lack of studies exploring the differential effects of CT interventions with age. Therefore, the cluster randomized controlled trial assessed the age differences in response to CT interventions.

The findings suggested that a CT intervention based on plugged-in coding activities improves the acquisition of coding skills in both first graders and fourth graders. Moreover, the effectiveness of the intervention was also large across grade levels in improving children's inhibition and planning skills. In summary, the study contributes some initial insights into the age-related dynamics of the impact of plugged-in coding interventions on computational thinking abilities and executive functions in children. Although other individual differences such as those related to delays in cognitive development or developmental disorders were not examined in the study, a focus on these individual factors represent a promising avenue for future research.

Research on the cognitive benefits of CT and coding interventions, particularly through randomized trials, is still limited. Existing studies suggest positive effects on computational thinking, problem-solving, and executive functions. However, most of the existing literature is based on non-experimental or quasi-experimental studies. This represents a major gap and limitation of the field. There is indeed need for more robust experimental evidence. The studies presented in this thesis addressed this gap by examining the consistency and generalizability of the effects of CT interventions across various executive functions and age groups.

Gathering evidence on the cognitive effectiveness of coding activities will impact on the design of instructional interventions for the school curricula. A recent meta-analysis (Kassai et al., 2019) shows that there are limited practical benefits - other than on the trained component - of training single executive function components in childhood. Thus, it might be more advisable, both in educational and in clinical practice, to use approaches like coding, that target multiple executive function components. As preschool years are one of the windows of greatest plasticity for EFs, coding interventions in this time span could have a powerful cascading effect on children's future developmental trajectories and not only on a more "active" approach toward digital technologies.

Executive functions have been shown foundational for children's school achievements (Altemeier et al., 2008; Purpura et al., 2017; Roebers et al., 2011; Spiegel et al., 2021; Vandenbroucke et al., 2017), their understanding of others' mind (Benson et al., 2013), as well as self-regulation (Luciana et al., 2009; Mägi et al., 2016; Marlowe, 2000). Therefore, boosting the development of these skills from an early age can also have potentially broad effects on the child's academic achievements and global well-being. By increasing our understanding of the effectiveness of CT activities we can help develop better CT and coding programs to promote children's coding and cognitive skills. This research will be thus useful for the design of learning environments that can effectively support and enhance children's cognitive development.

Although our study sheds light on the effectiveness of coding interventions, future research should extend its scope by directly comparing the results of CT interventions with those from other types of interventions targeting EFs. Future studies should also explore who benefits most from coding interventions, by focusing on individual differences in cognitive profiles and/or low and high EF performers. This data would help to provide insights for the use of CT activities for rehabilitative purposes or with children at risk of neurodevelopmental disorders. This could have significant implications for early intervention as well as remedial intervention. For individuals with EF deficits, tailored coding interventions can provide targeted support for specific EF domains, such as attentional control, cognitive flexibility, and planning abilities. Stratified analysis can also be used to inform early intervention efforts for children at risk of neurodevelopmental disorders. By identifying EF profiles associated with increased response to CT intervention, clinicians and educators can implement preventive measures and provide targeted coding activities to mitigate the impact of EF deficits on academic achievement and psychosocial functioning.

The benefits of exposure to computational thinking are not limited to the enhancement of cognitive functions, but also to learning the coding skills, which are essential in today's digitally pervasive environment. In the contemporary digital landscape, early exposure to coding literacy emerges as a crucial educational target. Beyond merely mastering digital technologies, such proficiency lays the groundwork for children to accumulate experience that proves instrumental for their future high school years and career.

Firstly, akin to any other skill, coding literacy fosters a solid foundation upon which children can build progressively complex digital competencies. Starting at an early age allows them time to familiarize themselves with coding languages and concepts, facilitating a seamless transition to more advanced programming principles as they progress through their education.

Moreover, early coding literacy instills problem-solving abilities and fosters critical thinking skills that are transferable across various domains. Through coding, children learn to approach challenges analytically, break them down into manageable components, and develop systematic strategies for resolution. These cognitive abilities are not only beneficial in the realm of technology but also translate effectively to academic pursuits across disciplines, providing students with a competitive edge in their high school years.

Furthermore, in an increasingly digitized job market, proficiency in coding opens doors to career opportunities across diverse sectors. By nurturing coding literacy from an early age, children are better prepared to pursue fulfilling careers that leverage their technological expertise.

In conclusion, early coding literacy holds profound implications for children's educational trajectories and future career prospects. Beyond mastering digital technologies, it equips them with invaluable skills and experiences that prove advantageous in their high school years and beyond, shaping their academic achievements and professional aspirations.

REFERENCES

Akcaoglu, M., & Koehler, M. J. (2014). Cognitive outcomes from the Game-Design and Learning (GDL) after-school program. Computers & Education, 75, 72–81. https://doi.org/10.1016/j.compedu.2014.02.003

Altemeier, L. E., Abbott, R. D., & Berninger, V. W. (2008). Executive functions for reading and writing in typical literacy development and dyslexia. Journal of Clinical and Experimental Neuropsychology, 30(5), 588–606. https://doi.org/10.1080/13803390701562818

Anderson, P. (2002). Assessment and Development of Executive Function (EF) During Childhood. Child Neuropsychology, 8(2), 71–82. https://doi.org/10.1076/chin.8.2.71.8724

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. Computers in Human Behavior, 105, 105954. https://doi.org/10.1016/j.chb.2019.03.018

Arfé, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. Computers & Education, 148, 103807. https://doi.org/10.1016/j.compedu.2020.103807

Arfé, B., Vardanega, T., & Ronconi, L. (2020). The effects of coding on children's planning and inhibition skills. Computers & Education, 148, 103807. https://doi.org/10.1016/j.compedu.2020.103807

Arfé, B., Vardanega, T., Montuori, C., & Lavanga, M. (2019). Coding in Primary Grades Boosts Children's Executive Functions. Frontiers in Psychology, 10. https://www.frontiersin.org/article/10.3389/fpsyg.2019.02713

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. Robotics and Autonomous Systems, 75, 661–670. https://doi.org/10.1016/j.robot.2015.10.008

Aydoğan, Y., Ömeroğlu, E., Büyüköztürk, Ş., & Özyürek, A. (2012). Problem Çözme Becerileri Ölçeği Rehber Kitap. Ankara: Karaca Eğitim Yayın Dağıtım AŞ.

Bakala, E., Gerosa, A., Hourcade, J. P., & Tejera, G. (2021). Preschool children, robots, and computational thinking: A systematic review. International Journal of Child-Computer Interaction, 29, 100337. https://doi.org/10.1016/j.ijcci.2021.100337

Barkley, R. A. (2001). The Executive Functions and Self-Regulation: An Evolutionary Neuropsychological Perspective. Neuropsychology Review, 11(1), 1-29.

