

The Cognitive Benefits of Learning Computer Programming:
A Meta-Analysis of Transfer Effects

Ronny Scherer

University of Oslo, Norway

Fazilat Siddiq

Nordic Institute for Studies in Innovation, Research and Education (NIFU), Norway

Bárbara Sánchez Viveros

Humboldt-Universität zu Berlin, Germany

Author Note

Ronny Scherer, Department of Teacher Education and School Research (ILS) & Centre for Educational Measurement at the University of Oslo (CEMO), Faculty of Educational Sciences, University of Oslo; Fazilat Siddiq, Nordic Institute for Studies in Innovation, Research and Education (NIFU), Oslo; Bárbara Sánchez Viveros, Faculty of Life Sciences, Humboldt-Universität zu Berlin, Germany.

Correspondence concerning this article should be addressed to Ronny Scherer, Faculty of Educational Sciences, Department of Teacher Education and School Research (ILS), Postbox 1099 Blindern, 0317 Oslo, Norway. E-Mail: ronny.scherer@cemo.uio.no

© 2018, American Psychological Association. This paper is not the copy of record and may not exactly replicate the final, authoritative version of the article. Please do not copy or cite without authors' permission. The final article will be available, upon publication, via its DOI: 10.1037/edu0000314

Abstract

Does computer programming teach students how to think? Learning to program computers has gained considerable popularity, and educational systems around the world are encouraging students in schools and even children in kindergartens to engage in programming activities. This popularity is based on the claim that learning computer programming improves cognitive skills, including creativity, reasoning, and mathematical skills. In this meta-analysis, we tested this claim performing a three-level, random-effects meta-analysis on a sample of 105 studies and 539 effect sizes. We found evidence for a moderate, overall transfer effect ($g = 0.49$, 95 % CI = [0.37, 0.61]), and identified a strong effect for near transfer ($g = 0.75$, 95 % CI = [0.39, 1.11]) and a moderate effect for far transfer ($g = 0.47$, 95 % CI = [0.35, 0.59]). Positive transfer to situations that required creative thinking, mathematical skills, and metacognition, followed by spatial skills and reasoning existed. School achievement and literacy, however, benefited the least from learning to program. Moderator analyses revealed significantly larger transfer effects for studies with untreated control groups than those with treated (active) control groups. Moreover, published studies exhibited larger effects than grey literature. These findings shed light on the cognitive benefits associated with learning computer programming and contribute to the current debate surrounding the conceptualization of computer programming as a form of problem solving.

Keywords: Cognitive skills; computational thinking; computer programming; three-level meta-analysis; transfer of skills

Educational Impact and Implications Statement: In this meta-analysis, we tested the claim that learning how to program a computer improves cognitive skills even beyond programming. The results suggested that students who learned computer programming outperformed those who did not in programming skills and other cognitive skills, such as

creative thinking, mathematical skills, metacognition, and reasoning. Learning computer programming has certain cognitive benefits for other domains.

The Cognitive Benefits of Learning Computer Programming:

A Meta-Analysis of Transfer Effects

Computer programming is an activity similar to solving problems in other domains: It requires skills, such as decomposing, abstracting, iterating, and generalizing, that are also required in mathematics and science—in fact, these skills are critical to human cognition (Román-González, Pérez-González, & Jiménez-Fernández, 2017; Shute, Sun, & Asbell-Clarke, 2017). Acknowledging these commonalities between the skills required in programming and the skills required to solve problems in other domains, researchers and computer scientists have claimed that learning to program computers has certain cognitive benefits (Grover & Pea, 2013; Liao & Bright, 1991; Pea & Kurland, 1984). According to this hypothesis, intervention studies that are aimed at fostering programming skills should not only reveal direct training effects but also transfer effects to situations that require other cognitive skills. Yet, the current research abounds in conflicting findings, as there is evidence both for and against the transferability of learning computer programming (Scherer, 2016), and some researchers claimed that far transfer does not exist (Denning, 2017). This observation is by no means unique to programming: Sala and Gobet (2017a) reviewed several meta-analyses in the domains of chess instruction, music education, and working memory training and concluded that so-called ‘far transfer’ (i.e., the transfer of knowledge or skills between two dissimilar contexts) may not exist. However, does this hold for learning to program computers as well? With the current meta-analysis, we investigated this question by testing the hypothesis that programming interventions have certain cognitive benefits. In particular, we examined (a) the overall transfer effect of learning computer programming, (b) the near transfer effects to situations that require programming skills and the far transfer effects to situations that require skills outside of programming, and (c) the differential far effects computer programming interventions may have in situations that require different

types of cognitive skills. In this meta-analysis, programming skills were defined as the skills to create, modify, and evaluate code *and* the knowledge about programming concepts and procedures (e.g., objects, algorithms). These two dimensions are referred to as ‘Computational concepts’ and ‘Computational practices’ in the existing frameworks of computational thinking (Lye & Koh, 2014).

The Transfer of Skills

The question whether acquired knowledge and skills can be transferred from one context or problem to another is key to cognitive and educational psychology. In fact, the transfer of learning lies in the very heart of education, as it taps the flexible application of what has been learned (Barnett & Ceci, 2002). Perkins and Salomon (1992) understood transfer as a situation in which learning in one context impacts learning and performance in other, perhaps new contexts. Although researchers agreed on this definition (Bransford & Schwartz, 1999), some questions remain: Which conditions foster successful transfer? What characterizes “other” or “new” contexts?

In their seminal article, Woodworth and Thorndike (1901) considered improvements in basic cognitive skills and the transfer to situations that require other cognitive skills. Their main proposal for explaining successful transfer is referred to as the ‘Theory of Common Elements’—a theory hypothesizing that the degree of successful transfer depends on the elements two different contexts or problem situations share. The authors argued that the transfer of skills between situations that have less in common (i.e., require only few shared skills or knowledge elements) occurs less often than transfer between closely related situations (see also Bray, 1928). Barnett and Ceci (2002) pointed out that the Theory of Common Elements has led to the distinction between *near* and *far transfer*. In this context, near transfer refers to successful transfer between similar contexts, that is, contexts that are closely related and require the performance of similar skills and strategies; far transfer refers

to successful transfer between dissimilar contexts, that is, contexts that are inherently different and may require different skills or strategies (Perkins & Salomon, 1992). In essence, the transfer of skills depends on the similarity and overlap between the contexts and problems in which the skills were acquired and those presented later on (Schunk, 2012). The issue with these definitions lies in the concepts of similarity and difference, both of which are features of the specific problem situations (Bransford et al., 2005). Greeno et al. (1998) emphasized that the transfer of skills to other contexts is highly situation-specific, that is, it largely depends on the situations in which the skills have been acquired previously—in other words, transfer is situated in experience and is influenced by the participation in previous activities (see also Lobato, 2006). Bransford and Schwartz (1999) pointed out that prior knowledge forms an additional prerequisite for successful transfer, in particular the knowledge about structure of a problem situation, the variables involved, and solution strategies (e.g., Bassok, 1990; Chen & Klahr, 2008; Cooper & Sweller, 1987). It therefore seems that the acquisition of schemata to solve problems may foster the transfer of learning between problem situations.

Although the existence of far transfer was often denied (Barnett & Ceci, 2002; Denning, 2017), several studies provided evidence for far transfer, yet to varying degrees (Bransford & Schwartz, 1999). In a recent review paper, Sala and Gobet (2017a) questioned the existence of far transfer and referred to a series of meta-analyses in the domains of chess instruction and music education. Indeed, the meta-analyses the authors referred to provided only limited evidence for far transfer effects—successful transfer could only be found in situations that required skills similar to those trained in the interventions. Melby-Lervåg, Redick, and Hulme (2016) supported this finding by their meta-analysis of working memory training, and so did Sala, Tatlidil, and Gobet (2018) in their meta-analysis of video gaming. These findings suggest that far transfer may be differentially effective for improving cognitive skills. Overall, our brief review of the existing literature of transfer revealed that (a) transfer is

more likely to occur between closely related contexts or problem situations; (b) the success of transfer depends on schematic knowledge; (c) far transfer may differ across contexts.

The Transfer of Programming Skills

Programming skills are considered critical to the development of “computational thinking”—a concept that “involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science” (Wing, 2006, p. 33). In their seminal review, Shute et al. (2017) named five cognitive processes involved in computational thinking: Problem reformulation, recursion, problem decomposition, abstraction, and systematic testing. These skills defined the concept as a form of problem solving (Lye & Koh, 2014). Despite the close relationship between programming skills and computational thinking, the two concepts are not identical—the latter also entails taking computational perspectives (i.e., students’ understanding of themselves and their interaction with others and with technology; Shute et al., 2017) as an element of computational participation (Kafai & Burke, 2014). Nevertheless, the processes involved in programming require problem-solving skills, such as decomposing problems, applying algorithms, abstracting, and automatizing, and ultimately aid the acquisition of computational thinking skills (Yadav, Good, Voogt, & Fisser, 2017). Programming may therefore be considered a way of teaching and learning computational thinking (Flórez et al., 2017), a way of assessing computational thinking (Grover & Pea, 2013), and a way of exposing students to computational thinking by creating computational artefacts, such as source code or computer programs (Lye & Koh, 2014). Barr and Stephenson (2011), as they compared core computational thinking with the demands of solving problems in STEM domains, concluded that programming skills, computational thinking, and problem solving are intertwined.

In this meta-analysis, we define programming skills as the skills to create, modify, and evaluate code *and* the conceptual and procedural knowledge needed to apply these skills, for

instance, in order to solve problems—a definition close to that of computational thinking. This definition includes two key dimensions of computational thinking: computational concepts (i.e., syntactic, semantic, and schematic knowledge) and computational practices (strategic knowledge and problem solving; e.g., Lye & Koh, 2014). Hence, the research on the transfer of programming skills we review here also targets aspects of the transfer of computational thinking skills.

As learning computer programming engages students in problem solving activities, transfer effects on students' performance in situations that require problem solving seem likely (Shute et al., 2017). Although Pea and Kurland (1984) doubted the existence of such effects, they still argue that some effects on thinking skills that are close to programming may exist. Part of this argument is the observation that problem solving and programming skills share certain subskills. In a conceptual review of problem solving, creative thinking, and programming skills, Scherer (2016) listed several subskills that are required to successfully solve tasks in these three domains. The author concludes that these commonalities provide sufficient ground to expect a positive transfer between them. Clements (1995) established that creativity plays a role in programming, and Grover and Pea (2013) supported this perspective. Reviewing further domains and contexts, Clements (1986a) claimed that programming skills can even be assigned to the cognitive dimensions of intelligence frameworks—hence, a transfer of programming skills to intelligence tasks seems likely. The author further suggested considering metacognitive skills as integral parts of programming. Finally, Shute et al. (2017) identified problem solving and modeling as two commonalities between programming and mathematical skills, arguing for the existence of transfer effects. The list of cognitive skills that overlap with programming could be extended even further (for a detailed overview, please refer to Scherer, 2016). However, the selection presented here already points into one

direction: programming skills and other cognitive skills share important subskills, and transfer effects of learning computer programming may therefore exist.

A recently published, cross-sectional study of computational thinking provided some evidence supporting this reasoning: Román-González et al. (2017) developed a performance test of computational thinking and administered it to 1,251 Spanish students in grade levels 5 to 10. The results showed that computational thinking was significantly and positively related to other cognitive skills, including spatial skills ($r = .44$), reasoning skills ($r = .44$), and problem-solving skills ($r = .67$). Drawing from the Cattell-Horn-Carroll [CHC] theory (McGrew, 2009), Román-González et al. (2017) concluded that computational thinking, operationally defined and measured as what we consider programming skills in this meta-analysis, represents a form of problem solving. Although these findings suggest that programming skills overlap with other cognitive skills, they do not provide evidence for the transferability of programming skills, due to the lack of experimental manipulation.

Previous Meta-Analyses on the Transfer of Programming Skills

Two meta-analyses addressed the transferability of programming skills, both of which resulted in positive and significant effect sizes. The first meta-analysis synthesized 432 effect sizes from 65 studies that presented students with programming activities and administered assessments of cognitive skills (Liao & Bright, 1991). Using a random-effects model, Liao and Bright obtained an overall effect size of $d = 0.41$ ($p < .01$) and thus supported the claim that programming skills can be transferred. Liao and Bright further found that this overall transfer effect size was moderated by the type of publication (with largest effects for published articles in the database ERIC), grade level (with largest effects for college and K-3 students), the programming language used during the intervention (with largest effects for Logo and BASIC), and the duration of the intervention (with largest effects for short-term

interventions). Neither the design of the primary studies nor the year of publication explained variation in the overall effect size.

Although this study provided evidence for the transferability of computer programming based on a large sample of effect sizes, we believe that it has got two shortcomings: First, the authors reported an overall effect size for transfer without differentiating between cognitive skills. Existing meta-analyses that examined transfer effects in other domains, however, found that transfer effects vary considerably across cognitive skills (Melby-Lervåg et al., 2016; Sala & Gobet, 2016). In other words, transfer intervention studies may be particularly effective in situations that require cognitive skills close to the trained skills (Sala & Gobet, 2017a). Second, Liao and Bright (1991) included a dataset that comprised 432 effect sizes from 65 studies—a dataset that clearly had a nested structure (i.e., effect sizes were nested in studies). Considering the recent methodological advancements of meta-analyses (M. W.-L. Cheung, 2014), a three-level random-effects modelling approach would have been more appropriate than the random-effects model Liao and Bright specified, as it quantifies both within- *and* between-study variation.

In the second meta-analysis, Liao (2000) updated the former meta-analysis and included 22 interventions and 86 effect sizes that were published between 1989 and 1999. Aggregating these effects resulted in a large overall transfer effect of $d = 0.76$ ($p < .05$). In contrast to the original meta-analysis, pre-experimental study designs were included (e.g., one-group pretest-posttest designs). Considering that these designs provided the smallest transfer effects ($d = 0.45$) among all other designs ($d = 0.56$ – 2.12), their inclusion may have biased the overall effect. Moreover, the reported effects must be interpreted with caution, given the small sample size of studies and effect sizes. In contrast to Liao and Bright (1991), Liao (2000) tested whether transfer effects differed across cognitive skills. Indeed, the strongest effects occurred for the near transfer of skills ($d = 2.48$), whereas the smallest effects

occurred for the far transfer to creative thinking situations ($d = -0.13$). Other skills such as critical thinking, problem solving, metacognition, and spatial skills benefited from learning computer programming moderately ($d = 0.37-0.58$).

Uttal et al. (2013) included seven studies that administered programming interventions to enhance students' spatial skills. Although the authors did not report an overall effect size for this selection of studies, six out of the seven primary effect sizes of these interventions were positive and significant ($g = 0.12-0.92$, $p < .05$). This finding uncovered that positive transfer of learning computer programming on situations that require the application of spatial skills may exist.

In their review of video gaming, Sala et al. (2018) claimed that “teaching the computer language Logo to improve pupils' thinking skills has produced unsatisfactory results” (p. 113) and referred to two intervention studies. Although this claim was in line with the authors' main argument, we believe that it stands on shaky legs, given the plethora of Logo intervention studies that showed positive far transfer effects (e.g., Clements & Sarama, 1997; Lye & Koh, 2014; Scherer, 2016; Shute et al., 2017). Nevertheless, we agree with their position that the existing research on far transfer in this area abounds in mixed results—some studies found significant effects, while others failed to provide evidence for far transfer (Palumbo, 1990; Salomon & Perkins, 1987). This controversy motivated the present meta-analysis. Overall, the previous meta-analyses of the transferability of computer programming suggested possible, positive transfer effects. However, we identified several methodological and substantive issues which primarily referred to the specification of meta-analytic models, the differentiation of cognitive skills, and the treatments of control groups.

The Present Meta-Analysis

In this meta-analysis, we synthesize the evidence on the transferability of learning computer programming to situations that require certain cognitive skills. Along with

providing an overall transfer effect, we examine the variation and consistency of effects across studies, types of transfer, and cognitive skills. We believe that the rapid advancements in technology and the development of visual programming languages (e.g., Scratch) next to text-based languages (e.g., Java) necessitate an update of the existing research. Besides, acquiring computational thinking skills through programming has received considerable attention lately: Programming is introduced into school curricula in several educational systems—this development is largely based on the claim that learning computer programming has certain cognitive benefits in other domains and contexts (Grover & Pea, 2013; Lye & Koh, 2014). We provide some answers to the question whether learning to program helps to improve cognitive skills and extend the existing research literature on the transfer of skills, which recently focused on chess and music instruction, working memory training, and video gaming, by testing the claims of transfer effects for the domain of computer programming. More specifically, we focus on the following research questions:

1. *Overall transfer effects:* (a) Does computer programming training improve performance on cognitive skills tasks, independent of the type of transfer or cognitive skill? (b) To what extent are these effects moderated by study, sample, and measurement characteristics?
2. *Near transfer effects:* (a) Does computer programming training improve performance on assessments of computer programming skills? (b) To what extent are these effects moderated by study, sample, and measurement characteristics?
3. *Overall far transfer effects:* (a) Does computer programming training improve performance on tasks assessing cognitive skills other than computer programming? (b) To what extent are these effects moderated by study, sample, and measurement characteristics?

4. *Far transfer effects by cognitive skills:* (a) Does computer programming training improve performance on tasks assessing reasoning, creative thinking, metacognition, spatial skills, mathematical skills, literacy, and school achievement in domains other than mathematical skills and literacy? (b) To what extent do these far transfer effects differ across the types of cognitive skills and subskills?