Barnett, S. M., & Ceci, S. J. (2002). When and where do we apply what we learn?: A taxonomy for far transfer. Psychological Bulletin, 128, 612–637. https://doi.org/10.1037/0033-2909.128.4.612

Barrouillet, P. (2015). Theories of cognitive development: From Piaget to today. Developmental Review, 38, 1–12. https://doi.org/10.1016/j.dr.2015.07.004

Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. Education and Information Technologies, 27(2), 2059–2082. https://doi.org/10.1007/s10639-021-10700-2

Bell, M. A., & Cuevas, K. (2016). Psychobiology of executive function in early development. In Executive function in preschool-age children: Integrating measurement, neurodevelopment, and translational research (pp. 157–179). American Psychological Association. https://doi.org/10.1037/14797-008

Belur, J., Tompson, L., Thornton, A., & Simon, M. (2021). Interrater Reliability in Systematic Review Methodology: Exploring Variation in Coder Decision-Making. Sociological Methods & Research, 50(2), 837–865. https://doi.org/10.1177/0049124118799372

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. Computers & Education, 72, 145–157. https://doi.org/10.1016/j.compedu.2013.10.020

Best, J. R., & Miller, P. H. (2010). A Developmental Perspective on Executive Function. Child Development, 81(6), 1641–1660. https://doi.org/10.1111/j.1467-8624.2010.01499.x

Bisiacchi, P., Cendron, M., Gugliotta, M., Tressoldi, P. E., & Vio, C. (2005). BVN 5-11—Batteria di valutazione neuropsicologica per l'età evolutiva. Erickson.

Borenstein, M., Hedges, L. V., Higgins, J. P. T., & Rothstein, H. R. (2009). Introduction to Meta-Analysis. https://doi.org/10.1002/9780470743386

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of Computational Thinking Skills through Unplugged Activities in Primary School. Proceedings of the 12th Workshop on Primary and Secondary Computing Education, 65–72. https://doi.org/10.1145/3137065.3137069

Brainin, E., Shamir, A., & Eden, S. (2022). Robot programming intervention for promoting spatial relations, mental rotation and visual memory of kindergarten children. Journal of Research on Technology in Education, 54(3), 345–358. https://doi.org/10.1080/15391523.2020.1858464

Brocki, K. C., & Bohlin, G. (2004). Executive functions in children aged 6 to 13: A dimensional and developmental study. Developmental Neuropsychology, 26(2), 571–593. https://doi.org/10.1207/s15326942dn2602_3

Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2008). Computer Aided Instruction as a Vehicle for Problem Solving: Scratch Programming Environment in the Middle Years Classroom. ASEE Annual Conference and Exposition, Conference Proceedings.

Brown, T. T., & Jernigan, T. L. (2012). Brain Development During the Preschool Years. Neuropsychology Review, 22(4), 313–333. https://doi.org/10.1007/s11065-012-9214-1

Cai, Z., & Fan, X. (2020). A Comparison of Fixed-Effects and Random-Effects Models for Multivariate Meta-Analysis Using an SEM Approach. Multivariate Behav. Res., 55(6), 839–854. https://doi.org/10.1080/00273171.2019.1689348

Çakir, R., Korkmaz, Ö., İdil, Ö., & Uğur Erdoğmuş, F. (2021). The effect of robotic coding education on preschoolers' problem solving and creative thinking skills. Thinking Skills and Creativity, 40, 100812. https://doi.org/10.1016/j.tsc.2021.100812

Cameron Ponitz, C. E., McClelland, M. M., Jewkes, A. M., Connor, C. M. D., Farris, C. L., & Morrison, F. J. (2008). Touch your toes! Developing a direct measure of behavioral regulation in early childhood. Early Childhood Research Quarterly, 23(2), 141–158. https://doi.org/10.1016/j.ecresq.2007.01.004

Campbell, C., & Walsh, C. (2017). Introducing the "new" digital literacy of coding in the early years. Practical Literacy, 22, 10–12.

Campbell, M. K., Elbourne, D. R., & Altman, D. G. (2004). CONSORT statement: Extension to cluster randomised trials. BMJ, 328(7441), 702–708. https://doi.org/10.1136/bmj.328.7441.702

Canbeldek, M., & Isikoglu, N. (2023). Exploring the effects of "productive children: Coding and robotics education program" in early childhood education. Education and Information Technologies, 28(3), 3359–3379. https://doi.org/10.1007/s10639-022-11315-x

Cao, Y., Huang, T., Huang, J., Xie, X., & Wang, Y. (2020). Effects and Moderators of Computer-Based Training on Children's Executive Functions: A Systematic Review and Meta-Analysis. Frontiers in Psychology, 11. https://www.frontiersin.org/articles/10.3389/fpsyg.2020.580329Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. Computers & Education, 109, 162–175. https://doi.org/10.1016/j.compedu.2017.03.001

Carlson, S. M. (2005). Developmentally sensitive measures of executive function in preschool children. Developmental Neuropsychology, 28(2), 595–616. https://doi.org/10.1207/s15326942dn2802_3

Casey, B. J., Trainor, R. J., Orendi, J. L., Schubert, A. B., Nystrom, L. E., Giedd, J. N., Castellanos, F. X., Haxby, J. V., Noll, D. C., Cohen, J. D., Forman, S. D., Dahl, R. E., & Rapoport, J. L. (1997). A Developmental Functional MRI Study of Prefrontal Activation during Performance of a Go-No-Go Task. Journal of Cognitive Neuroscience, 9(6), 835–847. https://doi.org/10.1162/jocn.1997.9.6.835

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. Computers & Education, 109, 162–175. https://doi.org/10.1016/j.compedu.2017.03.001

Cheung, M. W.-L. (2013). Multivariate Meta-Analysis as Structural Equation Models. Structural Equation Modeling: A Multidisciplinary Journal, 20(3), 429 454. https://doi.org/10.1080/10705511.2013.797827

Cheung, M. W.-L. (2019). A Guide to Conducting a Meta-Analysis with Non-Independent Effect Sizes. Neuropsychol. Rev., 29(4), 387–396. https://doi.org/10.1007/s11065-019-09415-6

Ching, Y.-H., & Hsu, Y.-C. (2023). Educational Robotics for Developing Computational Thinking in Young Learners: A Systematic Review. TechTrends. https://doi.org/10.1007/s11528-023-00841-1

Çiftci, S., & Bildiren, A. (2020). The effect of coding courses on the cognitive abilities and problem-solving skills of preschool children. Computer Science Education, 30(1), 3–21. https://doi.org/10.1080/08993408.2019.1696169

Çınar, M., & Tüzün, H. (2021). Comparison of object-oriented and robot programming activities: The effects of programming modality on student achievement, abstraction, problem solving, and motivation. Journal of Computer Assisted Learning, 37(2), 370–386. https://doi.org/10.1111/jcal.12495

Clark, C. A. C., Pritchard, V. E., & Woodward, L. J. (2010). Preschool executive functioning abilities predict early mathematics achievement. Developmental Psychology, 46(5), 1176–1191. https://doi.org/10.1037/a0019672

Clements, D. H., & Nastasi, B. K. (1999). Metacognition, Learning, and Educational Computer Environments. Information Technology in Childhood Education Annual, 1999(1), 3–36.

Colzato, L. S., van Wouwe, N. C., Lavender, T. J., & Hommel, B. (2006). Intelligence and cognitive flexibility: Fluid intelligence correlates with feature "unbinding" across perception and action. Psychonomic Bulletin & Review, 13(6), 1043–1048. https://doi.org/10.3758/BF03213923

Conklin, H. M., Luciana, M., Hooper, C. J., & Yarger, R. S. (2007). Working Memory Performance in Typically Developing Children and Adolescents: Behavioral Evidence of Protracted Frontal Lobe Development. Developmental Neuropsychology, 31(1), 103–128. https://doi.org/10.1080/87565640709336889

Critten, V., Hagon, H., & Messer, D. (2022). Can Pre-school Children Learn Programming and Coding Through Guided Play Activities? A Case Study in Computational Thinking. Early Childhood Education Journal, 50(6), 969–981. https://doi.org/10.1007/s10643-021-01236-8

Crocetti, E. (2016). Systematic Reviews With Meta-Analysis: Why, When, and How? Emerging Adulthood, 4(1), 3–18. https://doi.org/10.1177/2167696815617076

Davidson, M. C., Amso, D., Anderson, L. C., & Diamond, A. (2006). Development of cognitive control and executive functions from 4 to 13 years: Evidence from manipulations of memory, inhibition, and task switching. Neuropsychologia, 44(11), 2037–2078. https://doi.org/10.1016/j.neuropsychologia.2006.02.006

De Luca, C. R., Wood, S. J., Anderson, V., Buchanan, J.-A., Proffitt, T. M., Mahony, K., & Pantelis, C. (2003). Normative Data From the Cantab. I: Development of Executive Function Over the Lifespan. Journal of Clinical and Experimental Neuropsychology, 25(2), 242–254. https://doi.org/10.1076/jcen.25.2.242.13639

Demir, Ü. (2021). The Effect of Unplugged Coding Education for Special Education Students on Problem-Solving Skills. International Journal of Computer Science Education in Schools, 4(3), Article 3. https://doi.org/10.21585/ijcses.v4i3.95

Di Lieto, M. C., Inguaggiato, E., Castro, E., Cecchi, F., Cioni, G., Dell'Omo, M., Laschi, C., Pecini, C., Santerini, G., Sgandurra, G., & Dario, P. (2017). Educational Robotics intervention on Executive Functions in preschool children: A pilot study. Computers in Human Behavior, 71, 16–23. https://doi.org/10.1016/j.chb.2017.01.018

Di Lieto, M. C., Castro, E., Pecini, C., Inguaggiato, E., Cecchi, F., Dario, P., Cioni, G., & Sgandurra, G. (2020b). Improving Executive Functions at School in Children With Special Needs by Educational Robotics. Frontiers in Psychology, 10. https://www.frontiersin.org/articles/10.3389/fpsyg.2019.02813

Di Lieto, M. C., Pecini, C., Castro, E., Inguaggiato, E., Cecchi, F., Dario, P., Cioni, G., & Sgandurra, G. (2020a). Empowering Executive Functions in 5- and 6-Year-Old Typically Developing Children Through Educational Robotics: An RCT Study. Frontiers in Psychology, 10. https://www.frontiersin.org/article/10.3389/fpsyg.2019.03084

Diamond, A. (2006). The Early Development of Executive Functions. In E. Bialystok & F. I. M. Craik (Eds.), Lifespan Cognition: Mechanisms of Change (p. 0). Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195169539.003.0006

Diamond, A. (2013). Executive Functions. Annual Review of Psychology, 64(1), 135–168. https://doi.org/10.1146/annurev-psych-113011-143750

Diamond, A., & Ling, D. S. (2016). Conclusions about interventions, programs, and approaches for improving executive functions that appear justified and those that, despite much hype, do not. Developmental Cognitive Neuroscience, 18, 34–48. https://doi.org/10.1016/j.dcn.2015.11.005

Egger, M., Smith, G. D., Schneider, M., & Minder, C. (1997). Bias in meta-analysis detected by a simple, graphical test. BMJ, 315(7109), 629–634. https://doi.org/10.1136/bmj.315.7109.629

Erol, O., & Çırak, N. S. (2022). The effect of a programming tool scratch on the problem-solving skills of middle school students. Education and Information Technologies, 27(3), 4065–4086. https://doi.org/10.1007/s10639-021-10776-w

Fancello, G. S., Vio, C., & Cianchetti, C. (2013). ToL, Torre di Londra. Test di Valutazione delle Funzioni Esecutive (Pianificazione e Problem Solving). Erickson. https://www.erickson.it/it/test-tol-torre-di-londra-test-di-valutazione-delle-funzioni-esecutive

Ferrer-Mico, T., Prats-Fernàndez, M. À., & Redo-Sanchez, A. (2012). Impact of Scratch Programming on Students' Understanding of Their Own Learning Process. Procedia - Social and Behavioral Sciences, 46, 1219–1223. https://doi.org/10.1016/j.sbspro.2012.05.278

Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. Computers & Education, 63, 87–97. https://doi.org/10.1016/j.compedu.2012.11.016

Feurzeig, W., & Papert, S. A. (2011). Programming-languages as a conceptual framework for teaching mathematics. Interactive Learning Environments, 19(5), 487–501. https://doi.org/10.1080/10494820903520040

Finke, S., Kemény, F., Sommer, M., Krnjic, V., Arendasy, M., Slany, W., & Landerl, K. (2022). Unravelling the numerical and spatial underpinnings of computational thinking: A pre-registered replication study. Computer Science Education, 32(3), 313–334. https://doi.org/10.1080/08993408.2022.2079864

Flórez, B. F., Casallas, R., Hernández, 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

Frensch, P. A., & Funke, J. (1995). Complex Problem Solving. The European Perspective. Psychology Press.