First, we examine the overall transfer effects of computer programming training (Research Question 1a). These effects include benefits for programming skills *and* skills outside of the programming domain. The main purposes of providing answers to this research question are (a) to set a reference for the overall cognitive benefits, and (b) to compare the findings obtained from our meta-analysis with those reported by Liao and Bright (1991), who treated “cognitive skills”, although measured by several skills, as a univariate outcome. Although the overall transfer effect already provides insights into the cognitive benefits of learning computer programming, we believe that a further differentiation into the skills is needed that are required in the new situations and contexts. Indeed, the findings of existing meta-analyses examining transfer effects of cognitive skills trainings warranted further differentiation either by the type of transfer or by the cognitive skills (e.g., Bediou et al., 2018; Melby-Lervåg et al., 2016; Sala & Gobet, 2017a).

We add possible moderators to explain variation in the reported effect sizes (Research Question 1b). The key premise for addressing this question is that effect sizes may vary within and between studies—moderating variables can therefore explain variation at the study level or the level of effect sizes. Possible moderators represent the study, sample, and measurement characteristics, such as the statistical study design, types of control groups, educational level of study participants, programming tools, types of performance tests, and the subskills assessed by performance tests.

Second, we quantify the immediate, near transfer effects of learning computer programming to situations and tasks that require programming skills and explain possible variation within or between studies by the above-mentioned moderators (Research Questions 2a & 2b). Third, we examine the overall far transfer effects and possible moderators thereof (Research Questions 3a & 3b). This study of the overall far transfer is based on measures of skills other than programming and does not differentiate between the different types of cognitive skills. Finally, we differentiate between different types of cognitive skills to provide more information on the far transfer effects (Research Questions 4a & 4b). These skills represent a range of domain-general and domain-specific skills—skills that show a relative distance to computer programming. To further substantiate the skill- and situation-specificity of far transfer effects, we compare the resultant effect sizes across cognitive skills. This comparison also unravels whether certain cognitive skills benefit from computer programming training more than others.

Method

Literature Search and Initial Screening

To identify the primary literature relevant to this meta-analysis, we performed searches in literature databases, academic journals, reference lists of existing reviews and meta-analyses, publication lists of scholars, and the informal academic platform ResearchGate. The database search included ACM Digital Library, IEEE Xplore Digital Library, ERIC, PsycINFO, Learn Tech Library, ProQuest Dissertations and Theses Database, and Google Scholar (the first 100 publications as of January 31, 2017), and focused on publications that had been published between January 1, 1965 and January 31, 2017. The databases ACM Digital Library, IEEE Xplore Digital Library, Learn Tech Library, ResearchGate, and Google Scholar contained both publications in peer-reviewed academic journals and grey literature. We referred to Adams et al.'s (2017) definition of “grey

literature”, which included dissertations, conference proceedings, working papers, book chapters, technical reports, and other references that have not been published in scholarly journals after peer-review (see also Schmucker et al., 2017).

Whenever Boolean search operators were possible (e.g., ERIC, PsycINFO), the following search terms were used: (Programming OR coding OR code OR Scratch* OR Logo* OR Mindstorm* OR computing OR computational thinking) AND (teach* OR learn* OR educat* OR student* OR intervention OR training) AND Computer* AND (compar* OR control group* or experimental group* OR treatment). These terms were comprised of three core elements: the concepts of programming and relevant programming tools (e.g., Scratch and Logo), the context of teaching, training, and interventions, and the design of relevant studies (i.e., studies with treatment and control groups). Whenever needed, we adapted them to the search criteria set by the databases (for details, please refer to the Supplementary Material A2). All searches were limited to titles, abstracts, and keywords.

Besides the search in databases, we also hand-searched for publications in relevant academic journals, and reference and citation lists (whenever possible, via the ISI Web of Knowledge) of existing reviews and meta-analyses on the following topics: teaching and learning computer programming, the concept of computational thinking, and the effects of training spatial skills and creativity (see Supplementary Material A2). From the existing meta-analyses, however, (Liao, 2000; Liao & Bright, 1991), we could only retrieve the studies and effect sizes reported there to a limited extent, because (a) several publications were no longer available in a readable format given their publication year (before 2000)—we contacted twenty authors directly via email or via the messaging tool implemented in ResearchGate; five authors responded to our queries and sent us their publications; (b) inclusion and exclusion criteria of the transfer studies differed between the two meta-analyses; (c) pre-experimental designs were included in these meta-analyses. Finally, we reviewed the formal

and informal publication lists of scholars in the field (Bright, Clements, Kazakoff, Liao, Pardamean, Pea, Grover, and Resnick) via Google Scholar and ResearchGate. In August 2017, we received a notification about two additional, empirical studies that had been published that month (Erol & Kurt, 2017; Psycharis & Kallia, 2017)—these studies entered our list of possibly relevant publications. Despite our efforts to retrieve unpublished studies (e.g., in the form of conference presentations or informal communications) from authors and associations in the field, we did not receive any unpublished material.

Overall, our literature search resulted in 5,193 publications (see Figure 1). After removing duplicates and screening titles for content fit (i.e., the studies must concern computer programming), 708 publications were submitted to an initial screening of abstracts. We read each abstract and examined whether the publication presented a training of computer programming skills and was of quantitative nature; conceptual papers that presented computer programming tools and theoretical reviews without any quantitative evaluation were discarded. This initial screening addressed the criteria of relevance, quantitative data sufficiency, and the presence of an intervention, and resulted in 440 eligible abstracts. The results of both the literature search and the initial screening are shown in Figure 1.

Screening and Eligibility Criteria

The extracted publications were further screened based on inclusion and exclusion criteria (Figure 1). As the current meta-analysis focuses on the transfer effects of learning to program as results of an intervention—including near transfer effects (i.e., effects on performance in programming or computational thinking) *and* far transfer effects (i.e., effects on performance in related cognitive constructs, such as reasoning skills, creative thinking, spatial skills, or school achievement)—studies with an experimental or quasi-experimental design that reported pretest and posttest performance or posttest performance only were included. In line with existing meta-analyses on transfer effects in other domains (e.g., Melby-

Lervåg, Redick, & Hulme, 2016; Sala & Gobet, 2016), we excluded studies with pre-experimental designs (e.g., single-group pretest-posttest designs without any control group).

Overall, studies were included in our meta-analysis if they met the following criteria:

1. *Accessibility*: Full texts or secondary resources that describe the study in sufficient detail must have been available.
2. *Study design*: The study included a training of computer programming skills with an experimental or a quasi-experimental design and at least one control group (treated or untreated); correlational, ex-post facto studies, or pre-experimental designs (e.g., one-group pretest-posttest designs) were excluded.
3. *Transfer effects*: The effect of learning computer programming could be isolated; studies reporting the effects of two or more alternative programming trainings without any non-programming condition were excluded.
4. *Reporting of effect sizes*: The study reported data that were sufficient to calculate the effect sizes of learning computer programming.
5. *Grade levels*: Control and treatment group(s) had to include students of the same grade level or age group to achieve sample comparability.
6. *Performance orientation*: The study had to report on at least one cognitive, performance-based outcome measure, such as measures of computer programming, reasoning, creative thinking, critical thinking, spatial skills, school achievement, or similar; studies reporting only behavioral (e.g., number and sequence of actions, response times) or self-report measures (i.e., measures of competence beliefs, motivation or volition) were excluded.
7. *Educational context*: The study samples comprised children or students enrolled in pre-K to 12, and tertiary education; studies conducted outside of educational settings

were excluded to avoid further sample heterogeneity (a similar reasoning can be found in Naragon-Gainey, McMahon, & Chacko, 2017).

8. *Non-clinical sample*: Studies involving non-clinical samples were included; studies involving samples of students with specific learning disabilities or clinical conditions were excluded.
9. *Language of reporting*: Study results had to be reported in English; studies reporting results in other languages without any translation into English were excluded.

In total, 20 % of the studies entering the fine screening (i.e., the application of inclusion and exclusion criteria) were double-screened by the first and the second author. The overall agreement was high, weighted $\kappa = .97$. Disagreement was resolved in a discussion about whether and why specific inclusion and exclusion criteria might or might not apply until consent was achieved. The performance of the inclusion and exclusion criteria resulted in $m = 105$ studies providing $k = 539$ effect sizes, as shown in Figure 1 (for more details, please refer to the Supplementary Material A2).

Effect Size Measures

To examine the transfer effects on learning to program on cognitive skills, we extracted the relevant statistics from the eligible studies and transformed them into effect sizes. The resultant effect sizes indicated the degree to which gains in cognitive abilities existed in the treatment group that received a programming intervention, relative to a control group that did not. Hedges' g was reported as an effect size, because it accounted for possible bias due to differences in sample sizes (Borenstein, Hedges, Higgins, & Rothstein, 2009; Uttal et al., 2013). We calculated Hedges' g from pretest-posttest experimental or quasi-experimental and posttest-only designs using the available statistics (e.g., mean scores, standard deviations, Cohen's d , F -values, and t -values). If studies included multiple control groups, we included the transfer effects obtained from all possible treatment-control group

comparisons. Supplementary Material A2 details these calculations, and Supplementary Material A1 documents the resultant effect sizes. Given that only 43.2 % of the reliability coefficients of the cognitive skills measures were available and considering the current disagreement about the impact of unreliability corrections on effect size estimations (M. W.-L. Cheung, 2015), we did not correct the reported effect sizes for the unreliability of the outcome measures.

Coding of Studies

To understand the role of contextual variables for the transfer effects, we extracted information about the study design, the content, purpose, and language of programming, the types of outcome variables, the educational level of participants, the length of the intervention, the publication year and status. These variables were identified as possible moderators explaining variation in effect sizes in previous meta-analyses (Liao, 2000; Liao & Bright, 1991), and defined the contexts in which programming interventions may or may not succeed (Grover & Pea, 2013; Shute et al., 2017). Considering that transfer effects may vary within and between studies, possible moderators may operate at the study-level, the level of effect sizes (or measures), or both levels. Whereas most of the variables listed below served as study-level characteristics (e.g., average age of students, randomization and matching of experimental groups), some of them varied within studies and were thus considered effect-size-level predictors (e.g., statistical study design, treatment of control groups, cognitive skills). To ensure that the coding scheme was reliable, 25 % of the eligible studies were coded independently by first and the third author. The overall agreement was 94 %; conflicts were resolved during a discussion session until consensus was reached. Supplementary Material A1 documents the coded variables. Categorical moderator variables with more than one category were dummy-coded.

Sample characteristics. To describe the samples involved in the studies, we extracted information about participants' average age (in years), the educational level the intervention was targeted at (i.e., pre-kindergarten, kindergarten, primary school [1-6], secondary school [7-13], or university/college), and the proportion of female participants in the sample.

Randomization and matching. To supplement the list of study characteristics, we coded whether individuals or pairs were randomly assigned to the treatment and control conditions. Studies assigning entire classrooms (as a cluster) to the conditions were coded as 'non-random'. If authors failed to communicate the degree of randomization, their study was coded as 'non-random', even though authors labelled their design as 'experimental'. In addition, we coded the matching of the experimental groups with respect to relevant variables (e.g., basic cognitive abilities, computer experience, or sample characteristics including gender, age, and grade level) using the categories 'matched' or 'not matched'.

Type of control group. We coded the type of treatment of the control groups as 'treated' or 'untreated'. Control groups were coded as 'treated' (or active) if they received an alternative training that did not involve programming activities yet was aimed at training a certain cognitive skill. For example, Kim, Chung, and Yu (2013) examined the effects of learning programming with the language Scratch on creative thinking. Whereas the treatment group engaged in the programming instruction, the control group followed regular instruction that was not targeted at improving creativity. For this study, we coded the control group as untreated. Hayes and Stewart (2016) examined the effects of learning Scratch programming on reasoning. Given that the control group engaged in an alternative training of reasoning skills, we coded it as treated.

Studies may contain multiple outcome variables and control groups that were treated to support only one of these outcomes (i.e., they were treated considering one outcome variable, yet untreated considering another outcome variable). The treatment of control groups

is thus a variable measured at the level of effect sizes. At the same time, we tested whether this variable may also explain variation between studies and coded the treatment of control group(s) at the study level as “treated”, “untreated”, or “mixed” as well. Hence, the type of control group(s) served as both an effect size-level and study-level variable.

Student collaboration. A recently published meta-analysis indicated that learning computer programming can be more effective in groups than learning it individually (Umapathy & Ritzhaupt, 2017). Moreover, the transfer of problem-solving strategies may be more effective when students work in pairs (e.g., Uribe, Klein, & Sullivan, 2003). We therefore coded whether students collaborated during the intervention as another possible moderator ($0 = individual\ work$, $1 = collaborative\ work$ during the intervention).

Study context. We coded the context in which programming interventions were administered, either as embedded in regular lessons or as extracurricular activities.

Programming language. The programming languages (tools) used during the interventions were reported and categorized as ‘text-based programming languages’ (e.g., Basic, C, and Java) and ‘visual programming languages’ (e.g., Alice, Logo, and Scratch).

Intervention length. The length of interventions was extracted and reported in hours. In case authors provided the number of school lessons, we assumed an average lesson to last about 45 minutes. This assumption may not reflect the true intervention length but provided an approximation of it in most educational systems. The true time distribution may therefore result in slightly different moderation effects. A lack of reporting the intervention length resulted in missing values.

Cognitive skills. Cognitive skills measures were grouped according to the constructs they measured. These constructs comprised broad and narrow categories both of which are shown in Table 1. Overall, the outcome measures covered programming skills, skills that cannot be assigned to a single domain (i.e., creative thinking, reasoning, spatial skills, and

metacognition), and domain-specific skills (i.e., mathematical skills, literacy, and school achievement in subjects other than mathematics and literacy). Specifically, *creative thinking* comprised the skills needed to exhibit creative behavior, including originality, fluency, flexibility, and elaboration (Hennessy & Amabile, 2010). Creative thinking was mainly assessed by validated performance tests, such as the Torrance Test of Creative Thinking. *Reasoning skills* included not only the skills needed to perform logical (formal) reasoning, which are considered elements of fluid intelligence and problem solving (McGrew, 2009), but also critical thinking skills (i.e., informal reasoning); attention, perception, and memory also fell into the category of intelligence, due to their close relation to the reasoning and intelligence (Sternberg, 1982). Our classification of these subskills resonated with that proposed by Sala and Gobet (2018) and Bediou et al. (2018) in their papers on transfer effects of video gaming. Their classification summarized intelligence, attention, memory, and perception as general cognitive skills surrounding reasoning skills. By and large, reasoning skills were assessed by standardized tests of cognitive abilities and critical thinking (e.g., Cornell's Critical Thinking Test; see Table 1). *Spatial skills* included the skills to memorize and understand spatial objects or processes, and to perform reasoning (Uttal et al., 2013). These skills were mostly assessed by standardized tests of the understanding of two- or three-dimensional objects (Table 1). *Metacognition* referred to the processes underlying the monitoring, adaptation, evaluation, and planning of thinking and behavior (Flavell, 1979), and was mostly assessed in conjunction with certain problem-solving tasks. Despite the dominance of self-report measures of metacognition, the measures used in the selected studies were performance-based and comprised tasks that required, for instance, the representation of a problem, the evaluation of problem situations and strategies, the monitoring of students' comprehension, and the integration of new information in the presence of old information (Table 1). *Mathematical skills* comprised mathematical problem solving, modeling,

achievement in general (e.g., measured by course grades), and conceptual knowledge (Voss, Wiley, & Carretero, 1995). Some of the tests reported in primary studies used standardized mathematics tests, whereas others relied on self-developed assessments (Table 1). *Literacy* spanned several knowledge and skills components, including reading, writing, and listening skills. Most primary studies presented students with writing tasks and evaluated the written pieces against certain linguistic criteria; these tasks were often accompanied by reading comprehension tests (Table 1). Finally, *school achievement* was indicated by performance measures in domains other than mathematics and literacy. These measures assessed students' achievement in Earth Sciences, Social Sciences, and Engineering, often measured by national or teacher-developed achievement tests in these subjects (Table 1). Although mathematical skills and literacy can also be considered aspects of school achievement, we did not assign them to this category in order to avoid introducing further heterogeneity which may have compromised the comparability of the effect sizes within this category. We further extracted information about how these constructs were measured. This information included the origin of the tests (i.e., standardized test, performance-based test developed by researchers or teachers), along with the available reliability coefficients.

Type of transfer. On the basis of the coding of cognitive skills at the level of effect sizes, we coded whether studies focused on near transfer only (i.e., only programming skills were measured), far transfer only (i.e., only skills outside of programming were measured), or near and far transfer at the same time (i.e., programming skills and skills outside of programming were measured). This variable operated at the study level and allowed us to examine its possible moderating effects on the overall transfer effect.

Statistical study design. For the included studies, we coded the statistical design underlying the estimation of effect sizes and the overall study design. Several studies included multiple outcome measures, for which pretest and posttest scores were available to a different

extent. Generally, the statistical and the overall (implemented) study designs will agree; yet, in some cases, they may differ, as the following two examples illustrate: (1) *Transfer studies with one outcome measure*: Although authors reported a pretest-posttest control group design to examine the effects of learning to computer programming on mathematical skills, pretest and posttest measured entirely different skills in mathematics, for instance, the skills to deal with variables (pretest) and conceptual understanding of geometric shapes (posttest). Given the non-equivalence of the pretest and posttest, the statistical design was best represented as a posttest-only control group design (Carlson & Schmidt, 1999). In such cases, effect sizes were extracted using the posttest means and standard deviations only. (2) *Transfer studies with multiple outcome measures*: Statistical study designs sometimes differed within studies, in particular when multiple outcomes were measured. For instance, some authors reported both pretest and posttest scores for one outcome variable, yet only posttest scores for another outcome variable. Whereas the former represents a pretest-posttest design, the latter represents a posttest-only design. Hence, statistical study designs are primarily placed at the level of effect sizes. In addition to treating the study design as an effect size feature, we also coded the overall study design as a study feature using the categories “pretest-posttest design”, “posttest-only design”, or “mixed”. Comparable to the types of control groups, this variable served as both an effect size- and a study-level moderator.