Fu, C., Hao, X., Shi, D., Wang, L., & Geng, F. (2023). Effect of coding learning on the computational thinking of young Chinese children: Based on the three-dimensional framework. Education and Information Technologies. https://doi.org/10.1007/s10639-023-11807-4

Furuya-Kanamori, L., Xu, C., Lin, L., Doan, T., Chu, H., Thalib, L., & Doi, S. A. R. (2020). P value–driven methods were underpowered to detect publication bias: Analysis of Cochrane review meta-analyses. Journal of Clinical Epidemiology, 118, 86–92. https://doi.org/10.1016/j.jclinepi.2019.11.011

Ganesan, K., & Steinbeis, N. (2022). Development and plasticity of executive functions: A value-based account. Current Opinion in Psychology, 44, 215–219. https://doi.org/10.1016/j.copsyc.2021.09.012

Gardner, M. F., Brown, G. T., Rodger, S., Davis, A., Klein, S., Sollereder, P., Gierl, M., Ambacher McFall, S., Deitz, J. C., & Crowe, T. K. (1996). TVPS-R: Test of visual-perceptual skills (non-motor)--revised : manual. Psychological and Educational Publications.

Garon, N., Bryson, S. E., & Smith, I. M. (2008). Executive function in preschoolers: A review using an integrative framework. Psychological Bulletin, 134, 31–60. https://doi.org/10.1037/0033-2909.134.1.31

Gathercole, S. E., Pickering, S. J., Ambridge, B., & Wearing, H. (2004). The Structure of Working Memory From 4 to 15 Years of Age. Developmental Psychology, 40, 177–190. https://doi.org/10.1037/0012-1649.40.2.177

Gerosa, A., Koleszar, V., Tejera, G., Gómez-Sena, L., & Carboni, A. (2021). Cognitive abilities and computational thinking at age 5: Evidence for associations to sequencing and symbolic number comparison. Computers and Education Open, 2, 100043. https://doi.org/10.1016/j.caeo.2021.100043

Gioia, G. A., Isquith, P. K., Guy, S. C., & Kenworthy, L. (2000). TEST REVIEW Behavior Rating Inventory of Executive Function. Child Neuropsychology, 6(3), 235–238. https://doi.org/10.1076/chin.6.3.235.3152

Goldin-Meadow, S. (2015). From action to abstraction: Gesture as a mechanism of change. Developmental Review, 38, 167–184. https://doi.org/10.1016/j.dr.2015.07.007

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

Gugliotta, M., Bisiacchi, P., Cendron, M., Tressoldi, P. E., & Vio, C. (2009). BVN 12-18: Batteria di valutazione neuropsicologia per l'adolescenza. Erickson.

Hedges, L. V. (1984). Estimation of Effect Size under Nonrandom Sampling: The Effects of Censoring Studies Yielding Statistically Insignificant Mean Differences. Journal of Educational Statistics, 9(1), 61–85. https://doi.org/10.2307/1164832

Heppner, P. P. (1988). The problem-solving inventory: Manual. Consulting Psychologists Press.

Hooper, C. J., Luciana, M., Conklin, H. M., & Yarger, R. S. (2004). Adolescents' performance on the Iowa Gambling Task: Implications for the development of decision making and ventromedial prefrontal cortex. Developmental Psychology, 40(6), 1148–1158. https://doi.org/10.1037/0012-1649.40.6.1148

Huepe, D., Roca, M., Salas, N., Canales-Johnson, A., Rivera-Rei, Á. A., Zamorano, L., Concepción, A., Manes, F., & Ibañez, A. (2011). Fluid Intelligence and Psychosocial Outcome: From Logical Problem Solving to Social Adaptation. PLOS ONE, 6(9), e24858. https://doi.org/10.1371/journal.pone.0024858

Huizinga, M., Dolan, C. V., & van der Molen, M. W. (2006). Age-related change in executive function: Developmental trends and a latent variable analysis. Neuropsychologia, 44(11), 2017–2036. https://doi.org/10.1016/j.neuropsychologia.2006.01.010

Jacob, R., & Parkinson, J. (2015). The potential for school-based interventions that target executive function to improve academic achievement: A review. Review of Educational Research, 85, 512–552. https://doi.org/10.3102/0034654314561338

Jiang, S., & Wong, G. K. W. (2022). Exploring age and gender differences of computational thinkers in primary school: A developmental perspective. Journal of Computer Assisted Learning, 38(1), 60–75. https://doi.org/10.1111/jcal.12591

Jin, Z.-C., Zhou, X.-H., & He, J. (2015). Statistical methods for dealing with publication bias in meta-analysis. Statistics in Medicine, 34(2), 343–360. https://doi.org/10.1002/sim.6342

Johnson, E. S., & Meade, A. C. (1987). Developmental Patterns of Spatial Ability: An Early Sex Difference. Child Development, 58(3), 725–740. https://doi.org/10.2307/1130210

Johnston, M. V. (2009). Plasticity in the developing brain: Implications for rehabilitation. Developmental Disabilities Research Reviews, 15(2), 94–101. https://doi.org/10.1002/ddrr.64

Jonkman, L. M., Lansbergen, M., & Stauder, J. E. A. (2003). Developmental differences in behavioral and event-related brain responses associated with response preparation and inhibition in a go/nogo task. Psychophysiology, 40(5), 752–761. https://doi.org/10.1111/1469-8986.00075

Jun, S., Han, S., & Kim, S. (2017). Effect of design-based learning on improving computational thinking. Behaviour & Information Technology, 36(1), 43–53. https://doi.org/10.1080/0144929X.2016.1188415

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. Computers in Human Behavior, 52, 200–210. https://doi.org/10.1016/j.chb.2015.05.047
Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. Computers in Human Behavior, 52, 200–210. https://doi.org/10.1016/j.chb.2015.05.047
Kassai, R., Futo, J., Demetrovics, Z., & Takacs, Z. K. (2019). A meta-analysis of the experimental evidence on the near- and far-transfer effects among children's executive function skills. Psychological Bulletin, 145(2), 165–188. https://doi.org/10.1037/bul0000180

Kazakoff, E. R., & Bers, M. U. (2014). Put Your Robot in, Put Your Robot out: Sequencing through Programming Robots in Early Childhood. Journal of Educational Computing Research, 50(4), 553–573. https://doi.org/10.2190/EC.50.4.f

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood. Early Childhood Education Journal, 41(4), 245–255. https://doi.org/10.1007/s10643-012-0554-5

Keen, R. (2011). The Development of Problem Solving in Young Children: A Critical Cognitive Skill. Annual Review of Psychology, 62(1), 1–21. https://doi.org/10.1146/annurev.psych.031809.130730