Publication status. To examine the extent to which the reported effect sizes were moderated by the type of publication, we established publication status as another, possible moderating variable. Publication status was thus coded as ‘grey’ or ‘published’. In the current, meta-analytic sample, ‘unpublished’ studies did not exist.

Statistical Analyses

Several studies provided multiple effect sizes, either because they included multiple treatments or control groups, or they reported effects on multiple outcome variables. The

reported effect sizes were therefore dependent (Van den Noortgate, López-López, Marín-Martínez, & Sánchez-Meca, 2013). To account for these dependencies, M. W.-L. Cheung (2015) suggested using either multivariate meta-analysis, which models the covariance between multiple effect sizes derived from multiple outcomes measures, or three-level random-effects modeling, which quantifies the degree of dependence by adding a variance component at a third level of clustering (Pastor & Lazowski, 2018). The latter is particularly suitable for situations, in which the degree of dependence or covariance among multiple outcome variables is unknown (M. W.-L. Cheung, 2014), and returns unbiased estimates of fixed effects (Moeyaert et al., 2017). Considering this and the observation that very few primary studies reported covariances or correlations between multiple outcomes in the current meta-analysis, we decided to account for the clustering of effect sizes in studies by adopting a three-level random-effects modeling approach. For the i th effect size in the j th study, this approach decomposes the effect size y_{ij} into the average population effect β_0 , components u_{2ij} and $u_{(3)j}$ with level-specific variances $Var(u_{2ij}) = \sigma_2^2$ and $Var(u_{(3)j}) = \sigma_3^2$, and residuals e_{ij} with the known sampling variance $Var(e_{ij}) = v_{ij}$ (M. W.-L. Cheung, 2014):

$$y_{ij} = \beta_0 + u_{2ij} + u_{(3)j} + e_{ij} \quad (1)$$

Model (1) represents a three-level random-effects model which is based on the standard assumptions of multilevel modeling (see M. W.-L. Cheung, 2014, for details). This model quantifies sampling variability (level 1), within-study variability (level 2), and between-study variability (level 3). To establish which variance components (i.e., within and between studies) are statistically significant, we compared four models against each other, using likelihood-ratio tests and information criteria: Model 1 represented a random-effects, three-level model with within- and between-study variances (see equation [1]). Model 2 was a random-effects model with only between-study variance, and Model 3 was a random-effects model assuming only variation between effect sizes. Finally, Model 4 represented a fixed-

effects model without any variance component. To quantify the heterogeneity of effect sizes at both levels, we estimated the I^2 statistics based on the level-2 and level-3 variance estimates as follows (Cheung, 2015):

$$I_2^2 = 100\% \cdot \frac{\hat{\sigma}_2^2}{\hat{\sigma}_2^2 + \hat{\sigma}_3^2 + \tilde{v}} \text{ and } I_3^2 = 100\% \cdot \frac{\hat{\sigma}_3^2}{\hat{\sigma}_2^2 + \hat{\sigma}_3^2 + \tilde{v}} \quad (3)$$

In equation (3), \tilde{v} represents the typical within-study sampling variance proposed by Higgins and Thompson (2002).

If statistically significant variation within or between studies exists, the random-effects Models 1-3 can be extended to mixed-effects model by introducing covariates (i.e., possible moderator variables) at the level of effect sizes and studies. Under the standard assumptions of three-level regression, the mixed-effects model with level-2 and level-3 variances and a covariate at the level of effect sizes x_{ij} is:

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + u_{2ij} + u_{3j} + e_{ij} \quad (2)$$

The variance explained by a covariate at the level of effect sizes is estimated by the reduction of level-2 variance when comparing models (1) and (2). We specified all models in the R packages ‘metafor’ (Viechtbauer, 2017) and ‘metaSEM’ (M. W.-L. Cheung, 2018) using restricted maximum likelihood estimation. Supplementary Material A3 contains the R sample code.

Publication Bias and Sensitivity Analysis

To test the robustness of the obtained transfer effects, we conducted several analyses of publication bias: First, we examined the funnel plot and performed trim-and-fill-analyses (Duval & Tweedie, 2000). Second, we compared the effect sizes obtained from published studies and grey literature (Schmucker et al., 2017). Third, we examined the p -curve that resulted from the statistics underlying the transfer effect sizes (Simonsohn, Nelson, & Simmons, 2014). If studies had evidential value, the p -curve should have been right-skewed; a left-skewed curve would indicate publication bias (Melby-Lervåg et al., 2016). Fourth, we

performed a fail-safe N analysis on the basis of Rosenberg's weighted procedure (Rosenberg, 2005). In contrast to other fail-safe N procedures (e.g., Rosenthal's and Orwin's procedures), Rosenberg proposed a weighted approach, which is applicable to both fixed- and random-effects models in meta-analysis and might represent the number of unpublished studies better than the alternative approaches. Fifth, we applied Vevea's and Hedges' (1995) weight function procedure that assumes a dependency between the p -value in a study and the probability of publication (linked via a weight function). All approaches to publication bias were performed in the R packages 'metafor' (Viechtbauer, 2017), 'weightr' (Coburn & Vevea, 2017), and the ' P -curve Online App' (Simonsohn, Nelson, & Simmons, 2017).

We tested the sensitivity of our findings to several factors, including the estimation method, the presence of influential cases, the handling of missing data in moderators, and the different assumptions on the variance components in the main model. For instance, we compared the transfer effects and the existence of possible variation within and between studies between restricted maximum likelihood (REML) and maximum likelihood (ML) estimation. Existing simulation studies indicate that, although both methods may not differ in the estimation of intercepts (i.e., overall effect sizes; Snijders & Bosker, 2012), REML creates less biased between-study variance estimates of random-effects models than ML does (Veroniki et al., 2016). M. W.-L. Cheung (2013) therefore argued for the use of REML in multilevel situations yet suggests comparing the variance components obtained from both estimation methods for validation purposes (see also M. W.-L. Cheung, 2014). Furthermore, the dataset underlying our meta-analysis may contain influential effect sizes. We therefore compared the results of our meta-analysis with and without influential effect sizes. We identified influential effect sizes using Viechtbauer's and Cheung's (2010) diagnostics based on random-effects models in the R package 'metafor'. These diagnostics included student residuals, Cook's distances, and other leave-one-out deletion measures.

Results

Description of Studies

Table 2 summarizes the distribution of the study design, sample, and publication characteristics among the $m = 105$ studies and $k = 539$ effect sizes. Most studies reported effect sizes based on pretest-posttest control group designs and random group assignment but did not match the experimental groups—hence, they were quasi-experimental. Most studies targeted far transfer effects only (87.6 %), and about 70 % of the effect sizes were based on untreated control groups. Interventions were mostly conducted during regular school lessons. Besides these design features, studies primarily used visual rather than text-based programming tools in their interventions. Study participants used these tools to design computer games, maneuver robots, or engage in pure programming activities. Control groups, however, did not use programming tools, but attended lectures or other forms of instruction (see Supplementary Material A2). Standardized and unstandardized test were administered to almost the same extent. These tests measured a variety of cognitive skills, with a clear focus on reasoning, mathematical, and creative thinking skills. Overall, the sample of participants was comprised of mainly primary and secondary school students in Asia and North America. The overall sample contained $N = 9,139$ participants of the primary studies (treatment groups: $N_T = 4,544$; control groups: $N_C = 4,595$), with an average sample size of 87 ($SD = 72$, $Mdn = 66$, range = 14–416). Considering the central tendencies of sample sizes, treatment and control groups were balanced (treatment groups: $M = 43$, $SD = 37$, $Mdn = 30$; control groups: $M = 44$, $SD = 43$, $Mdn = 29$). On average, interventions lasted for 25 hours and ranged between 2 and 120 hours ($SD = 20.9$, $Mdn = 20$ hours). Of the study participants, 49.1 % were female. Most publications describing the study results dated back to the 1980s and 1990s, followed by studies published in the 2010s.

Publication bias

Before quantifying the overall transfer effects, we examined the degree of publication bias in the sample of primary studies. The funnel plot indicated some degree of asymmetry (see Figure 2a)—this observation was supported by Egger's regression test, $t(537) = 4.10$, $p < .001$. Trim-and-fill analysis resulted in an overall transfer effect size of $g = 0.43$, 95 % CI = [0.37, 0.50], without any additional studies to be filled left of the mean. Rosenberg's fail-safe N suggested that 77,765 additional effect sizes would be necessary to turn the overall transfer effect size into insignificant (with $p > .01$). Finally, p -curve analyses indicated that observed p -values had evidential value, $z = -38.9$, $p < .0001$ (continuous test for a right-skewed curve; Simonsohn et al., 2014), and that the p -curve was right-skewed (see Figure 2b). Vevea's and Hedges' (1995) weight function model with a selection function based on p -values with cut-off points of 0.05 and 1 resulted in an adjusted overall effect size of $g = 0.63$, 95 % CI = [0.52, 0.74] that was based on random effects. The difference between this weighted model and a model containing constant weights (i.e., no publication bias) was significant, $\chi^2(1) = 20.4$, $p < .001$. Hence, the publication of effect sizes could depend on the reported p -value, because the model adjusted for publication bias fits better than the unadjusted model (for more details, please refer to Vevea & Hedges, 1995). Taken together, these findings suggest the presence of some publication bias and small-study effects (Egger's test) in the present data. At the same time, p -curve analysis did not uncover the presence of p -hacking, and the fail-safe N indicated that it is unlikely that the key results obtained from the main and moderation models are mainly due to publication bias.

Overall Transfer Effects

To aggregate the transfer effects of learning computer programming on cognitive skills, including programming skills and skills outside of the programming domain, we

established a main (baseline) model, which formed the basis for the subsequent moderator analyses of the overall transfer effects.

Main model (Research Question 1a). To identify the main model, we performed a sequence of modeling steps: First, a random-effects three-level model (Model 1) resulted in positive and moderate transfer effect, $g = 0.49$ ($m = 105$, $k = 539$, 95 % CI = [0.37, 0.61], $z = 8.1$, $p < .001$; Model fit: $-2LL = 1127.8$, $df = 3$, AIC = 1133.8, BIC = 1146.7). This effect was accompanied by significant heterogeneity ($Q[538] = 2985.2$, $p < .001$), which also surfaced in variation of the effect size within studies ($\sigma_2^2 = 0.204$, 95 % CI = [0.164, 0.252], $I_2^2 = 36.7\%$) and between studies ($\sigma_3^2 = 0.281$, 95 % CI = [0.189, 0.415], $I_2^2 = 50.7\%$). The corresponding profile likelihood plots peaked at both variance estimates, and the log-likelihood decreased for higher values of these variances—thus, both variance components were identifiable (see Supplementary Material A2, Figure S1). The intraclass correlations of the true effects were 0.42 (level 2) and 0.58 (level 3), indicating substantial variation within and across studies.

Second, we specified a model with constrained level-2 variance ($\sigma_2^2 = 0$), but freely estimated level-3 variance (Model 2). This model showed the same transfer effect size as the three-level model ($g = 0.49$, 95 % CI = [0.38, 0.61]), along with significant level-3 variance, $\sigma_3^2 = 0.328$, 95 % CI = [0.238, 0.458], $I_2^2 = 82.4\%$ (Model fit: $-2LL = 1589.0$, $df = 2$, AIC = 1593.0, BIC = 1601.6). In comparison to Model 1, this model degraded model fit significantly, $\chi^2(1) = 461.2$, $p < .001$.

The third model assumed variation at level 2, yet not at level 3 ($\sigma_3^2 = 0$), thus representing a standard random-effects model (Model 3). This model revealed a positive and moderate effect size, which was slightly smaller than that obtained from the three-level model ($g = 0.43$, $m = 105$, $k = 539$, 95 % CI = [0.37, 0.49], $z = 13.8$, $p < .001$; Model fit: $-2LL = 1266.5$, $df = 2$, AIC = 1270.5, BIC = 1279.1), with significant between-study variation

($\sigma_2^2 = 0.415$, 95 % CI = [0.352, 0.490], $I_2^2 = 85.6$ %). Introducing the constraint of zero level-3 variance degraded the model fit significantly, as the results of a likelihood-ratio test comparing Models 1 and 3 indicated, $\chi^2(1) = 138.7$, $p < .001$.

The fourth model constrained both level-2 and level-3 variances to zero ($\sigma_2^2 = 0$, $\sigma_3^2 = 0$), assuming fixed effects (Model 4). The resultant overall transfer effect amounted to $g = 0.35$ ($m = 105$, $k = 539$, 95 % CI = [0.33, 0.37], $z = 31.1$, $p < .001$; Model fit: - $2LL = 2740.0$, $df = 1$, $AIC = 2742.0$, $BIC = 2746.3$). The three-level random-effects model, however, fitted the data significantly better than this model, $\chi^2(2) = 1612.2$, $p < .001$.

Overall, this sequence of model specifications and comparisons indicated significant level-2 and level-3 variance of the overall transfer effect and the sensitivity of the overall effect size to these variance components. It also showed that the three-level random-effects model represented the data best, $g = 0.49$, $m = 105$, $k = 539$, 95 % CI = [0.37, 0.61].

Moderator analysis (Research Question 1b). Model 1 formed the basis for further moderator analyses. Table 3 shows the results of these analyses for the categorical moderators. Significantly higher effects occurred for published literature ($g = 0.60$, 95 % CI = [0.45, 0.75]) than for grey literature ($g = 0.34$, 95 % CI = [0.15, 0.52]; $Q_M [1] = 4.67$, $p = .03$). Besides the publication status, only the type of treatment that control groups received (i.e., treated vs. untreated) significantly explained level-2 variance, $Q_M (1) = 40.12$, $p < .001$, $R_2^2 = 16.7$ %. More specifically, transfer effect sizes were significantly lower for studies including treated control groups ($g = 0.16$) than for studies including untreated control groups ($g = 0.65$). Concerning the z -transformed, continuous moderators at level 3, neither publication year ($B = 0.09$, $SE = 0.06$, $Q_M [1] = 2.36$, $p = .12$, $R_3^2 = 0.0$ %), students' average age ($B = -0.07$, $SE = 0.07$, $Q_M [1] = 0.86$, $p = .35$, $R_3^2 = 3.6$ %), the proportion of female students in the study samples ($B = -0.07$, $SE = 0.07$, $Q_M [1] = 0.86$, $p = .35$, $R_3^2 = 1.1$ %), nor the intervention length ($B = 0.00$, $SE = 0.06$, $Q_M [1] = 0.00$, $p = .98$,

$R_3^2 = 0.0\%$) affected the overall transfer effect, thus leaving large proportions of level-2 and level-3 variances unexplained.

Sensitivity analyses. The variance components of the overall transfer effect, obtained from REML, differed only marginally from the ML variances (ML level-2 variance: $\sigma_2^2 = 0.203$, 95 % CI = [0.160, 0.247], $I_2^2 = 37.0\%$; ML level-3 variance: $\sigma_3^2 = 0.277$, 95 % CI = [0.169, 0.385], $I_2^2 = 50.3\%$; see Supplementary Material A2, Table S1). Some moderator variables exhibited missing data. Hence, we compared the variance explanations of effect sizes between the maximum likelihood and the full-information maximum likelihood (FIML) approaches. The FIML approach handles missing data within the analysis model by using all observed effect sizes and study characteristics to compensate the loss of information due to missing values (Little et al., 2014) and is implemented in the R package ‘metaSEM’ (‘meta3X()’ function; M. W.-L. Cheung, 2018). Overall, the differences in variance explanations between FIML and ML, and FIML and REML were only marginal (see Supplementary Material A2, Table S2).

The influential cases diagnostics flagged ten influential effect sizes that were obtained from five studies (see Supplementary Material A2, Figure S2). These effect sizes ranged between $g = 2.10$ and $g = 8.63$ ($Mdn = 3.31$), with an average of $g = 3.99$ ($SD = 1.99$). The studies exhibiting these effects all contained primary school students, used visual programming tools, and examined transfer effects on cognitive skills outside of programming; all other sample and study characteristics differed. After removing these effect sizes, the remaining $m = 103$ studies comprising $k = 529$ effect sizes were submitted to the three-level meta-analysis, following the same procedure as for the full data set. Model 1 fitted the data based and revealed a positive, significant, and moderate overall transfer effect size of $g = 0.41$, 95 % CI = [0.32, 0.50], which was slightly lower than the original effect size (see Supplementary Material A2, Table S3). The moderator analyses supported the finding that

studies comprising treated control groups exhibited significantly smaller transfer effects than studies with untreated control groups (see Supplementary Material A2, Table S4). The continuous moderation effects did not change after excluding influential cases (see Supplementary Material A2, Table S5). Nevertheless, two findings contrasted previous moderator analyses with the full data set: First, the difference between published literature and grey literature diminished after removing influential cases, suggesting a possible reduction of publication bias in the data. Indeed, the funnel plot indicated improved graphical symmetry, and the p -curve did not provide evidence for further publication bias (see Supplementary Material A2, Figure S3). Second, studies administering standardized tests showed a significantly lower transfer effect size ($g = 0.33$) than studies administering unstandardized tests ($g = 0.49$; $Q_M [1] = 4.56$, $p = .03$, $R_3^2 = 7.8\%$). Overall, the sensitivity analyses showed marginal differences in the overall transfer effects, their variance components, and possible moderation effects between the conditions—substantive causes for differences could not be identified.

Near and Far Transfer Effects

Taking a second step in our meta-analysis, we analyzed the transfer effects for *near transfer* (i.e., effects on programming skills) and *far transfer* (i.e., effects on cognitive skills outside programming). To allow for possible differences in (a) the selection of a main model, (b) the within- and between-study variances, and (c) the moderation effects, we conducted two separate meta-analyses, following the same procedure as for the overall transfer.

Main models (Research Questions 2a & 3a). Comparisons between models with different variance constraints identified a random-effects model with between-study variation of effect sizes (Model 2) as the best-fitting main model for near transfer effects (Table 4; please find the forest plot in the Supplementary Material A1); for far transfer effects, the random-effects three-level model (Model 1) described the data best (Table 4), indicating

significant variance within and between studies. The overall effect size for near transfer was high ($g = 0.75$, $m = 13$, $k = 19$, 95 % CI = [0.39, 1.11], $z = 4.1$, $p < .001$), and showed substantial heterogeneity across studies ($I_3^2 = 85.5$ %). In contrast, the overall far transfer effect size was lower ($g = 0.47$, $m = 102$, $k = 520$, 95 % CI = [0.35, 0.59], $z = 7.8$, $p < .001$), and showed heterogeneity within ($I_2^2 = 37.1$ %) and between studies ($I_3^2 = 50.0$ %) with intraclass correlations of 0.43 and 0.57, respectively. For both types of transfer, the profile likelihood plots peaked at the estimated variances, testifying to the identification of both variances (see Supplementary Material A2, Figures S4 and S5). Overall, the selection of main models suggested positive and significant near and far transfer effects.

Moderator analyses (Research Questions 2b & 3b). The moderator effects differed between near and far transfer (see Tables 5 and 6): Whereas neither publication status nor the treatment of control groups showed significant moderation for near transfer, far transfer effect sizes were significantly lower for treated control groups ($g = 0.15$) than for untreated control groups ($g = 0.64$) at the level of effect sizes, and significantly higher for published studies ($g = 0.58$) than for grey literature ($g = 0.43$). Studies with random group assignment ($g = 0.29$) showed lower near transfer effects than for those without ($g = 0.95$). None of the continuous study and sample characteristics moderated the two transfer effects (Table 7). Notably, the confidence intervals accompanying near transfer effects were large, due to the limited number of studies addressing this type of transfer. Hence, the moderation effects of near transfer must be treated with caution.

Publication bias. As noted earlier, publication status did not explain variance in near transfer but in far transfer effects, indicating some bias toward published studies in the latter. Moreover, the funnel plots for near and far transfer confirmed this tendency, as they showed some skewness only for far transfer (see Supplementary Material A2, Figure S6). Trim-and-fill analyses suggested adding two more effect sizes for near, yet no further effects for far

transfer; for both, fail-safe N s were large in comparison to the available number of effect sizes (see Supplementary Material A2, Table S10). Overall, some publication bias may exist in the far transfer effects.

Sensitivity analyses. To further substantiate the findings surrounding near and transfer effects, we replicated the sensitivity analyses conducted for the overall transfer effect. Tables S6 to S9 in the Supplementary Material A2 report the results of these analyses. Overall, the selection of the main models underlying near and far transfer effects was neither affected by the method of estimation (REML vs. ML) nor the exclusion of influential cases for the far transfer data set (Tables S6 and S7). The far transfer effect decreased slightly after removing influential cases, $g = 0.39$. Notably, variances decreased after removing influential cases. Moderation effects of far transfer did not differ across sensitivity conditions (Tables S8 and S9). After removing influential cases, the moderation by publication status disappeared and the effect of test type became significant for far transfer; indicators of publication bias were not affected. Taken together, sensitivity analyses provided evidence for the robustness of our findings yet indicated some degree of sensitivity to influential cases.

Far Transfer Effects by Cognitive Skills and Subskills

Finally, we aimed at providing a more fine-grained view on the far transfer effects. Our initial analyses revealed considerable variation in far transfer effect sizes—variation that might be explained by the diversity of cognitive skills assessed in the primary studies. Consequently, we addressed this diversity by conducting separate meta-analyses for each cognitive skill. These meta-analyses included the selection of a main model, the quantification of variance components, and the comparisons of effects between cognitive skills measures.

Main models (Research Question 4a). For the meta-analytic data sets focusing on reasoning, creative thinking, spatial, and mathematical skills, three-level random-effects models represented the data best (Model 1); for metacognitive skills, however, only

significant between-study variance existed (Model 2; see Table 8). Fixed-effects models represented the data for studies assessing literacy and school achievement, due to the small number of available effect sizes showing insignificant within- and between-study variation. Table S11 in the Supplementary Material A2 details the selection of main models. Moreover, Figures S6 to S12 show the profile likelihood plots for each variance component in the meta-analyses; these plots indicated that all variance components were identified.

Overall, the largest effects appeared for creative thinking ($g = 0.73$, 95 % CI = [0.27, 1.20]), followed by mathematical skills ($g = 0.57$, 95 % CI = [0.34, 0.80]). We found positive and moderate effects for metacognitive skills ($g = 0.44$, 95 % CI = [0.01, 0.88]), reasoning skills ($g = 0.37$, 95 % CI = [0.23, 0.52]), spatial skills ($g = 0.37$, 95 % CI = [0.08, 0.67]), and school achievement ($g = 0.28$, 95 % CI = [0.14, 0.42]), all of which were statistically significant ($p < .05$). Nevertheless, we could not find support for positive transfer effects on literacy ($g = -0.02$, 95 % CI = [-0.12, 0.08]). The forest plots for all far transfer effects are shown in the Supplementary Material A1.

Effect size differences across cognitive skills and subskills (Research Question 4b). To test whether far transfer effects differed significantly across cognitive skills, we extended the three-level random-effects models describing far transfer by an additional level, that is, the level of skills. This extended, four-level model fitted the data significantly better than the three-level model (Model fit: $-2LL = 1073.6$, $df = 4$, AIC = 1081.7, BIC = 1098.7; Model comparison: $\chi^2(1) = 13.5$, $p < .001$), suggesting that the between-skills variance of $\sigma_4^2 = 0.045$ (95 % CI = [0.009, 0.227], $I_4^2 = 7.8$ %, ICC = .089) was significant. Nevertheless, the post-hoc tests of effect size differences indicated only few significant differences (see Supplementary Material A2, Table S24). Overall, far transfer effects varied across cognitive skills. However, although the absolute values of effects (i.e., fixed effects) may suggest a hierarchy of transfer effects on cognitive skills, not all effect sizes differed significantly.

We further tested whether the types of subskills or the assessment mode (i.e., verbal vs. non-verbal) explained variation in transfer effects (see Table 9). Indeed, except for mathematical skills, the differentiation into certain subskills moderated the transfer effects on reasoning, creative thinking, and spatial skills significantly, explaining up to 66 % of variance. Notably, problem solving ($g = 0.47$), originality ($g = 1.28$), and spatial memory ($g = 1.25$) benefited the most from learning computer programming. The type of assessment, however, did not show any moderation effect. Tables S12 to S16 in Supplementary Material A2 give a more detailed account of the effects of additional moderator variables.

Publication bias. The finding that far transfer effects were moderated by publication status for selected cognitive skills indicated the existence of some publication bias. Funnel plots, as shown in Figures S13 to S19 (Supplementary Material A2), further uncovered some skewness in the plots, especially for the subsets of studies with few effect sizes. Nevertheless, the trim-and-fill analyses did not show substantial deviations of Hedges' g from the original effect sizes after adding more effects, except for metacognition (see Supplementary Material A2, Table S23). The fail-safe N s were large in comparison to the number of available effect sizes, except for literacy and school achievement.

Sensitivity analyses. We further found that excluding influential cases reduced the variances, especially the between-study variance; however, the decisions for main models remained, except in one case (i.e., data on spatial skills no longer showed between-study variation). Tables S17 and S18 (Supplementary Material A2) detail the results underlying these observations. The main reason for the substantial variance reduction lies in the fact that large effect sizes (i.e., those of influential cases) are more influential when sample sizes are small. Nevertheless, apart from only few deviations, the effects of moderating variables remained in most sensitivity conditions (see Supplementary Material A2, Tables S19 to S22). Notably, transfer effects decreased after removing influential cases for reasoning ($g = 0.32$),

creative thinking ($g = 0.52$), and spatial skills ($g = 0.27$), yet remained statistically significant. The transfer effect on school achievement, however, became insignificant ($g = 0.22$). These findings supported the overall robustness of the results yet point to some sensitivity due to the large effects of influential cases in small meta-analytic samples.

Discussion

In this meta-analysis, we synthesized the existing research that focused on the transferability of learning computer programming to cognitive skills. Adopting a three-level random-effects, meta-analytic approach, we identified a positive, overall transfer effect. Further differentiating between near and far transfer revealed positive effects for both types of transfer. Of the cognitive skills examined for far transfer, the transfer effects were large for creative thinking, mathematical skills, and reasoning—other cognitive skills benefited less (e.g., school achievement, literacy). Only some of these findings were sensitive to the removal of influential cases; all of them were robust across different estimation methods and treatments of missing covariates. The status of control group treatment and publication moderated most of the transfer effects. Table 10 summarizes these key findings.

Overall Transfer Effects of Computer Programming

The overall transfer effect size of $g = 0.49$ was comparable to that reported by Liao and Bright (1991) in their meta-analysis ($d = 0.41, p < .05$), although we were able to include more studies and effect sizes and although we excluded pre-experimental study designs without control groups. Despite the differences between our meta-analysis and theirs, the overall finding—that is, the existence of positive transfer of moderate effect sizes—could be replicated. However, the grand mean of transfer effects Liao (2000) reported in the update of the preceding study ($d = 0.76$) was considerably larger than the one we identified. One possible explanation for this deviation might refer to the small sample of primary studies Liao (2000) extracted from existing research ($m = 22$ studies, $k = 86$ effects).

Overall, the positive transfer effect suggested that learning computer programming has certain cognitive benefits. This finding may have at least two explanations: First, the cognitive skills assessed in the primary studies share conceptual commonalities with programming skills. These commonalities may comprise skills that are relevant for both, programming and other skills—hence, learning computer programming possibly helps developing other skills (Liao & Bright, 1991; Shute et al., 2017). Second, the measures used to assess cognitive skills were aligned with what the interventions focused on. This alignment puts the overall effects in a different perspective: they may be considered immediate training effects rather than transfer effects (Melby-Lervåg et al., 2016). However, the second explanation seems unlikely to us, because (a) cognitive skills measures and the programming skills taught during the interventions differed (e.g., programming intervention with a standardized test of creativity outside the programming domain); (b) the overall effects were only moderate, whereas direct effects of aligned measures in programming tended to be larger in a previous meta-analysis that synthesized direct training effects ($g = 0.64, p < .05$; Umapathy & Ritzhaupt, 2017). Clearly, differentiating between the types of transfer and cognitive skills could shed further light on these explanations.

The overall effect size we found in our meta-analysis was comparable to those obtained from similar transfer effects studies. For instance, examining the effects of chess instruction on several cognitive skills, Sala and Gobet (2016) identified a moderate, overall effect of $g = 0.34 (p < .05)$ for a sample of 24 studies and 40 effect sizes. Bediou et al. (2018) synthesized 90 effect sizes that were derived from 20 intervention studies in the domain of video gaming and found a moderate overall effect, $g = 0.34, p < .05$. Focusing on the same domain, Sala et al. (2018) meta-analyzed experimental and quasi-experimental interventions; aggregating the effect sizes obtained from these interventions resulted in a range of transfer effects from $g = -0.04$ to $g = 0.41$. Examining the transfer effects of working memory training,

Melby-Lervåg et al. (2016) found an even broader range of effects from $g = 0.01$ to $g = 1.88$ for a variety of cognitive skills measures. Although the list of similar transfer effects studies can be extended, this selection of studies shows that (a) the overall transfer effect size of learning computer programming ($g = 0.49$) fell into the range of effects reported for other domains; and (b) the effects may vary considerably across cognitive skills. However, the effects decreased whenever control groups were engaged in alternative treatments—a finding testifying that study designs matter to transfer effects.

Does Far Transfer of Programming Skills Exist?

Differentiating between near and far transfer effects revealed strong effects for near transfer ($g = 0.75$) and moderate effects for far transfer ($g = 0.47$). The former suggests that learning computer programming is effective in teaching programming skills. This finding supported Lye's and Koh's (2014) claim that programming instruction can aid the acquisition of programming skills as elements of computational thinking, a concept that goes even beyond programming. Furthermore, the reported effect size of near transfer was comparable to those reported in a meta-analysis of the effects of pair programming ($g = 0.41$ – 0.64 , $p < .05$; Umapathy & Ritzhaupt, 2017), and indicated the success of direct transfer, that is, transfer to the very skills that were trained during the interventions.

The finding that far transfer effect existed suggests that learning computer programming can support the acquisition of other cognitive skills. The overall claim that programming aids other forms of thinking skills can therefore be substantiated. We believe that programming skills, as key elements of computational thinking, comprise skillsets that are also needed in other domains. In fact, the claims surrounding the transferability of computer programming are largely based on this argumentation: For instance, Shute et al. (2017) conceptualized programming activities as forms of problem solving. Ambrósio, Pereira Júnior, and Georges (2015) argued that programming skills require fluid intelligence

and other forms of reasoning—this argument has later been evidenced by a study conducted by Román-González et al. (2017). These findings exemplify the commonalities programming skills and other cognitive skills share.

We differentiated transfer effects between cognitive skills and the results indicated the differential effectiveness of learning computer programming. Far transfer effects were positive and significant for skills that share certain subskills with programming. For instance, the largest transfer effects occurred for *creative thinking skills*. Existing frameworks outlining what creative thinking skills entail contain skills that are also relevant for the creation of computer code (Scherer, 2016). In fact, Grover and Pea (2013) considered programming “a creative process that produces computational artefacts” (p. 39). This process requires translating problems into computational models to find original solutions (Clements, 1995; Ma, 2006). Indeed, we found that transfer effects were larger for the originality dimension of creative thinking than for the other dimensions. Reviewing studies on the malleability of creativity ($g = 0.68-0.69$, $p < .05$; Ma, 2009; Scott, Leritz, & Mumford, 2004), the effects we extracted from programming interventions were comparable in size ($g = 0.73$). In conclusion, it seems that learning computer programming could be as effective as alternative approaches to enhancing creativity.

Shute et al. (2017) reviewed the similarities and differences between programming as a part of computational thinking and *mathematical thinking* and concluded that skills such as problem solving and modeling are involved in both. More specifically, both programming and mathematical modeling require the abstraction of real-world problems, the formulation as computational models, the application of strategies and algorithms to solve them, and the interpretation of a solution. These shared subskills may explain the strong and positive transfer effects on mathematical skills. Another explanation refers to the tasks used to assess mathematical thinking: in several studies, the understanding of geometric concepts and shapes

was assessed following an intervention that used the Logo programming language with geometric objects. In this sense, the transfer of skills needed to program geometric objects to mathematical skills seems obvious (Clements & Sarama, 1997). Yet another explanation is that mathematically skilled students are more likely to engage in programming activities and therefore excel more than those students who may not be as mathematically skilled as they are (Pea & Kurland, 1984). In any case, the transfer effects on mathematical skills ($g = 0.57$) were larger than those found in similar meta-analyses that focused on the transfer effects of chess instruction ($g = 0.38$, $p < .05$; Sala & Gobet, 2016), technology-based instruction ($d = 0.28$, $p < .05$; Li & Ma, 2010), music education ($d = 0.17$, $p < .05$; Sala & Gobet, 2017b), or working memory training ($g = 0.06$ – 0.12 ; Melby-Lervåg et al., 2016). The effect size was comparable to that of direct training studies (e.g., $d = 0.55$ – 0.58 , $p < .05$; Jacobse & Harskamp, 2011; Stockard, Wood, Coughlin, & Khoury, 2018). Thus, learning computer programming might be an effective approach to developing students' mathematical skills.

Both *metacognition* and *reasoning* are involved in programming activities: whereas the former comprises activities such as debugging and evaluating solution strategies (McCauley et al., 2008), the latter involves algorithmic and logical thinking as well as formal and informal reasoning (Shute et al., 2017; Yadav et al., 2017). Considering these activities and the existing evidence on the positive effects of teaching metacognitive strategies (e.g., self-monitoring) on the transfer of skills (Bransford & Schwartz, 1999; Chen & Klahr, 2008), transfer effects were expected; however, they were not as high as those of creativity and mathematical skills, possibly due to a larger degree of domain-generality of reasoning and metacognition (Greiff et al., 2014). The effects reported in our study (metacognition: $g = 0.44$, $p < .05$; reasoning: $g = 0.37$, $p < .05$) were comparable to those reported in other meta-analyses (e.g., chess instruction: $g = 0.33$, $p < .05$; Sala & Gobet, 2016; programming: $d = 0.39$, Liao, 2000).

The finding that *spatial skills* benefit from learning computer programming may be explained by the focus of several transfer studies on geometry objects and the understanding of movements (Clements & Sarama, 1997). The resultant transfer effect ($g = 0.37$) falls in the range of transfer studies using video gaming ($g = 0.20$ – 0.45 , Bediou et al., 2018; Sala et al., 2018) and working memory training ($g = 0.28$ – 0.51 ; Melby-Lervåg et al., 2016)—it is, however, smaller than the average effect of direct training studies of spatial skills ($g = 0.47$, $p < .05$; Uttal et al., 2013).