Keren, G., & Fridin, M. (2014). Kindergarten Social Assistive Robot (KindSAR) for children's geometric thinking and metacognitive development in preschool education: A pilot study. Computers in Human Behavior, 35, 400–412. https://doi.org/10.1016/j.chb.2014.03.009

Klenberg, L., Korkman, M., & Lahti-Nuuttila, P. (2001). Differential Development of Attention and Executive Functions in 3- to 12-Year-Old Finnish Children. Developmental Neuropsychology, 20(1), 407–428. https://doi.org/10.1207/S15326942DN2001_6

Klingberg, T. (2010). Training and plasticity of working memory. Trends in Cognitive Sciences, 14(7), 317–324. https://doi.org/10.1016/j.tics.2010.05.002

Kolb, B., Harker, A., & Gibb, R. (2017). Principles of plasticity in the developing brain. Developmental Medicine & Child Neurology, 59(12), 1218–1223. https://doi.org/10.1111/dmcn.13546

Korkman, M., Kirk, U., & Kemp, S. (2007). NEPSY-II: A developmental neurospychological assessment. The Psychological Corporation. https://psycnet.apa.org/doiLanding?doi=10.1037%2Ft15125-000

Kyza, E. A., Georgiou, Y., Agesilaou, A., & Souropetsis, M. (2022). A Cross-Sectional Study Investigating Primary School Children's Coding Practices and Computational Thinking Using ScratchJr. Journal of Educational Computing Research, 60(1), 220–257. https://doi.org/10.1177/07356331211027387

La Paglia, F., La Cascia, C., Francomano, M. M., & La Barbera, D. (2017). Educational Robotics to Improve Mathematical and Metacognitive Skills. Annual Review of CyberTherapy and Telemedicine, 15.

Lai, A. F., & Yang, S. M. (2011). The learning effect of visualized programming learning on 6th graders' problem solving and logical reasoning abilities. 2011 International Conference on Electrical and Control Engineering, 6940–6944. https://doi.org/10.1109/ICECENG.2011.6056908

Lee, J., & Junoh, J. (2019). Implementing Unplugged Coding Activities in Early Childhood Classrooms. Early Childhood Education Journal, 47(6), 709–716. https://doi.org/10.1007/s10643-019-00967-z

Lee, K. T. H., Sullivan, A., & Bers, M. U. (2013). Collaboration by Design: Using Robotics to Foster Social Interaction in Kindergarten. Computers in the Schools, 30(3), 271–281. https://doi.org/10.1080/07380569.2013.805676

Lehto, J. E., Juujärvi, P., Kooistra, L., & Pulkkinen, L. (2003). Dimensions of executive functioning: evidence from children. British Journal of Developmental Psychology, 21 (1), 59–80. https://doi.org/10.1348/026151003321164627

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using Robotics and Game Design to Enhance Children's Self-Efficacy, STEM Attitudes, and Computational Thinking Skills. Journal of Science Education and Technology, 25(6), 860–876. https://doi.org/10.1007/s10956-016-9628-2

Li, F., Wang, X., He, X., Cheng, L., & Wang, Y. (2022). The effectiveness of unplugged activities and programming exercises in computational thinking education: A Meta-analysis. Education and Information Technologies, 27(6), 7993–8013. https://doi.org/10.1007/s10639-022-10915-x

Liao, Y. (2000). A Meta-analysis of Computer Programming on Cognitive Outcomes: An Updated Synthesis. Paper presented at the Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, Montreal, Québec, Canada. Retrieved from https://www.learntechlib.org/primary/p/16132/

Liao, Y. K. C., & Bright, G. W. (1991). Effects of Computer Programming on Cognitive Outcomes: A Meta-Analysis. Journal of Educational Computing Research, 7(3), 251–268. https://doi.org/10.2190/E53G-HH8K-AJRR-K69M

Liu, Y., Ma, Z., & Qian, Y. (2019). Developing Chinese Elementary School Students' Computational Thinking: A Convergent Cognition Perspective. Proceedings of the ACM Conference on Global Computing Education, 238. https://doi.org/10.1145/3300115.3312514

Luciana, M., & Nelson, C. A. (2002). Assessment of Neuropsychological Function Through Use of the Cambridge Neuropsychological Testing Automated Battery: Performance in 4- to 12-Year-Old Children. Developmental Neuropsychology, 22(3), 595–624. https://doi.org/10.1207/S15326942DN2203_3

Luciana, M., Collins, P. F., Olson, E. A., & Schissel, A. M. (2009). Tower of London Performance in Healthy Adolescents: The Development of Planning Skills and Associations With Self-Reported Inattention and Impulsivity. Developmental Neuropsychology, 34(4), 461–475. https://doi.org/10.1080/87565640902964540

Luna, B., Garver, K. E., Urban, T. A., Lazar, N. A., & Sweeney, J. A. (2004). Maturation of cognitive processes from late childhood to adulthood. Child Development, 75(5), 1357–1372. https://doi.org/10.1111/j.1467-8624.2004.00745.x

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

Macdonald, J. A., Beauchamp, M. H., Crigan, J. A., & Anderson, P. J. (2014). Age-related differences in inhibitory control in the early school years. Child Neuropsychology, 20(5), 509–526. https://doi.org/10.1080/09297049.2013.822060

Mägi, K., Männamaa, M., & Kikas, E. (2016). Profiles of self-regulation in elementary grades: Relations to math and reading skills. Learning and Individual Differences, 51, 37–48. https://doi.org/10.1016/j.lindif.2016.08.028

Mammarella, I. C., Toso, C., Pazzaglia, F., & Cornoldi, C. (2008). BVS - Corsi—Batteria per la valutazione della memoria visiva e spaziale. Erickson.

Marinus, E., Powell, Z., Thornton, R., McArthur, G., & Crain, S. (2018). Unravelling the Cognition of Coding in 3-to-6-year Old: The development of an assessment tool and the relation between coding ability and cognitive compiling of syntax in natural language. Proceedings of the 2018 ACM Conference on International Computing Education Research, 133–141. https://doi.org/10.1145/3230977.3230984

Marshall, P. J., & Drew, A. R. (2014). What makes Simon Says so difficult for young children? Journal of Experimental Child Psychology, 126, 112–119. https://doi.org/10.1016/j.jecp.2014.03.011

Marzocchi, G. M., Re, A. M., & Cornoldi, C. (2010). BIA. Batteria italiana per l'ADHD. Erickson.