School achievement, primarily in natural and social sciences, was aided the least by programming interventions ($g = 0.28$). Programming interventions did not show any transfer effects on *literacy*. A possible explanation for the former finding may lie in the fact that school achievement was assessed by distal measures, such as grades or subject-specific tests—measures that tap subject-specific knowledge next to generic skills. We also observed that literacy was mainly measured by reading comprehension and writing skills—skills that overlap only marginally with programming. Sala and Gobet (2017b) obtained a similar result while studying effects of music education on literacy ($d = -0.07$). We emphasize that the number of studies used to study effects on literacy was small, and the interventions presented therein may not have been tailored to foster literacy through programming. Despite the existing enthusiasm related to the possible transferability of computer programming to situations that require literacy (Hutchison, Nadolny, & Estapa, 2016), our meta-analysis does not provide support for it.

Overall, the positive effects of far transfer were not homogeneous across cognitive skills and thus necessitated further differentiation. Nevertheless, both the analyses of the full data set and the analyses in the sensitivity conditions did not suggest a stringent hierarchy of far transfer effects, in which creative thinking, mathematical skills, and metacognition benefit the most. Not all differences in effect sizes between cognitive skills were significant, and

smaller effects occurred after excluding influential cases so that the “order” of effects changed. These results were mainly due to large uncertainties and small meta-analytic samples.

Despite the increasing attention computer programming has received recently (Grover & Pea, 2013), programming skills do not transfer equally to different skills—a finding that Sala and Gobet (2017a) supported in other domains. The findings of our meta-analysis may support a similar reasoning: the more distinct the situations students are invited to transfer their skills to are from computer programming the more challenging the far transfer is. However, we notice that this evidence cannot be interpreted causally—alternative explanations for the existence of far transfer exist. For instance, besides the possible, common elements between what is learned during programming interventions and what tasks assessing metacognition require as causes for positive far transfer on metacognition, the programming interventions may have indeed improved students’ metacognitive skills, such as the evaluation and monitoring of problem solving processes (Clements, 1986b). This interpretation assumes that learning computer programming may stimulate the development of cognitive skills that are required to solve complex problems in other contexts and domains. In any case, the positive effect sizes at least suggest that learning programming “does not do harm” to other cognitive skills. Denning’s (2017) recent claim that no evidence existed for the transfer of programming cannot be substantiated.

Sensitivity of Findings and Effects of Moderators

Overall, despite marginal differences in the reported effect sizes, their variance components, and the effects of moderators, the sensitivity analyses confirmed the findings we presented in our meta-analysis. Nevertheless, further differentiating between the two types of transfer and cognitive skills, as it decreases the number of available effect sizes for each of the separate meta-analyses, increased the impact of some sensitivity conditions. For instance, the

impact of excluding influential effect sizes for creativity decreased the transfer effect size substantially. This observation is primarily an effect of smaller meta-analytic samples; yet, secondary, more substantive reasons could not be identified. Finally, the sensitivity analyses revealed the robustness of reported effects against estimation procedures and the treatment of missing covariates.

Considering our moderator analyses, the treatment of control groups moderated the overall and far transfer effect, with larger effect sizes for untreated controls. As we have mentioned earlier, it is critical to review the activities the control groups are engaged in to obtain a more differentiated view on transfer effects. The fact that transfer effects were smaller in studies with treated control groups shows that alternative trainings exist that might be equally effective as learning computer programming (Hayes & Stewart, 2016; Salomon & Perkins, 1987).

Similar to the treatment of control groups, publication status moderated the overall and far transfer effects, with larger effects for published studies. This finding replicates the significant moderation effects in previous meta-analyses (Liao, 2000; Liao & Bright, 1991), and may indicate some publication bias toward published studies (Polanin, Tanner-Smith, & Hennessy, 2016). For instance, if the grey literature was removed, the overall transfer effect size would have increased to $g = 0.60$. However, these moderation effects were not consistent across all types of transfer (i.e., they did not exist for near transfer) and cognitive skills—considering this, publication bias may not exist to the same extent in all sub-samples. Overall, our meta-analysis provides only limited hints on possible reasons for the mixed results of transfer effect studies in programming, as only the treatment of control groups and publication status moderated the effects. To this end, it remains for future research to clarify which additional factors may explain both within- and between-study variation of transfer effects.

In contrast to our expectations and the findings reported by Liao and Bright (1991) and Liao (2000), study design characteristics neither moderated the overall transfer effects nor near and far transfer effects; only for metacognitive and spatial skills, moderation by study design and randomization was apparent. The effect sizes extracted from the primary studies can therefore be considered homogeneous across designs. Similarly, we did not find any contextual moderator effects (i.e., effects of the programming languages, educational level of participants, and intervention length). On the one hand, this observation highlights the robustness of the results; yet, on the other hand, it complicates the practical implications our study may have. At this point, we cannot identify specific conditions under which the transfer of programming skills is most effective. The question which instructional approach may foster the transfer of programming best remains unanswered.

Methodological Issues of Transfer Effect Studies

This meta-analytic review uncovered several methodological issues with studying near and far transfer effects of computer programming: First, only few studies included a baseline measure to disentangle the training effects of computer programming from the actual transfer effects on other constructs. To draw a more detailed picture of the nature of transfer, however, baseline measures of programming skills, next to measures of other skills, are needed on all measurement occasions (for a general discussion of this issue, please refer to Gagné, Foster, & Crowley, 1948; Melby-Lervåg et al., 2016).

Second, studies including multiple outcome measures often failed to report all relevant statistics to calculate precise effect sizes, such as the correlations or covariances between the outcome measures. If in fact correlations had been made available for such studies, multivariate meta-analysis could have been performed—a second, effective approach to handling dependent effect sizes next to three-level meta-analysis (M. W.-L. Cheung, 2014). A

combination between multivariate and multilevel meta-analysis could have provided deeper insights into the link between multiple outcome variables across studies.

Third, another concern relates to the sample sizes achieved in the primary studies. Most studies included small samples for both the treatment and control groups, thus limiting the generalizability of transfer effects under study-specific interventions. Without any doubts, these studies also show large standard errors of effect sizes and low statistical power to detect transfer effects (Cohen, 1992; Hedges, 1982).

Fourth, from a substantive perspective, reasoning tests comprised several skills, including problem solving, intelligence, and memory. Although these skills might be distinct (Leighton & Sternberg, 2003; Ray, 1955), their measures are substantially correlated (e.g., Stadler, Becker, Gödker, Leutner, & Greiff, 2015). The current meta-analysis could not distinguish between them and therefore considered problem solving, intelligence, and memory facets of reasoning.

Fifth, the programming interventions presented in the primary studies varied considerably, for instance, from game design to pure programming activities. This variation, however, challenges the interpretation of an overall transfer effect and does not allow for drawing conclusion on what approach works best for fostering transfer (Barnett & Ceci, 2002). We therefore encourage empirical studies that are aimed at replicating the effectiveness of interventions for different content areas, contexts, and samples. From our perspective, it also remains for future research which factors determine the successful transfer of programming skills to other cognitive skills. These factors may well represent the conceptual underpinnings, contexts, and instructional approaches of experimental studies, next to the psychometric quality of cognitive skills tests (Mayer, 2015).

Considering that the above-mentioned issues, which are by no means unique to studies of the transferability of computer programming (Gagné et al., 1948; Melby-Lervåg et al.,

2016; Sala & Gobet, 2017a), we believe that an optimal study design includes both treated and untreated control groups, contains measures of both programming skills and other cognitive skills, and administers pre-, post-, and follow-up tests thereof, with a substantial time lag between post- and follow-up testing. These design characteristics allow researchers to provide better evidence for the existence of transfer effects.

Limitations and Future Directions

Considering the procedures applied in this meta-analysis, we emphasized the importance of selecting an appropriate model to estimate the transfer effect sizes in the presence of nested data. Testing different assumptions and approaches to describe the data, including the assumption of significant within- and between-study variances and the existence of random instead of fixed effects, is critical to the selection of meta-analytic models (Card, 2012). In our meta-analysis, the hierarchical data structure (i.e., effect sizes nested in studies) necessitated considering three-level meta-analysis and robust estimation procedures. Whenever both within- and between-study variation of effect sizes existed, three-level models were specified—if, however, one variance component was not significant, two-level models had to be specified. As we differentiated effect sizes across domains (i.e., overall transfer effects vs. specific effects on reasoning, programming, and other skills), this analytic approach resulted in different meta-analytic baseline models used to synthesize effect sizes (i.e., two-level fixed- and random-effects models *and* three-level random-effects models). Although this diversity threatens the comparability of baseline models and thus limits the possibilities of effect size comparisons across meta-analyses, a “one-ruler-fits-it-all”-decision with one common type of model applied to all types of transfer and cognitive skills could have created substantial bias in the estimation of variance components, especially in cases where substantial within-study variance across effect sizes exists (Wilson, Polanin, & Lipsey,

2016). We believe that further research is needed to explore ways of dealing with such situations.

Conclusion

Our meta-analysis showed that learning computer programming is associated with certain cognitive benefits. These benefits included gains in programming skills, pointing to the trainability of computational thinking through programming, and gains in other cognitive skills, pointing to the transferability of programming skills. In this meta-analysis, we could not confirm the doubts surrounding the far transfer of programming skills (e.g., Denning, 2017)—far transfer seems to exist for computer programming, yet not for all cognitive skills and not to the same extent. We encourage taking a differentiated perspective on the study of far transfer—a perspective that considers the diversity of cognitive skills and thus the differential effectiveness of programming interventions. The finding that learning computer programming aids some cognitive skills more than others is in line with previous meta-analyses that examined transfer effects of programming and underlines that far transfer is more likely to occur in situations that require cognitive skills close to programming. The conceptualization of programming as a key element of computational thinking and thus problem solving provides a suitable frame for explaining commonalities with other skills. We encourage researchers to carefully design transfer effect studies that address the methodological issues identified in the current body of research in several domains.

References

References marked with an asterisk indicate studies included in the meta-analysis.

- Adams, R. J., Smart, P., & Huff, A. S. (2017). Shades of grey: Guidelines for working with the grey literature in systematic reviews for management and organizational studies. *International Journal of Management Reviews*, 19(4), 432-454.
<http://dx.doi.org/10.1111/ijmr.12102>

- *Akcaoglu, M. (2013). *Cognitive and Motivational Impacts of Learning Game Design on Middle School Children* (Order No. 3587683, Doctoral Dissertation), Michigan State University, Ann Arbor. Retrieved from <https://search.proquest.com/docview/1427344597?accountid=14699> ProQuest Dissertations & Theses A&I database. (1427344597)
- *Akcaoglu, M., & Koehler, M. J. (2014). Cognitive outcomes from the Game-Design and Learning (GDL) after-school program. *Computers & Education*, 75, 72-81.
doi:10.1016/j.compedu.2014.02.003
- *Akdag, F. S. (1985). *The effects of computer programming on young children's learning*. (Order No. 8602964, Doctoral Dissertation), The Ohio State University, Ann Arbor. Retrieved from <https://search.proquest.com/docview/303418725?accountid=14699> ProQuest Dissertations & Theses A&I database. (303418725)
- Ambrósio, A., Pereira Júnior, C., & Georges, F. (2015). *Digital ink for cognitive assessment of computational thinking*. Paper presented at the IEEE Frontiers in Education Conference (FIE), Madrid, Spain.
- *Au, W. K., & Leung, J. P. (1991). Problem Solving, Instructional Methods and Logo Programming. *Journal of Educational Computing Research*, 7(4), 455-467.
doi:10.2190/K88Q-RWV1-AVPU-3DTK
- *Baker, S. H. (1987). *The effect of learning Logo on the problem-solving skills of elementary school children*. (Order No. 8602964, Doctoral Dissertation), University of Cincinnati, Ann Arbor. Retrieved from <https://search.proquest.com/docview/303548553?accountid=14699> ProQuest Dissertations & Theses A&I database. (303418725)

- *Barker, B. S., & Ansorge, J. (2007). Robotics as Means to Increase Achievement Scores in an Informal Learning Environment. *Journal of Research on Technology in Education*, 39(3), 229-243. doi:10.1080/15391523.2007.10782481
- Barnett, S. M., & Ceci, S. J. (2002). When and Where Do We Apply What We Learn? A Taxonomy for Far Transfer. *Psychological Bulletin*, 128(4), 612-637.
<http://dx.doi.org/10.1037//0033-2909.128.4.612>
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54. <http://dx.doi.org/10.1145/1929887.1929905>
- Bassok, M. (1990). Transfer of domain-specific problem-solving procedures. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16(3), 522-533.
<http://dx.doi.org/10.1037/0278-7393.16.3.522>
- Batista, A. L. F., Connolly, T. M., & Angotti, A. P. (2016). A Framework for Games-Based Construction Learning: A Text-Based Programming Languages Approach. Paper presented at the 10th European Conference on Games Based Learning, At Paisley, Scotland.
- *Battista, M. T., & Clements, D. H. (1986). The effects of Logo and CAI problem-solving environments on problem-solving abilities and mathematics achievement. *Computers in Human Behavior*, 2(3), 183-193. doi:10.1016/0747-5632(86)90002-6
- *Bebell, D. F. (1988). *Higher-level cognitive effects of Logo computer programming and problem-solving heuristics instruction*. (Order No. 8826276, Doctoral Dissertation), University of Denver, Ann Arbor. Retrieved from <https://search.proquest.com/docview/303716651?accountid=14699> ProQuest Dissertations & Theses A&I database. (303716651)

Bediou, B., Adams, D. M., Mayer, R. E., Tipton, E., Green, C. S., & Bavelier, D. (2018).

Meta-analysis of action video game impact on perceptual, attentional, and cognitive skills. *Psychological Bulletin*, 144(1), 77-110. <http://dx.doi.org/10.1037/bul0000130>

*Bernardo, M. A., & Morris, J. D. (1994). Transfer Effects of a High School Computer Programming Course on Mathematical Modeling, Procedural Comprehension, and Verbal Problem Solution. *Journal of Research on Computing in Education*, 26(4), 523-536. <http://dx.doi.org/10.1080/08886504.1994.10782108>

*Blackwelder, C. K. (1986). *Logo: A possible aid in the development of Piagetian formal reasoning (conceptual, cognitive growth)*. (Order No. 8703946, Doctoral Dissertations), Georgia State University, Ann Arbor. Retrieved from <https://search.proquest.com/docview/303475827?accountid=14699> ProQuest Dissertations & Theses A&I database. (303475827)

*Block, E. B., Simpson, D. L., & Reid, D. (1987). Teaching young children programming and word processing skills: The effects of three preparatory conditions. *Journal of Educational Computing Research*, 3(4), 435-442. <http://dx.doi.org/10.2190/TWKV-24AR-H12K-MFRL>

Borenstein, M., Hedges, L. V., Higgins, J. P., & Rothstein, H. R. (2009). *Introduction to meta-analysis*. Chichester, West Sussex: John Wiley & Sons, Ltd.

Bransford, J. D., & Schwartz, D. L. (1999). Rethinking Transfer: A Simple Proposal With Multiple Implications. *Review of Research in Education*, 24(1), 61-100. <http://dx.doi.org/10.3102/0091732x024001061>

Bransford, J. D., Vye, N., Stevens, R., Kuhl, P., Schwartz, D., Bell, P., . . . Reeves, B. (2005). Learning theories and education: Toward a decade of synergy. In P. Alexander & P. Winne (Eds.), *Handbook of Educational Psychology* (2 ed., pp. 209-244). Mahwah, N: Lawrence Erlbaum.

Bray, C. W. (1928). Transfer of learning. *Journal of Experimental Psychology*, 11(6), 443-467. <http://dx.doi.org/10.1037/h0071273>

*Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E., & Fontecchio, A. (2008). *Computer Aided Instruction As A Vehicle For Problem Solving: Scratch Boards In The Middle Years Classroom*. Paper presented at the Annual Conference & Exposition, Pittsburgh, Pennsylvania.

*Bruggeman, J. G. (1985). *The effects of modeling and inspection methods upon problem solving in a computer programming course*. (Doctoral Dissertation), Montana State University, Bozeman, Montana.

*Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing Mathematical Thinking with Scratch. In G. Conole, T. Klobučar, C. Rensing, J. Konert, & É. Lavoué (Eds.), *Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning, EC-TEL 2015, Toledo, Spain, September 15-18, 2015, Proceedings* (pp. 17-27). Cham: Springer International Publishing.

*Campbell, P. F., Fein, G. G., & Schwartz, S. S. (1991). The Effects of Logo Experience on First-Grade Children's Ability to Estimate Distance. *Journal of Educational Computing Research*, 7(3), 331-349. doi:10.2190/QJY1-9JEG-0UAY-J30D

Card, N. A. (2012). *Applied meta-analysis for social science research*. New York, NY: Guilford Press.

Carlson, K. D., & Schmidt, F. L. (1999). Impact of experimental design on effect size: Findings from the research literature on training. *Journal of Applied Psychology*, 84(6), 851-862. <http://dx.doi.org/10.1037/0021-9010.84.6.851>

- *Cathcart, W. G. (1990). Effects of Logo Instruction on Cognitive Style. *Journal of Educational Computing Research*, 6(2), 231-242. doi:10.2190/XNFC-RC25-FA6M-B33N
- *Chartier, D. (1996). An evaluation of two cognitive learning methods in adults on pre-qualification schemes: Logo and logical reasoning workshops (ARL). *European Journal of Psychology of Education*, 11(4), 443-457.
<http://dx.doi.org/10.1007/BF03173283>
- Chen, Z., & Klahr, D. (2008). Remote Transfer of Scientific-Reasoning and Problem-Solving Strategies in Children. In R. V. Kail (Ed.), *Advances in Child Development and Behavior* (Vol. 36, pp. 419-470): JAI.
- *Cheshire, F. D. (1981). *The effect of learning computer programming skills on developing cognitive abilities*. (Order No. 8117163, Doctoral Dissertation), Arizona State University, Ann Arbor. Retrieved from
<https://search.proquest.com/docview/303092565?accountid=14699> ProQuest Dissertations & Theses A&I database. (303092565)
- Cheung, M. W.-L. (2013). Implementing Restricted Maximum Likelihood Estimation in Structural Equation Models. *Structural Equation Modeling: A Multidisciplinary Journal*, 20(1), 157-167. <http://dx.doi.org/10.1080/10705511.2013.742404>
- Cheung, M. W.-L. (2014). Modeling dependent effect sizes with three-level meta-analyses: A structural equation modeling approach. *Psychological Methods*, 19(2), 211-229.
<http://dx.doi.org/10.1037/a0032968>
- Cheung, M. W.-L. (2015). *Meta-Analysis: A Structural Equation Modeling Approach*. Chichester, West Sussex: John Wiley & Sons, Ltd.
- Cheung, M. W.-L. (2018). *metaSEM: Meta-Analysis using Structural Equation Modeling* (Version 1.0.0). Retrieved from <https://github.com/mikewlcheung/metasem>

- *Choi, W. S. (1991). *Effect of Pascal and FORTRAN programming instruction on the problem-solving cognitive ability in formal operational stage students*. (Order No. 9129372, Doctoral Dissertation), Texas Tech University, Ann Arbor. Retrieved from <https://search.proquest.com/docview/303955348?accountid=14699> ProQuest Dissertations & Theses A&I database. (303955348)
- Clements, D. H. (1986a). Logo and cognition: A theoretical foundation. *Computers in Human Behavior*, 2(2), 95-110. [http://dx.doi.org/10.1016/0747-5632\(86\)90026-9](http://dx.doi.org/10.1016/0747-5632(86)90026-9)
- *Clements, D. H. (1986b). Effects of Logo and CAI environments on cognition and creativity. *Journal of Educational Psychology*, 78(4), 309-318. <http://dx.doi.org/10.1037/0022-0663.78.4.309>
- *Clements, D. H. (1987). Longitudinal Study of the Effects of Logo Programming on Cognitive Abilities and Achievement. *Journal of Educational Computing Research*, 3(1), 73-94. doi:10.2190/RCNV-2HYF-60CM-K7K7
- *Clements, D. H. (1990). Metacomponential development in a Logo programming environment. *Journal of Educational Psychology*, 82(1), 141-149. doi:10.1037/0022-0663.82.1.141
- *Clements, D. H. (1991). Enhancement of Creativity in Computer Environments. *American Educational Research Journal*, 28(1), 173-187. <http://dx.doi.org/10.3102/00028312028001173>
- Clements, D. H. (1995). Teaching creativity with computers. *Educational Psychology Review*, 7(2), 141-161. <http://dx.doi.org/10.1007/bf02212491>
- *Clements, D. H., & Battista, M. T. (1989). Learning of geometric concepts in a Logo environment. *Journal for Research in Mathematics Education*, 20(5), 450-467. <http://dx.doi.org/10.2307/749420>

- *Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76(6), 1051-1058.
<http://dx.doi.org/10.1037/0022-0663.76.6.1051>
- *Clements, D. H., & Nastasi, B. K. (1988). Social and Cognitive Interactions in Educational Computer Environments. *American Educational Research Journal*, 25(1), 87-106.
doi:10.3102/00028312025001087
- Clements, D. H., & Sarama, J. (1997). Research on Logo. *Computers in the Schools*, 14(1-2), 9-46. http://dx.doi.org/10.1300/J025v14n01_02
- Coburn, K. M., & Vevea, J. L. (2017). weightr: Estimating Weight-Function Models for Publication Bias. *R package version 1.1.2*.
- Cohen, J. (1992). A power primer. *Psychological Bulletin*, 112(1), 155-159.
<http://dx.doi.org/10.1037/0033-2909.112.1.155>
- Cooper, G., & Sweller, J. (1987). Effects of schema acquisition and rule automation on mathematical problem-solving transfer. *Journal of Educational Psychology*, 79(4), 347-362. <http://dx.doi.org/10.1037/0022-0663.79.4.347>
- *Cooper, S., Dann, W., & Pausch, R. (2003). Teaching objects-first in introductory computer science. *ACM SIGCSE Bulletin*, 35(1), 191-195. doi:10.1145/792548.611966
- Costa, J. M., & Miranda, G. L. (2017). Relation between Alice software and programming learning: a systematic review of the literature and meta-analysis. *British Journal of Educational Technology*, 48(6), 1464-1474. <http://dx.doi.org/10.1111/bjet.12496>
- *Dalton, D. W. (1986). *A Comparison of the Effects of LOGO Use and Teacher-Directed Problem-Solving Instruction on the Problem-Solving Skills, Achievement, and Attitudes of Low, Average, and High Achieving Junior High School Learners*. Paper presented at the Annual Convention of the Association for Educational Communications and Technology, Las Vegas, NV.

- *Dalton, D. W., & Goodrum, D. A. (1991). The Effects of Computer Programming on Problem-Solving Skills and Attitudes. *Journal of Educational Computing Research*, 7(4), 483-506. doi:10.2190/762V-KV6T-D3D1-KDY2
- *Degelman, D., Free, J. U., Scarlato, M., Blackburn, J. M., & Golden, T. (1986). Concept learning in preschool children: Effects of a short-term LOGO experience. *Journal of Educational Computing Research*, 2(2), 199-205. doi:10.2190/RH2K-4AQ7-2598-TVEA
- Denning, P. J. (2017). Remaining Trouble Spots with Computational Thinking. *Communications of the ACM*, 60(6), 33-39. <http://dx.doi.org/10.1145/2998438>
- *Dillashaw, F. G., & Bell, S. R. (1985, April 15-18, 1985). *Learning Outcomes of Computer Programming Instruction for Middle-Grades Students: A Pilot Study*. Paper presented at the Annual Meeting of the National Association for Research in Science Teaching, French Lick Springs, IN.
- Duval, S., & Tweedie, R. (2000). Trim and Fill: A Simple Funnel-Plot–Based Method of Testing and Adjusting for Publication Bias in Meta-Analysis. *Biometrics*, 56(2), 455-463. <http://dx.doi.org/10.1111/j.0006-341X.2000.00455.x>
- *Dziak, B. S. (1985). *Programming computer graphics and the development of concepts in geometry*. (Doctoral Dissertation), Ohio State University, Columbus, Ohio.
- Erol, O., & Kurt, A. A. (2017). The effects of teaching programming with scratch on pre-service information technology teachers' motivation and achievement. *Computers in Human Behavior*, 77, 11-18. <http://dx.doi.org/10.1016/j.chb.2017.08.017>
- *Fickel, M. G. (1986). *The effects of Logo study on problem-solving cognitive abilities of sixth-grade students*. (Order No. 8614452, Doctoral Dissertation), The University of Nebraska - Lincoln, Ann Arbor. Retrieved from

<https://search.proquest.com/docview/303442646?accountid=14699> ProQuest

Dissertations & Theses A&I database. (303442646)

Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American Psychologist*, 34(10), 906-911.

<http://dx.doi.org/10.1037/0003-066X.34.10.906>

*Flores, A. (1985). *Effect of computer programming on the learning of calculus concepts*.

(Order No. 8602996, Doctoral Dissertation), The Ohio State University, Ann Arbor.

Retrieved from <https://search.proquest.com/docview/303376565?accountid=14699>

ProQuest Dissertations & Theses A&I database. (303376565)

Flórez, F. B., 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.

<http://dx.doi.org/10.3102/0034654317710096>

Gagné, R. M., Foster, H., & Crowley, M. E. (1948). The measurement of transfer of training.

Psychological Bulletin, 45(2), 97-130. <http://dx.doi.org/10.1037/h0061154>

*Gallini, J. K. (1987). A Comparison of the Effects of Logo and a CAI Learning Environment on Skills Acquisition. *Journal of Educational Computing Research*, 3(4), 461-477.

doi:10.2190/JRXE-4CAG-P150-GQ8G

Garneli, V., Giannakos, M. N., & Chorianopoulos, K. (2015). *Computing Education in K-12*

Schools: A Review of the Literature. Paper presented at the IEEE Global Engineering

Education Conference (EDUCON), Tallinn, Estonia.

*Geva, E., & Cohen, R. (1987). *Transfer of spatial concepts from Logo to map-reading*.

Retrieved from Toronto, Ontario, Canada: <https://eric.ed.gov/?id=ED288608>

*Gibbon, L. W. (2007). *Effects of LEGO Mindstorms on convergent and divergent problem-solving and spatial abilities in fifth and sixth grade students*. (Doctoral Dissertation),

Seattle Pacific University, Seattle, WA. Retrieved from

<http://adsabs.harvard.edu/abs/2007PhDT.....49G>

*González, M. R. (2016). *Codigofabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas.*

(Doctoral Dissertation), Universidad Nacional de Educación a Distancia. Retrieved from <https://www.educacion.es/teseo/mostrarRef.do?ref=1313319>

Greeno, J. G., & Middle School Mathematics through Applications Project Group. (1998).

The situativity of knowing, learning, and research. *American Psychologist*, 53(1), 5-26. <http://dx.doi.org/10.1037/0003-066X.53.1.5>

Greiff, S., Wüstenberg, S., Csapó, B., Demetriou, A., Hautamäki, J., Graesser, A. C., & Martin, R. (2014). Domain-general problem solving skills and education in the 21st century. *Educational Research Review*, 13, 74-83.

<http://dx.doi.org/10.1016/j.edurev.2014.10.002>

Grover, S., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43.

<http://dx.doi.org/10.3102/0013189x12463051>

*Hamada, R. M. (1986). *The relationship between learning Logo and proficiency in mathematics.* (Order No. 8623535, Doctoral Dissertation), Columbia University, Ann Arbor. Retrieved from

<https://search.proquest.com/docview/303487782?accountid=14699> ProQuest Dissertations & Theses A&I database. (303487782)

*Hayes, J., & Stewart, I. (2016). Comparing the effects of derived relational training and computer coding on intellectual potential in school-age children. *British Journal of Educational Psychology*, 86(3), 397-411. <http://dx.doi.org/10.1111/bjep.12114>

Hedges, L. V. (1982). Estimation of Effect Size From a Series of Independent Experiments.

Psychological Bulletin, 92(2), 490-499. <http://dx.doi.org/10.1037/0033-2909.92.2.490>

Hennessy, S., & Amabile, T. (2010). Creativity. *Annual Review of Psychology*, 61, 569-598.

<http://dx.doi.org/10.1146/annurev.psych.093008.100416>

Higgins, J. P. T., & Thompson, S. G. (2002). Quantifying heterogeneity in a meta-analysis.

Statistics in Medicine, 21(11), 1539-1558. <http://dx.doi.org/10.1002/sim.1186>

Hutchison, A., Nadolny, L., & Estapa, A. (2016). Using Coding Apps to Support Literacy

Instruction and Develop Coding Literacy. *The Reading Teacher*, 69(5), 493-503.

<http://dx.doi.org/10.1002/trtr.1440>

Jacobse, A., & Harskamp, E. (2011). *A meta-analysis of the effects of instructional*

interventions on students' mathematics achievement. Retrieved from Groningen:

[https://www.rug.nl/research/portal/publications/a-metaanalysis-of-the-effects-of-](https://www.rug.nl/research/portal/publications/a-metaanalysis-of-the-effects-of-instructional-interventions-on-students-mathematics-achievement(1a0ea36d-3ca3-4639-9bb4-6fa220e50f38).html)

[instructional-interventions-on-students-mathematics-achievement\(1a0ea36d-3ca3-4639-9bb4-6fa220e50f38\).html](https://www.rug.nl/research/portal/publications/a-metaanalysis-of-the-effects-of-instructional-interventions-on-students-mathematics-achievement(1a0ea36d-3ca3-4639-9bb4-6fa220e50f38).html)

*Jenkins, C. (2015). Poem Generator: A Comparative Quantitative Evaluation of a

Microworlds-Based Learning Approach for Teaching English. *International Journal of Education and Development using Information and Communication Technology*, 11(2), 153-167.

*Johnson-Gentile, K., Clements, D. H., & Battista, M. T. (1994). Effects of Computer and Noncomputer Environments on Students' Conceptualizations of Geometric Motions.

Journal of Educational Computing Research, 11(2), 121-140.

<http://dx.doi.org/10.2190/49EE-8PXL-YY8C-A923>

*Julie, C. (1988). *Effects of an integrated computer and manipulative environment on preservice elementary teachers' understanding of novel mathematical concepts*. (Order No. 8815364, Doctoral Dissertation), University of Illinois at Urbana-Champaign,

Ann Arbor. Retrieved from

<https://search.proquest.com/docview/303689813?accountid=14699> ProQuest

Dissertations & Theses A&I database. (303689813)

Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. Cambridge, MA: The MIT Press.

Kafai, Y. B., & Burke, Q. (2015). Constructionist Gaming: Understanding the Benefits of Making Games for Learning. *Educational Psychologist, 50*(4), 313-334.
<http://dx.doi.org/10.1080/00461520.2015.1124022>

*Kapa, E. (1999). Problem solving, planning ability and sharing processes with Logo. *Journal of Computer Assisted Learning, 15*(1), 73-84. doi:10.1046/j.1365-2729.1999.151077.x

*Kazakoff, E. R., & Bers, M. (2012). Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia, 21*(4), 371-391.

*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. doi:10.1007/s10643-012-0554-5

*Kim, B., Kim, T., & Kim, J. (2013). Paper-and-Pencil Programming Strategy toward Computational Thinking for Non-Majors: Design Your Solution. *Journal of Educational Computing Research, 49*(4), 437-459.
<http://dx.doi.org/10.2190/EC.49.4.b>

*Kim, S., Chung, K., & Yu, H. (2013). Enhancing Digital Fluency through a Training Program for Creative Problem Solving Using Computer Programming. *The Journal of Creative Behavior, 47*(3), 171-199. <http://dx.doi.org/10.1002/jocb.30>

- *Kiser, S. S. (1989). *Logo programming, metacognitive skills in mathematical problem-solving, and mathematics achievement*. (Order No. 9032983, Doctoral Dissertation), The University of North Carolina at Chapel Hill, Ann Arbor. Retrieved from <https://search.proquest.com/docview/303791402?accountid=14699> ProQuest Dissertations & Theses A&I database. (303791402)
- *Koohang, A. A. (1984). *Traditional Method versus Computer-Aided Instruction Method in Teaching BASIC Programming to Vocational High School Students*. Retrieved from Carbondale, IL: <http://ovidsp.ovid.com/ovidweb.cgi?T=JS&CSC=Y&NEWS=N&PAGE=fulltext&D=eric2&AN=ED273247>
- *Kurland, D. M., Pea, R. D., Clement, C., & Mawby, R. (1986). A Study of the Development of Programming Ability and Thinking Skills in High School Students. *Journal of Educational Computing Research*, 2(4), 429-458. doi:10.2190/BKML-B1QV-KDN4-8ULH
- *Lai, A.-F., & Yang, S.-M. (2011). *The learning effect of visualized programming learning on 6 th graders' problem solving and logical reasoning abilities*. Paper presented at the International Conference on Electrical and Control Engineering (ICECE).
- *Lee, K. O. (1995). The effects of computer programming training on the cognitive development of 7-8 year-old children. *Korean Journal of Child Studies*, 16(1), 79-88.
- *Lehrer, R., & Randle, L. (1987). Problem Solving, Metacognition and Composition: The Effects of Interactive Software for First-Grade Children. *Journal of Educational Computing Research*, 3(4), 409-427. <http://dx.doi.org/10.2190/UFWW-FADF-BK21-YR5N>

- *Lehrer, R., Guckenberg, T., & Lee, O. (1988). Comparative study of the cognitive consequences of inquiry-based Logo instruction. *Journal of Educational Psychology*, 80(4), 543-553. doi:10.1037/0022-0663.80.4.543
- *Lehrer, R., Randle, L., & Sancilio, L. (1989). Learning Preproof Geometry With LOGO. *Cognition and Instruction*, 6(2), 159-184. doi:10.1207/s1532690xci0602_2
- Lehtonen, M., Soveri, A., Laine, A., Järvenpää, J., de Bruin, A., & Antfolk, J. (2018). Is bilingualism associated with enhanced executive functioning in adults? A meta-analytic review. *Psychological Bulletin*. <http://dx.doi.org/10.1037/bul0000142>
- Leighton, J. P., & Sternberg, R. J. (2003). Reasoning and Problem Solving. In I. B. Weiner (Ed.), *Handbook of Psychology*. Hoboken, NJ: John Wiley & Sons, Inc.
- *Lenamond, D. L. (1992). *A comparison of the effects of Lego TC Logo and problem-solving software on problem-solving skills*. (Order No. 1346643, Master Dissertation), University of Houston-Clear Lake, Ann Arbor. Retrieved from <https://search.proquest.com/docview/303997213?accountid=14699> ProQuest Dissertations & Theses A&I database. (303997213)
- Li, Q., & Ma, X. (2010). A Meta-analysis of the Effects of Computer Technology on School Students' Mathematics Learning. *Educational Psychology Review*, 22(3), 215–243. <http://dx.doi.org/10.1007/s10648-010-9125-8>
- Liao, Y.-k. C. (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, Canada.
- 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. <http://dx.doi.org/10.2190/e53g-hh8k-ajrr-k69m>
- Lipsey, M. W., & Wilson, D. (2001). *Practical meta-analysis*. Thousand Oaks, CA: Sage.