Mathes, T., & Kuss, O. (2018). A comparison of methods for meta-analysis of a small number of studies with binary outcomes. Research Synthesis Methods, 9(3), 366–381. https://doi.org/10.1002/jrsm.1296

Mathes, T., & Kuss, O. (2018). A comparison of methods for meta-analysis of a small number of studies with binary outcomes. Research Synthesis Methods, 9(3), 366–381. https://doi.org/10.1002/jrsm.1296

McGuckian, T. B., Wilson, P. H., Johnston, R. D., Rahimi-Golkhandan, S., Piek, J., Green, D., Rogers, J. M., Maruff, P., Steenbergen, B., & Ruddock, S. (2023). Development of complex executive function over childhood: Longitudinal growth curve modeling of performance on the Groton Maze Learning Task. Child Development, 94(3), 648–658. https://doi.org/10.1111/cdev.13888

Meixner, J. M., Warner, G. J., Lensing, N., Schiefele, U., & Elsner, B. (2019). The relation between executive functions and reading comprehension in primary-school students: A cross-lagged-panel analysis. Early Childhood Research Quarterly, 46, 62–74. https://doi.org/10.1016/j.ecresq.2018.04.010

Melby-Lervåg, M., & Hulme, C. (2013). Is working memory training effective? A meta-analytic review. Developmental Psychology, 49(2), 270–291. https://doi.org/10.1037/a0028228

Methley, A. M., Campbell, S., Chew-Graham, C., McNally, R., & Cheraghi-Sohi, S. (2014). PICO, PICOS and SPIDER: A comparison study of specificity and sensitivity in three search tools for qualitative systematic reviews. BMC Health Services Research, 14(1), 579. https://doi.org/10.1186/s12913-014-0579-0

Miller, M. R., Giesbrecht, G. F., Müller, U., McInerney, R. J., & Kerns, K. A. (2012). A Latent Variable Approach to Determining the Structure of Executive Function in Preschool Children. Journal of Cognition and Development, 13(3), 395–423. https://doi.org/10.1080/15248372.2011.585478

Miyake, A., Friedman, N. P., Emerson, M. J., Witzki, A. H., Howerter, A., & Wager, T. D. (2000). The unity and diversity of executive functions and their contributions to complex "Frontal Lobe" tasks: A latent variable analysis. Cognitive Psychology, 41(1), 49–100. https://doi.org/10.1006/cogp.1999.0734

Monette, S., Bigras, M., & Guay, M. C. (2011). The role of the executive functions in school achievement at the end of Grade 1. Journal of Experimental Child Psychology, 109(2), 158–173. https://doi.org/10.1016/j.jecp.2011.01.008

Montuori, C., Ronconi, L., Vardanega, T., & Arfé, B. (2022). Exploring Gender Differences in Coding at the Beginning of Primary School. Frontiers in Psychology, 13. https://www.frontiersin.org/articles/10.3389/fpsyg.2022.887280

Montuori, C., Pozzan, G., Padova, C., Vardanega, T., Arfé, B. (2023). Combined Unplugged and Educational Robotics training to promote Computational Thinking and Cognitive Abilities in Preschoolers. Education Sciences, 13, 858. https://doi.org/10.3390/educsci13090858 Published under a CC BY LICENSE

Montuori, C., Gambarota, F., Altoé, G., & Arfé, B. (2024). The cognitive effects of computational thinking: A systematic review and meta-analytic study. Computers & Education, 210, 104961. https://doi.org/10.1016/j.compedu.2023.104961 Published under a CC BY NC ND LICENSE

Moreno-León, J., Robles, G., & Román-González, M. (2016). Comparing computational thinking development assessment scores with software complexity metrics. 2016 IEEE Global Engineering Education Conference (EDUCON), 1040–1045. https://doi.org/10.1109/EDUCON.2016.7474681

Morris, S. B. (2008). Estimating Effect Sizes From Pretest-Posttest-Control Group Designs. Organizational Research Methods, 11(2), 364–386. https://doi.org/10.1177/1094428106291059

Nam, D., Kim, Y., & Lee, T. (2010). The Effects of Scaffolding-Based Courseware for The Scratch Programming Learning on Student Problem Solving Skill. Proceedings of the 18th International Conference on Computers in Education (ICCE).

Nam, K. W., Kim, H. J., & Lee, S. (2019). Connecting Plans to Action: The Effects of a Card-Coded Robotics Curriculum and Activities on Korean Kindergartners. The Asia-Pacific Education Researcher, 28(5), 387–397. https://doi.org/10.1007/s40299-019-00438-4

Nardelli, E. (2019). Do we really need computational thinking? Communications of the ACM, 62(2), 32–35. https://doi.org/10.1145/3231587

Nouwens, S., Groen, M. A., Kleemans, T., & Verhoeven, L. (2021). How executive functions contribute to reading comprehension. British Journal of Educational Psychology, 91(1), e12355. https://doi.org/10.1111/bjep.12355

Novack, M. A., Congdon, E. L., Hemani-Lopez, N., & Goldin-Meadow, S. (2014). From Action to Abstraction: Using the Hands to Learn Math. Psychological Science, 25(4), 903–910. https://doi.org/10.1177/0956797613518351

OECD (2017), PISA 2015 Results (Volume V): Collaborative Problem Solving, PISA, OECD Publishing, Paris. http://dx.doi.org/10.1787/9789264285521-en

OECD. (2003). PISA 2003 Assessment Framework: Mathematics, Reading, Science and Problem Solving Knowledge and Skills. https://www.oecd.org/education/school/programmeforinternationalstudentassessmentpisa/33694881.pdf

Oğuz, V., & Köksal Akyol, A. (2015). Problem-Solving Skills Scale (PSSS) Validity and Reliability. Cukurova University Faculty of Educational Journal, 44(1). https://avesis.ankara.edu.tr/yayin/ea04cac4-92de-4348-8408-4797177d4dc8/problem-solving-skills-scale-psss-validity-and-reliability

Oluk, A., & Saltan, F. (2015). Effects of Using the Scratch Program in 6th Grade Information Technologies Courses on Algorithm Development and Problem Solving Skills. Participatory Educational Research (PER), 2(5), 10-20. https://doi.org/10.17275/per.15.spi.2.2

Özcan, M. Ş., Çetinkaya, E., Göksun, T., & Kisbu-Sakarya, Y. (2021). Does learning to code influence cognitive skills of elementary school children? Findings from a randomized experiment. The British Journal of Educational Psychology, 91(4), 1434–1455. https://doi.org/10.1111/bjep.12429

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., … Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. Systematic Reviews, 10(1), 89. https://doi.org/10.1186/s13643-021-01626-4

Papavlasopoulou, S., Sharma, K., Giannakos, M., & Jaccheri, L. (2017). Using Eye-Tracking to Unveil Differences Between Kids and Teens in Coding Activities. Proceedings of the 2017 Conference on Interaction Design and Children, 171–181. https://doi.org/10.1145/3078072.3079740

Pardamean, B., Evelin, E., & Honni, H. (2011). The effect of logo programming language for creativity and problem solving. Proceedings of the 10th WSEAS International Conference on E-Activities, 151–156.