- Little, T. D., Jorgensen, T. D., Lang, K. M., & Moore, W. G. (2014). On the joys of missing data. *Journal of Pediatric Psychology, 39*(2), 151-162.
<http://dx.doi.org/10.1093/jpepsy/jst048>
- *Littlefield, J., Delclos, V. R., Bransford, J. D., Clayton, K. N., & Franks, J. J. (1989). Some prerequisites for teaching thinking: Methodological issues in the study of LOGO programming. *Cognition and Instruction, 6*(4), 331-366.
[doi:10.1207/s1532690xci0604_4](https://doi.org/10.1207/s1532690xci0604_4)
- Lobato, J. (2006). Alternative perspectives on the transfer of learning: History, issues, and challenges for future research. *The Journal of the Learning Sciences, 15*(4), 431-449.
http://dx.doi.org/10.1207/s15327809jls1504_1
- *Luckow, J. J. (1984). *The effects of studying logo turtle graphics on spatial ability*. (Doctoral Dissertation), Boston University, Boston, MA.
- 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. <http://dx.doi.org/10.1016/j.chb.2014.09.012>
- Ma, H.-H. (2006). A Synthetic Analysis of the Effectiveness of Single Components and Packages in Creativity Training Programs. *Creativity Research Journal, 18*(4), 435-446. http://dx.doi.org/10.1207/s15326934crj1804_3
- Ma, H.-H. (2009). The Effect Size of Variables Associated With Creativity: A Meta-Analysis. *Creativity Research Journal, 21*(1), 30-42.
<http://dx.doi.org/10.1080/10400410802633400>
- *Many, W. A., Lockard, J., Abrams, P. D., & Friker, W. (1988). The effect of learning to program in Logo on reasoning skills of junior high school students. *Journal of Educational Computing Research, 4*(2), 203-213. [doi:10.2190/TC38-RJT8-L241-DV9W](https://doi.org/10.2190/TC38-RJT8-L241-DV9W)

- Mayer, R. E. (2015). On the Need for Research Evidence to Guide the Design of Computer Games for Learning. *Educational Psychologist, 50*(4), 349-353.
<http://dx.doi.org/10.1080/00461520.2015.1133307>
- McCauley, R., Fitzgerald, S., Lewandowski, G., Murphy, L., Simon, B., Thomas, L., & Zander, C. (2008). Debugging: a review of the literature from an educational perspective. *Computer Science Education, 18*(2), 67-92.
<http://dx.doi.org/10.1080/08993400802114581>
- McGrew, K. S. (2009). CHC theory and the human cognitive abilities project: Standing on the shoulders of the giants of psychometric intelligence research. *Intelligence, 37*(1), 1-10.
<http://dx.doi.org/10.1016/j.intell.2008.08.004>
- Melby-Lervåg, M., Redick, T. S., & Hulme, C. (2016). Working Memory Training Does Not Improve Performance on Measures of Intelligence or Other Measures of "Far Transfer": Evidence From a Meta-Analytic Review. *Perspectives on Psychological Science, 11*(4), 512-534. <http://dx.doi.org/10.1177/1745691616635612>
- *Mevarech, Z. R., & Kramarski, B. (1992). How and how much can cooperative Logo environments enhance creativity and social relationships? *Learning and Instruction, 2*(3), 259-274. doi:10.1016/0959-4752(92)90012-B
- *Miller, E. A. (1985). *The use of Logo-Turtle graphics in a training program to enhance spatial visualization*. (Doctoral Dissertation), Concordia University, Québec, Canada.
- *Miller, G. E., & Emihovich, C. (1986). The effects of mediated programming instruction on preschool children's self-monitoring. *Journal of Educational Computing Research, 2*(3), 283-297. doi:10.2190/CEMM-LQHL-XN6D-1U15
- *Miller, R. B., Kelly, G. N., & Kelly, J. T. (1988). Effects of Logo computer programming experience on problem solving and spatial relations ability. *Contemporary Educational Psychology, 13*(4), 348-357. doi:10.1016/0361-476X(88)90034-3

*Milner, S. (1973). *The Effects of Computer Programming on Performance in Mathematics*.

Paper presented at the Annual meeting of the American Educational Research Association, New Orleans, Louisiana.

*Missiuna, C., & et al. (1987). *Development and Evaluation of the "Thinking with LOGO"*

Curriculum. Retrieved from Alberta, Canada:

<http://ovidsp.ovid.com/ovidweb.cgi?T=JS&CSC=Y&NEWS=N&PAGE=fulltext&D=eric2&AN=ED287453>

Moeyaert, M., Ugille, M., Natasha Beretvas, S., Ferron, J., Bunuan, R., & Van den Noortgate,

W. (2017). Methods for dealing with multiple outcomes in meta-analysis: a comparison between averaging effect sizes, robust variance estimation and multilevel meta-analysis. *International Journal of Social Research Methodology*, 20(6), 559-572.

<http://dx.doi.org/10.1080/13645579.2016.1252189>

Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & The PRISMA Group, et al. (2009).

Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement. *PLOS Medicine*, 6(7), 1-6. <http://dx.doi.org/10.1371/journal.pmed.1000097>

Moreno-Leon, J., & Robles, G. (2016). *Code to learn with Scratch? A systematic literature review*. Paper presented at the 2016 IEEE Global Engineering Education Conference (EDUCON).

*Moreno-León, J., Robles, G., & Román-González, M. (2016). Code to Learn: Where Does It

Belong in the K-12 Curriculum? *Journal of Information Technology Education: Research*, 15, 283-303.

Morris, S. B. (2008). Estimating Effect Sizes from Pretest-Posttest-Control Group Designs.

Organizational Research Methods, 11(2), 364-386. doi: 10.1177/1094428106291059

- *Mynatt, B. T., Smith, K. H., Kamouri, A. L., & Tykodi, T. A. (1986). Which way to computer literacy, programming or applications experience? *International Journal of Man-Machine Studies*, 25(5), 557-572. doi:10.1016/S0020-7373(86)80023-2
- Naragon-Gainey, K., McMahon, T. P., & Chacko, T. P. (2017). The structure of common emotion regulation strategies: A meta-analytic examination. *Psychological Bulletin*, 143(4), 384-427. <http://dx.doi.org/10.1037/bul0000093>
- *Nastasi, B. K., & Clements, D. H. (1992). Social-cognitive behaviors and higher-order thinking in educational computer environments. *Learning and Instruction*, 2(3), 215-238. doi:10.1016/0959-4752(92)90010-J
- *Nastasi, B. K., Clements, D. H., & Battista, M. T. (1990). Social-cognitive interactions, motivation, and cognitive growth in Logo programming and CAI problem-solving environments. *Journal of Educational Psychology*, 82(1), 150-158. doi:10.1037/0022-0663.82.1.150
- *Nugent, G., Barker, B., Grandgenett, N., & Adamchuk, V. I. (2010). Impact of Robotics and Geospatial Technology Interventions on Youth STEM Learning and Attitudes. *Journal of Research on Technology in Education*, 42(4), 391-408. <http://dx.doi.org/10.1080/15391523.2010.10782557>
- *Olson, J. K. (1985). *Using Logo to supplement the teaching of geometric concepts in the elementary school classroom*. (Doctoral Dissertation), Oklahoma State University. Retrieved from <https://shareok.org/handle/11244/17733>
- *Oprea, J. M. (1984). *The effects of computer programming on a student's mathematical generalization and understanding of variables*. (Order No. 8504061, Doctoral Dissertation), The Ohio State University, Ann Arbor. Retrieved from <https://search.proquest.com/docview/303317313?accountid=14699> ProQuest Dissertations & Theses A&I database. (303317313)

- *Ortiz, A. M. (2015). Examining students' proportional reasoning strategy levels as evidence of the impact of an integrated LEGO robotics and mathematics learning experience. *Journal of Technology Education*, 26(2), 46-69.
<http://dx.doi.org/10.21061/jte.v26i2.a.3>
- *Ortiz, E. (1987). *A comparison of a computer programming approach to a textbook approach in teaching the mathematics concept "variable" to sixth graders*. (Order No. 8728210, Doctoral Dissertation), Louisiana State University and Agricultural & Mechanical College, Ann Arbor. Retrieved from
<https://search.proquest.com/docview/303605676?accountid=14699> ProQuest Dissertations & Theses A&I database. (303605676)
- *Owston, R., Wideman, H., Ronda, N. S., & Brown, C. (2009). Computer game development as a literacy activity. *Computers & Education*, 53(3), 977-989.
<http://dx.doi.org/10.1016/j.compedu.2009.05.015>
- Palumbo, D. B. (1990). Programming Language/Problem-Solving Research: A Review of Relevant Issues. *Review of Educational Research*, 60(1), 65-89.
<http://dx.doi.org/10.3102/00346543060001065>
- *Palumbo, D. B., & Michael Reed, W. (1991). The Effect of BASIC Programming Language Instruction on High School Students' Problem Solving Ability and Computer Anxiety. *Journal of Research on Computing in Education*, 23(3), 343-372.
doi:10.1080/08886504.1991.10781967
- *Papaevripidou, M., Constantinou, C. P., & Zacharia, Z. C. (2007). Modeling complex marine ecosystems: an investigation of two teaching approaches with fifth graders. *Journal of Computer Assisted Learning*, 23(2), 145-157. doi:10.1111/j.1365-2729.2006.00217.x

- *Pardamean, B., Evelin, & Honni. (2011). *The effect of Logo programming language for creativity and problem solving*. Paper presented at the Proceedings of the 10th WSEAS international conference on E-Activities, Jakarta, Indonesia.
- *Pardamean, B., Suparyanto, T., & Evelin. (2015). Improving Problem-Solving Skills through Logo Programming Language. *New Educational Review*, 41(3), 52-64.
doi:10.15804/ner.2015.41.3.04
- *Park, J. (2015). Effect of Robotics enhanced inquiry based learning in elementary Science education in South Korea. *Journal of Computers in Mathematics and Science Teaching*, 34(1), 71-95.
- Pastor, D. A., & Lazowski, R. A. (2018). On the Multilevel Nature of Meta-Analysis: A Tutorial, Comparison of Software Programs, and Discussion of Analytic Choices. *Multivariate Behavioral Research*, 53(1), 74-89.
<http://dx.doi.org/10.1080/00273171.2017.1365684>
- *Pea, R. D., & Kurland, D. M. (1984). *Logo Programming and the Development of Planning Skills*. Retrieved from
<http://ovidsp.ovid.com/ovidweb.cgi?T=JS&CSC=Y&NEWS=N&PAGE=fulltext&D=eric2&AN=ED249930>
- Pea, R. D., & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New Ideas in Psychology*, 2(2), 137-168.
[http://dx.doi.org/10.1016/0732-118X\(84\)90018-7](http://dx.doi.org/10.1016/0732-118X(84)90018-7)
- Perkins, D. N., & Salomon, G. (1992). Transfer of learning. In T. N. Postlethwaite & T. Husen (Eds.), *International encyclopedia of education* (Vol. 2, pp. 6452-6457). Oxford, UK: Pergamon Press.

- Polanin, J. R., Tanner-Smith, E. E., & Hennessy, E. A. (2016). Estimating the Difference Between Published and Unpublished Effect Sizes: A Meta-Review. *Review of Educational Research, 86*(1), 207-236. <http://dx.doi.org/10.3102/0034654315582067>
- *Pollock, M. L. (1997). *Facilitating cognitive abilities and positive school attitudes among elementary school students through Lego-Logo programming*. (Order No. 9737665, Doctoral Dissertation), North Carolina State University, Ann Arbor. Retrieved from <https://search.proquest.com/docview/304369072?accountid=14699> ProQuest Dissertations & Theses A&I database. (304369072)
- *Poulin-DuBois, D., McGilly, C. A., & Shultz, T. R. (1989). Psychology of Computer Use: X. Effect of Learning Logo on Children's Problem-Solving Skills. *Psychological Reports, 64*(3), 1327-1337. doi:10.2466/pr0.1989.64.3c.1327
- *Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science, 45*(5), 583-602. <http://dx.doi.org/10.1007/s11251-017-9421-5>
- Ray, W. S. (1955). Complex tasks for use in human problem-solving research. *Psychological Bulletin, 52*(2), 134-149. <http://dx.doi.org/10.1037/h0044763>
- *Reding, A. H. (1981). *The effects of computer programming on problem solving abilities of fifth grade students*. (Order No. 8201793, Doctoral Dissertation), University of Wyoming, Ann Arbor. Retrieved from <https://search.proquest.com/docview/303188829?accountid=14699> ProQuest Dissertations & Theses A&I database. (303188829)
- *Reeder, L. K., & Leming, J. S. (1994). The Effect of Logo on the Nonverbal Reasoning Ability of Rural and Disadvantaged Third Graders. *Journal of Research on Computing in Education, 26*(4), 558-564. doi:10.1080/08886504.1994.10782111

- 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. <http://dx.doi.org/10.1016/j.chb.2016.08.047>
- *Rose, N. S. (1984). *Effects of learning computer programming on the general problem-solving abilities of fifth grade students*. (Doctoral Dissertation), North Texas State University, Denton, TX. Retrieved from <https://digital.library.unt.edu/ark:/67531/metadc331886/>
- Rosenberg, M. S. (2005). The file-drawer problem revisited: A general weighted method for calculating fail-safe numbers in meta-analysis. *Evolution, 59*(2), 464-468. <http://dx.doi.org/10.1111/j.0014-3820.2005.tb01004.x>
- *Rucinski, T. T. (1986). *The effects of computer programming on the problem-solving strategies of preservice teachers*. (Order No. 8701602, Doctoral Dissertation), University of Illinois at Urbana-Champaign, Ann Arbor. Retrieved from <https://search.proquest.com/docview/303417189?accountid=14699> ProQuest Dissertations & Theses A&I database. (303417189)
- Sala, G., & Gobet, F. (2016). Do the benefits of chess instruction transfer to academic and cognitive skills? A meta-analysis. *Educational Research Review, 18*, 46-57. <http://dx.doi.org/10.1016/j.edurev.2016.02.002>
- Sala, G., & Gobet, F. (2016). Do the benefits of chess instruction transfer to academic and cognitive skills? A meta-analysis. *Educational Research Review, 18*, 46-57. doi:10.1016/j.edurev.2016.02.002
- Sala, G., & Gobet, F. (2017a). Does Far Transfer Exist? Negative Evidence From Chess, Music, and Working Memory Training. *Current Directions in Psychological Science, 26*(6), 515-520. <http://dx.doi.org/10.1177/0963721417712760>

- Sala, G., & Gobet, F. (2017b). When the music's over. Does music skill transfer to children's and young adolescents' cognitive and academic skills? A meta-analysis. *Educational Research Review*, 20, 55-67. <http://dx.doi.org/10.1016/j.edurev.2016.11.005>
- Sala, G., Tatlidil, K. S., & Gobet, F. (2018). Video game training does not enhance cognitive ability: A comprehensive meta-analytic investigation. *Psychological Bulletin*, 144(2), 111-139. <http://dx.doi.org/10.1037/bul0000139>
- Salleh, S. M., Shukur, Z., & Judi, H. M. (2013). Analysis of Research in Programming Teaching Tools: An Initial Review. *Procedia - Social and Behavioral Sciences*, 103, 127-135. <http://dx.doi.org/10.1016/j.sbspro.2013.10.317>
- Salomon, G., & Perkins, D. N. (1987). Transfer of cognitive skills from programming: When and how? *Journal of Educational Computing Research*, 3(2), 149-169. <https://doi.org/10.2190/6F4Q-7861-QWA5-8PL1>
- Scherer, R. (2016). Learning from the past – The need for empirical evidence on the transfer effects of computer programming skills. *Frontiers in Psychology*, 7(1390). <http://dx.doi.org/10.3389/fpsyg.2016.01390>
- Schmidt, F. L., & Hunter, J. E. (2014). *Methods of meta-analysis: Correcting error and bias in research findings* (3 ed.). Thousand Oaks, CA: Sage.
- Schmucker, C. M., Blümle, A., Schell, L. K., Schwarzer, G., Oeller, P., Cabrera, L., . . . on behalf of the, O. c. (2017). Systematic review finds that study data not published in full text articles have unclear impact on meta-analyses results in medical research. *PLoS One*, 12(4), 1-16. <http://dx.doi.org/10.1371/journal.pone.0176210>
- Schunk, D. H. (2012). *Learning theories: An educational perspective* (6th ed.). Boston, MA: Pearson Education.

Scott, G., Leritz, L. E., & Mumford, M. D. (2004). The effectiveness of creativity training: A quantitative review. *Creativity Research Journal*, 16(4), 361-388.

<http://dx.doi.org/10.1080/10400410409534549>

*Seidman, R. H. (1981). *The Effects of Learning a Computer Programming Language on the Logical Reasoning of School Children*. Paper presented at the Annual Meeting of the American Educational Research Association, Los Angeles, CA.

*Seo, Y.-H., & Kim, J.-H. (2016). Analyzing the Effects of Coding Education through Pair Programming for the Computational Thinking and Creativity of Elementary School Students. *Indian Journal of Science and Technology*, 9(46).