Passolunghi, M. C., & Costa, H. M. (2016). Working memory and early numeracy training in preschool children. Child Neuropsychology, 22(1), 81–98. https://doi.org/10.1080/09297049.2014.971726

Pennington, B. F., & Ozonoff, S. (1996). Executive Functions and Developmental Psychopathology. Journal of Child Psychology and Psychiatry, 37(1), 51–87. https://doi.org/10.1111/j.1469-7610.1996.tb01380.x

Perkins, D. N., & Salomon, G. (1992). Transfer of learning. In International encyclopedia of education (Vol. 2, pp. 6452–6457). Pergamon Press.

Ping, R., & Goldin-Meadow, S. (2008). Hands in the air. Developmental Psychology, 44(5), 1277–1287. https://doi.org/10.1037/0012-1649.44.5.1277

Polat, E., Hopcan, S., Kucuk, S., & Sisman, B. (2021). A comprehensive assessment of secondary school students' computational thinking skills. British Journal of Educational Technology, 52(5), 1965–1980. Scopus. https://doi.org/10.1111/bjet.13092

Poutanen, M., Berg, S., Kangas, T., Peltomaa, K., Lahti-Nuuttila, P., & Hokkanen, L. (2016). Before and after entering school: The development of attention and executive functions from 6 to 8 years in Finnish children. Scandinavian Journal of Psychology, 57(1), 1–11. https://doi.org/10.1111/sjop.12264

Pozuelos, J. P., Combita, L. M., Abundis, A., Paz-Alonso, P. M., Conejero, Á., Guerra, S., & Rueda, M. R. (2019). Metacognitive scaffolding boosts cognitive and neural benefits following executive attention training in children. Developmental Science, 22(2), e12756. https://doi.org/10.1111/desc.12756

R Core Team. (2022). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing.

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. Computers & Education, 169, 104222. https://doi.org/10.1016/j.compedu.2021.104222

Rijke, W. J., Bollen, L., Eysink, T. H. S., & Tolboom, J. L. J. (2018). Computational Thinking in Primary School: An Examination of Abstraction and Decomposition in Different Age Groups. Informatics in Education, 17(1), 77–92. https://doi.org/10.15388/infedu.2018.05

Robledo-Castro, C., Castillo-Ossa, L. F., & Hederich-Martínez, C. (2023). Effects of a computational thinking intervention program on executive functions in children aged 10 to 11. International Journal of Child-Computer Interaction, 35, 100563. https://doi.org/10.1016/j.ijcci.2022.100563

Robledo-Castro, C., Hederich-Martínez, C., & Castillo-Ossa, L. F. (2023). Cognitive stimulation of executive functions through computational thinking. Journal of Experimental Child Psychology, 235, 105738. https://doi.org/10.1016/j.jecp.2023.105738

Román-González, M., Pérez-González, J.-C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. Computers in Human Behavior, 72, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Román-González, M., Pérez-González, J.-C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. International Journal of Child-Computer Interaction, 18, 47–58. https://doi.org/10.1016/j.ijcci.2018.06.004

Roussou, E., & Rangoussi, M. (2020). On the Use of Robotics for the Development of Computational Thinking in Kindergarten: Educational Intervention and Evaluation. In M. Merdan, W. Lepuschitz, G. Koppensteiner, R. Balogh, & D. Obdržálek (Eds.), Robotics in Education (pp. 31–44). Springer International Publishing. https://doi.org/10.1007/978-3-030-26945-6_3

Sáez-López, J.-M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. Computers & Education, 97, 129–141. https://doi.org/10.1016/j.compedu.2016.03.003

Salas, N., & Silvente, S. (2020). The role of executive functions and transcription skills in writing: A cross-sectional study across 7 years of schooling. Reading and Writing, 33(4), 877–905. https://doi.org/10.1007/s11145-019-09979-y

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. Journal of Educational Psychology, 111, 764–792. https://doi.org/10.1037/edu0000314

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2020). A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. Computers in Human Behavior, 109, 106349. https://doi.org/10.1016/j.chb.2020.106349

Schmitt, S. A., McClelland, M. M., Tominey, S. L., & Acock, A. C. (2015). Strengthening school readiness for Head Start children: Evaluation of a self-regulation intervention. Early Childhood Research Quarterly, 30, 20–31. https://doi.org/10.1016/j.ecresq.2014.08.001

Schunk, D. H. (2012). Learning theories: An educational perspective (6th ed.). Pearson Education.

Scionti, N., Cavallero, M., Zogmaister, C., & Marzocchi, G. M. (2020). Is Cognitive Training Effective for Improving Executive Functions in Preschoolers? A Systematic Review and Meta-Analysis. Frontiers in Psychology, 10. https://www.frontiersin.org/articles/10.3389/fpsyg.2019.02812

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

Simon, H. A. (1975). The functional equivalence of problem-solving skills. Cognitive Psychology, 7(2), 268–288. https://doi.org/10.1016/0010-0285(75)90012-2

Socratous, C., & Ioannou, A. (2021). Structured or unstructured educational robotics curriculum? A study of debugging in block-based programming. Educational Technology Research and Development, 69(6), 3081–3100. https://doi.org/10.1007/s11423-021-10056-x

Spiegel, J. A., Goodrich, J. M., Morris, B. M., Osborne, C. M., & Lonigan, C. J. (2021). Relations between executive functions and academic outcomes in elementary school children: A meta-analysis. Psychological Bulletin, 147(4), 329–351. https://doi.org/10.1037/bul0000322

Spiegel, J. A., Goodrich, J. M., Morris, B. M., Osborne, C. M., & Lonigan, C. J. (2021). Relations between executive functions and academic outcomes in elementary school children: A meta-analysis. Psychological Bulletin, 147(4), 329–351. https://doi.org/10.1037/bul0000322

Steegen, S., Tuerlinckx, F., Gelman, A., & Vanpaemel, W. (2016). Increasing Transparency Through a Multiverse Analysis. Perspectives on Psychological Science, 11(5), 702–712. https://doi.org/10.1177/1745691616658637

Sterne, J. A. C., Sutton, A. J., Ioannidis, J. P. A., Terrin, N., Jones, D. R., Lau, J., Carpenter, J., Rücker, G., Harbord, R. M., Schmid, C. H., Tetzlaff, J., Deeks, J. J., Peters, J., Macaskill, P., Schwarzer, G., Duval, S., Altman, D. G., Moher, D., & Higgins, J. P. T. (2011). Recommendations for examining and interpreting funnel plot asymmetry in meta-analyses of randomised controlled trials. BMJ, 343, d4002. https://doi.org/10.1136/bmj.d4002

Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. Educational Technology Research and Development, 67(3), 541–575. https://doi.org/10.1007/s11423-018-9622-x

Sun, L., Hu, L., & Zhou, D. (2021). Improving 7th-graders' computational thinking skills through unplugged programming activities: A study on the influence of multiple factors. Thinking Skills and Creativity, 42, 100926. https://doi.org/10.1016/j.tsc.2021.100926

Thayer, J. F., & Lane, R. D. (2000). A model of neurovisceral integration in emotion regulation and dysregulation. Journal of Affective Disorders, 61(3), 201–216. https://doi.org/10.1016/S0165-0327(00)00338-4

Thurstone, L. L., & Thurstone, T. G. (1965). Primary Mental Ability. SRA (Science Research Associates, Inc.).

Traverso, L., Viterbori, P., & Usai, M. C. (2015). Improving executive function in childhood: Evaluation of a training intervention for 5-year-old children. Frontiers in Psychology, 6. https://www.frontiersin.org/articles/10.3389/fpsyg.2015.00525

Tsai, C. Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy. Computers in Human Behavior, 95, 224–232. https://doi.org/10.1016/j.chb.2018.11.038

Tsarava, K., Leifheit, L., Ninaus, M., Román-González, M., Butz, M. V., Golle, J., Trautwein, U., & Moeller, K. (2019). Cognitive Correlates of Computational Thinking: Evaluation of a Blended Unplugged/Plugged-In Course. Proceedings of the 14th Workshop in Primary and Secondary Computing Education, 1–9. https://doi.org/10.1145/3361721.3361729

Turesky, T. K., Vanderauwera, J., & Gaab, N. (2021). Imaging the rapidly developing brain: Current challenges for MRI studies in the first five years of life. Developmental Cognitive Neuroscience, 47, 100893. https://doi.org/10.1016/j.dcn.2020.100893

Unterrainer, J. M., & Owen, A. M. (2006). Planning and problem solving: From neuropsychology to functional neuroimaging. Journal of Physiology-Paris, 99(4), 308–317. https://doi.org/10.1016/j.jphysparis.2006.03.014

Unuakhalu, M. F. (2008). Enhancing Problem-Solving Capabilities Using Object-Oriented Programming Language. Journal of Educational Technology Systems, 37(2), 121–137. https://doi.org/10.2190/ET.37.2.b

Usai, M. C., Viterbori, P., Traverso, L., & De Franchis, V. (2014). Latent structure of executive function in five- and six-year-old children: A longitudinal study. European Journal of Developmental Psychology, 11(4), 447–462. https://doi.org/10.1080/17405629.2013.840578

Uttal, D. H., Miller, D. I., & Newcombe, N. S. (2013). Exploring and Enhancing Spatial Thinking: Links to Achievement in Science, Technology, Engineering, and Mathematics? Current Directions in Psychological Science, 22(5), 367–373. https://doi.org/10.1177/0963721413484756

Van den Noortgate, W., López-López, J. A., Marín-Martínez, F., & Sánchez-Meca, J. (2015). Meta-analysis of multiple outcomes: A multilevel approach. Behav. Res. Methods, 47(4), 1274–1294. https://doi.org/10.3758/s13428-014-0527-2

Vandenbroucke, L., Verschueren, K., & Baeyens, D. (2017). The development of executive functioning across the transition to first grade and its predictive value for academic achievement. Learning and Instruction, 49, 103–112. https://doi.org/10.1016/j.learninstruc.2016.12.008

Veroniki, A. A., Jackson, D., Viechtbauer, W., Bender, R., Bowden, J., Knapp, G., Kuss, O., Higgins, J. P. T., Langan, D., & Salanti, G. (2016). Methods to estimate the between-study variance and its uncertainty in meta-analysis. Res. Synth. Methods, 7(1), 55–79. https://doi.org/10.1002/jrsm.1164

Veroniki, A. A., Jackson, D., Viechtbauer, W., Bender, R., Bowden, J., Knapp, G., Kuss, O., Higgins, J. P. T., Langan, D., & Salanti, G. (2016). Methods to estimate the between-study variance and its uncertainty in meta-analysis. Res. Synth. Methods, 7(1), 55–79. https://doi.org/10.1002/jrsm.1164

Viechtbauer, W. (2010). Conducting meta-analyses in R with the metafor package. In Journal of Statistical Software (Vol. 36, Issue 3, pp. 1–48). https://doi.org/10.18637/jss.v036.i03

Welsh, J. A., Nix, R. L., Blair, C., Bierman, K. L., & Nelson, K. E. (2010). The development of cognitive skills and gains in academic school readiness for children from low-income families. Journal of Educational Psychology, 102(1), 43–53. https://doi.org/10.1037/a0016738

Welsh, M. C., & Pennington, B. F. (1988). Assessing frontal lobe functioning in children: Views from developmental psychology. Developmental Neuropsychology, 4, 199–230. https://doi.org/10.1080/87565648809540405

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215

Yang, W., Ng, D. T. K., & Gao, H. (2022). Robot programming versus block play in early childhood education: Effects on computational thinking, sequencing ability, and self-regulation. British Journal of Educational Technology, 53(6), 1817–1841. https://doi.org/10.1111/bjet.13215

Yaşar, O. (2018). A new perspective on computational thinking. Communications of the ACM, 61(7), 33–39. https://doi.org/10.1145/3214354

Zelazo, P. D., Carter, A., Reznick, J. S., & Frye, D. (1997). Early development of executive function: A problem-solving framework. Review of General Psychology, 1(2), 198–226. https://doi.org/10.1037/1089-2680.1.2.198

Zelazo, P. D., Müller, U., Frye, D., Marcovitch, S., Argitis, G., Boseovski, J., Chiang, J. K., Hongwanishkul, D., Schuster, B. V., & Sutherland, A. (2003). The development of executive function in early childhood. Monographs of the Society for Research in Child Development, 68(3), vii–137. https://doi.org/10.1111/j.0037-976x.2003.00260.x

Zhan, Z., He, W., Yi, X., & Ma, S. (2022). Effect of Unplugged Programming Teaching Aids on Children's Computational Thinking and Classroom Interaction: With Respect to Piaget's Four Stages Theory. Journal of Educational Computing Research, 60(5), 1277–1300. https://doi.org/10.1177/07356331211057143

Zhang, J. X., Liu, L., Pablos, P. O. de, & She, J. (2014). The auxiliary role of information technology in teaching: Enhancing programming course using alice. The International Journal of Engineering Education, 30(3), 560–565.

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. Computers & Education, 141, 103607. https://doi.org/10.1016/j.compedu.2019.103607