<http://dx.doi.org/10.17485/ijst/2016/v9i46/107837>

*Shaw, D. G. (1986). Effects of Learning to Program A Computer in BASIC or Logo on Problem-solving Abilities. *AEDS Journal*, 19(2-3), 176-189.

doi:10.1080/00011037.1986.11008422

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking.

Educational Research Review, 22, 142-158.

<http://dx.doi.org/10.1016/j.edurev.2017.09.003>

Simonsohn, U., Nelson, L. D., & Simmons, J. P. (2014). P-Curve: A Key to the File-Drawer.

Journal of Experimental Psychology: General, 143(2), 534-547.

<http://dx.doi.org/10.1037/a0033242>

Simonsohn, U., Nelson, L. D., & Simmons, J. P. (2017). P-curve *Online App Version 4.06*.

<http://www.p-curve.com/app4/>

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling* (2 ed.). London: Sage.

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling* (2 ed.). London: Sage.

- Stadler, M., Becker, N., Gödker, M., Leutner, D., & Greiff, S. (2015). Complex problem solving and intelligence: A meta-analysis. *Intelligence, 53*, 92-101.
<http://dx.doi.org/10.1016/j.intell.2015.09.005>
- Sternberg, R. J. (1982). Reasoning, problem solving, and intelligence. In R. J. Sternberg (Ed.), *Handbook of Human Intelligence* (pp. 225-307). New York, NY: Cambridge University Press.
- Stockard, J., Wood, T. W., Coughlin, C., & Khoury, C. R. (2018). The Effectiveness of Direct Instruction Curricula: A Meta-Analysis of a Half Century of Research. *Review of Educational Research*. <http://dx.doi.org/10.3102/0034654317751919>
- *Subhi, T. (1999). The impact of LOGO on gifted children's achievement and creativity. *Journal of Computer Assisted Learning, 15*(2), 98-108. doi:10.1046/j.1365-2729.1999.152082.x
- Sung, Y.-T., Chang, K.-E., & Liu, T.-C. (2016). The effects of integrating mobile devices with teaching and learning on students' learning performance: A meta-analysis and research synthesis. *Computers & Education, 94*, 252-275.
doi:10.1016/j.compedu.2015.11.008
- *Swan, K. (1991). Programming objects to think with: LOGO and the teaching and learning of problem solving. *Journal of Educational Computing Research, 7*(1), 89-112.
doi:10.2190/UX0M-NHM2-1G5X-01X4
- *Swan, K., & Black, J. B. (1990). *Logo programming, problem solving, and knowledge-based instruction*. Paper presented at the Annual Meeting of the American Educational Research Association, Boston, MA.
- *Taitt, N. P. (1985). *The effect of computer programming instruction on the problem solving ability of pre-service elementary teachers*. (Order No. 8521891, Doctoral Dissertation), University of Illinois at Urbana-Champaign, Ann Arbor. Retrieved from

<https://search.proquest.com/docview/303345336?accountid=14699> ProQuest

Dissertations & Theses A&I database. (303345336)

- *Teahan, M. (2001). *The effect of an introductory course in QBasic programming on the mathematical problem-solving skills of transition year students and on their attitudes to mathematics*. (Doctoral Dissertation), Dublin City University, Dublin. Retrieved from computing.dcu.ie/wpapers/MCE/2001/0501.ps
- *Thompson, A. D., & Chen Wang, H.-m. (1988). Effects of a Logo microworld on student ability to transfer a concept. *Journal of Educational Computing Research*, 4(3), 335-347. doi:10.2190/1U7L-33HQ-R2R1-6DCF
- Tsai, K. C. (2013). A Review of the Effectiveness of Creative Training on Adult Learners. *Journal of Social Science Studies*, 1(1), 17. <http://dx.doi.org/10.5296/jsss.v1i1.4329>
- *Tsuei, M. (1998). *The effects of Logo programming and multimedia software on fifth-grade students' creativity in Taiwan*. (Doctoral Dissertation), The University of Texas at Austin, Austin, TX. Retrieved from <https://www.learntechlib.org/p/129241>
- *Turner, S. V., & Land, M. L. (1988). Cognitive Effects of a Logo-Enriched Mathematics Program for Middle School Students. *Journal of Educational Computing Research*, 4(4), 443-452. doi:10.2190/R0XH-2VDD-4VFA-YB3H
- Umapathy, K., & Ritzhaupt, A. D. (2017). A Meta-Analysis of Pair-Programming in Computer Programming Courses: Implications for Educational Practice. *ACM Transactions on Computing Education*, 17(4), 1-13. <http://dx.doi.org/10.1145/2996201>
- Uribe, D., Klein, J. D., & Sullivan, H. (2003). The effect of computer-mediated collaborative learning on solving ill-defined problems. *Educational Technology Research & Development*, 51(1), 5-19. <http://dx.doi.org/10.1007/BF02504514>

- Uttal, D. H., Meadow, N. G., Tipton, E., Hand, L. L., Alden, A. R., Warren, C., & Newcombe, N. S. (2013). The malleability of spatial skills: a meta-analysis of training studies. *Psychological Bulletin*, *139*(2), 352-402. <http://dx.doi.org/10.1037/a0028446>
- Van den Noortgate, W., López-López, J. A., Marín-Martínez, F., & Sánchez-Meca, J. (2013). Three-level meta-analysis of dependent effect sizes. *Behavior Research Methods*, *45*(2), 576-594. <http://dx.doi.org/10.3758/s13428-012-0261-6>
- *VanLengen, C., & Maddux, C. (1990). Does instruction in computer programming improve problem solving ability? *Journal of IS Education*, *12*(2), 11-16.
- Veroniki, A. A., Jackson, D., Viechtbauer, W., Bender, R., Bowden, J., Knapp, G., . . . Salanti, G. (2016). Methods to estimate the between-study variance and its uncertainty in meta-analysis. *Research Synthesis Methods*, *7*(1), 55-79. <http://dx.doi.org/10.1002/jrsm.1164>
- Vevea, J. L., & Hedges, L. V. (1995). A general linear model for estimating effect size in the presence of publication bias. *Psychometrika*, *60*(3), 419-435. <http://dx.doi.org/10.1007/bf02294384>
- Viechtbauer, W. (2017). metafor: Meta-Analysis Package for R. *R package version 2.0-0*. Retrieved from <https://cran.r-project.org/web/packages/metafor/index.html>
- Viechtbauer, W., & Cheung, M. W.-L. (2010). Outlier and influence diagnostics for meta-analysis. *Research Synthesis Methods*, *1*(2), 112-125. <http://dx.doi.org/10.1002/jrsm.11>
- Vihavainen, A., Airaksinen, J., & Watson, C. (2014). *A systematic review of approaches for teaching introductory programming and their influence on success*. Paper presented at the 2014 Tenth Annual Conference on International Computing Education Research (ICER), Glasgow, Scotland.

- Voss, J. F., Wiley, J., & Carretero, M. (1995). Acquiring Intellectual Skills. *Annual Review of Psychology*, 46(1), 155-181. <http://dx.doi.org/10.1146/annurev.ps.46.020195.001103>
- *Weaver, C. L. (1991). *Young children learn geometric and spatial concepts using Logo with a screen turtle and a floor turtle*. (Order No. 9135151, Doctoral Dissertation), State University of New York at Buffalo, Ann Arbor. Retrieved from <https://search.proquest.com/docview/303972489?accountid=14699> ProQuest Dissertations & Theses A&I database. (303972489)
- Wilson, S. J., Polanin, J. R., & Lipsey, M. W. (2016). Fitting meta-analytic structural equation models with complex datasets. *Research Synthesis Methods*, 7(2), 121-139. <http://dx.doi.org/10.1002/jrsm.1199>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <http://dx.doi.org/10.1145/1118178.1118215>
- Woodworth, R. S., & Thorndike, E. L. (1901). The influence of improvement in one mental function upon the efficiency of other functions. (I). *Psychological Review*, 8(3), 247-261. <http://dx.doi.org/10.1037/h0074898>
- Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational Thinking as an Emerging Competence Domain. In M. Mulder (Ed.), *Competence-based Vocational and Professional Education: Bridging the Worlds of Work and Education* (pp. 1051-1067). Cham: Springer International Publishing.
- *Yi, B. J., & Eu, L. K. (2016). Effects of using Logo on pupils' learning in two-dimensional spaces. *Malaysian Online Journal of Educational Technology*, 4(3), 27-36.
- *Yoder, V. A. (1988). *Exploration of the interaction of the van Hiele levels of thinking with Logo and geometry understandings in preservice elementary teachers*. (Order No. 8826845, Doctoral Dissertation), The Pennsylvania State University, Ann Arbor.

Retrieved from <https://search.proquest.com/docview/303722431?accountid=14699>

ProQuest Dissertations & Theses A&I database. (303722431)

*Yusuf, M. M. (1995). The Effects of Logo-Based Instruction. *Journal of Educational Computing Research*, 12(4), 335-362. doi:10.2190/NNLP-MN2R-M6CV-9EJF

Tables

Table 1

Types of Cognitive Skills Measured in Primary Studies

<i>Cognitive skills</i>	<i>Facets of these skills</i>	<i>Examples measures and references</i>
Programming skills	<ul style="list-style-type: none"> ▪ Programming skills (including creating, modifying, and evaluating programming code) ▪ Programming knowledge (including procedural and conceptual knowledge) 	<p>Logo Knowledge Test (syntactic, semantic, schematic, and strategic programming knowledge; Lehrer, Lee, & Jong, 1999)</p> <p>Computational Thinking Test (Jenkins, 2015)</p> <p>Logo Criterion Task (Block, Simpson, & Reid, 1987)</p>
Reasoning	<ul style="list-style-type: none"> ▪ Intelligence, attention, perception, and memory ▪ Problem solving ▪ Critical thinking 	<p>Cornell Critical Thinking Test (Psycharis & Kallia, 2017)</p> <p>Developing Cognitive Abilities Test (Rose, 1984)</p> <p>Ross Test of Higher Cognitive Processes (Bebell, 1988)</p> <p>Group Assessment of Logical Thinking (Kim, Kim, & Kim, 2013)</p>
Creative thinking	<ul style="list-style-type: none"> ▪ Flexibility ▪ Fluency ▪ Elaboration ▪ Originality 	<p>Torrance Test of Creative Thinking (Seo & Kim, 2016; Clements, 1991)</p>
Spatial skills	<ul style="list-style-type: none"> ▪ Spatial understanding and reasoning ▪ Spatial memory 	<p>Spatial Aptitude Test (Chartier, 1996)</p> <p>Eliot-Price Spatial Test (Miller, 1985)</p>
Metacognition	Overall metacognitive skills	Assessment of Metacognitive Skills (Clements, 1986b)

		Awareness of Comprehension Failure Measure (Clements & Gullo, 1984)
		Metacognitive Components of Problem Solving (Lehrer & Randle, 1987)
Mathematical skills	<ul style="list-style-type: none"> ▪ Mathematics achievement, modeling, and problem solving ▪ Knowledge about mathematical concepts 	<p>Wide Range Achievement Test (Clements, 1986b)</p> <p>California Achievement Test (Bernardo & Morris, 1994)</p> <p>Mathematical Proportional Reasoning Test (Ortiz, 2015)</p> <p>Geometry Achievement Test (Johnson-Gentile, Clements, & Battista, 1994)</p>
Literacy	<ul style="list-style-type: none"> ▪ General language skills and spelling ▪ Reading ▪ Writing 	<p>Group Reading Assessment and Diagnostic Evaluation, Student Writing Test (Owston et al., 2009)</p> <p>New York State Holistic Writing Assessment (Lehrer & Randle, 1987)</p>
School achievement	School achievement in domains other than mathematics and literacy (e.g., Engineering and Social Sciences)	<p>Program Criterion Reference Test (Dalton, 1986)</p> <p>Science Achievement Test (Park, 2015)</p> <p>Engineering Achievement Test (Nugent et al., 2010)</p>

Table 2

Summary of Study Design, Sample, and Publication Characteristics ($m = 105$, $k = 539$)

Characteristics	m	k	Proportion of studies ^b	Proportion of effect sizes
<i>Study design and sample characteristics</i>				
Statistical study design (coded at the level of effect sizes)				
Pretest-posttest control group design	77	363	-	67.3 %
Posttest-only design	34	176	-	32.7 %
Statistical study design (coded at the level of studies)				
Pretest-posttest control group design	72	335	68.6 %	62.2 %
Posttest-only design	27	117	25.7 %	21.7 %
Mixed design	6	87	5.7 %	16.1 %
Randomization				
Random group assignment	47	304	44.8 %	56.4 %
Non-random group assignment	58	235	55.2 %	43.6 %
Treatment of control group(s) (coded at the level of effect sizes)				
Treated controls	41	163	-	30.2 %
Untreated controls	80	376	-	69.8 %
Treatment of control group(s) (coded at the level of studies)				
Treated controls	25	88	23.8 %	16.3 %
Untreated controls	64	268	61.0 %	49.7 %
Mixed groups	16	183	15.2 %	34.0 %
Matching [#]				
Matched	29	158	27.6 %	29.3 %
Not matched	71	365	67.6 %	67.7 %
Student collaboration [#]				
Collaboration	49	303	46.7 %	56.2 %
No collaboration	21	99	20.0 %	18.4 %
Programming tool [#]				
Visual tool	85	467	81.0 %	86.6 %
Text-based tool	16	55	15.2 %	10.2 %
Programming context				
Regular school instruction	89	446	84.8 %	82.7 %
Extra-curricular activities	16	93	15.2 %	17.3 %
Type of outcome measure				
Standardized test	58	288	-	53.4 %
Unstandardized test	56	251	-	46.6 %
Tests developed by researchers	52	235	-	43.6 %
Tests developed by teachers	6	16	-	3.0 %

Cognitive skills measures				
Programming	13	19	-	3.5 %
Reasoning	59	229	-	42.5 %
Creative thinking	10	77	-	14.3 %
Metacognition	10	48	-	8.9 %
Spatial skills	19	38	-	7.1 %
Mathematical skills	36	102	-	18.9 %
Literacy	9	19	-	3.5 %
School achievement	6	7	-	1.3 %
Type of transfer				
Near transfer only	3	3	2.9 %	0.6 %
Far transfer only	92	490	87.6 %	90.9 %
Near and far transfer	10	46	9.5 %	8.5 %
Educational level ^a				
Kindergarten	7	22	6.4 %	4.1 %
Primary school	68	390	61.8 %	72.4 %
Secondary school	23	112	20.9 %	20.8 %
College and university	12	15	10.9 %	2.8 %
Average age of students [#]				
5-10 years	25	191	23.8 %	35.4 %
11-15 years	9	31	8.6 %	5.8 %
16-20 years	2	6	1.9 %	1.1 %
> 20 years	2	8	1.9 %	1.5 %
Location of the study sample				
Asia	15	76	14.2 %	14.1 %
Europe	8	52	7.5 %	9.6 %
North America	80	403	75.5 %	74.8 %
South America	3	8	2.8 %	1.5 %
<i>Publication characteristics</i>				
Publication status				
Published	62	355	59.0 %	65.9 %
Grey literature	43	184	41.0 %	34.1 %
Publication year				
1970-1979	1	2	1.0 %	0.4 %
1980-1989	51	297	48.6 %	55.1 %
1990-1999	26	144	24.8 %	26.7 %
2000-2009	7	17	6.7 %	3.2 %
2010-2017	20	79	19.0 %	14.7 %

Note. m = Number of studies, k = Number of effect sizes.

^a The overall number of effect sizes may exceed $k = 539$, because some studies contained samples from different educational levels.

^b Some of these proportions are not provided here, because certain characteristics were considered effect size rather than study characteristics.

[#] Missing data in moderator variables occurred.

Table 3

Categorical Moderator Analyses of the Overall Transfer Effects ($m = 105$, $k = 539$)

<i>Moderator variables</i>	<i>m</i>	<i>k</i>	<i>g</i>	95 % CI	$Q_M(df)$	<i>p</i>	$Q_E(df)$	<i>p</i>	R_2^2	R_3^2
Study characteristics										
Study design (<i>coded at the level of effect sizes</i>)										
Pretest-posttest control group design	77	363	0.50	[0.37, 0.63]	0.07 (1)	.79	2958.06 (537)	< .001	0.0 %	0.0 %
Posttest-only design	34	176	0.47	[0.30, 0.65]						
Study design (<i>coded at the level of study</i>)										
Pretest-posttest control group design	72	335	0.46	[0.31, 0.60]	1.03 (2)	.60	- ^b	- ^b	0.0 %	0.0 %
Posttest-only design	27	117	0.53	[0.30, 0.77]						
Mixed design	6	87	0.69	[0.23, 1.16]						
Randomization										
Random group assignment	47	304	0.56	[0.38, 0.73]	1.04 (1)	.31	2877.23 (537)	< .001	0.0 %	1.1 %
Non-random group assignment	58	235	0.44	[0.27, 0.60]						
Treatment of control group(s) (<i>coded at the level of effect sizes</i>)										
Treated controls	41	163	0.16	[0.01, 0.33]	40.12 (1)	< .001	2897.38 (537)	< .001	16.7 %	0.0 %
Untreated controls	80	376	0.65	[0.51, 0.78]						
Treatment of control group(s) (<i>coded at the level of studies</i>)										
Treated controls	25	88	0.58	[0.33, 0.83]	2.24 (2)	.33	- ^b	- ^b	0.0 %	0.0 %
Untreated controls	64	268	0.51	[0.36, 0.67]						

Mixed controls	16	183	0.31	[0.03, 0.59]						
Type of transfer										
Near transfer only	3	3	1.44	[0.53, 2.35]	4.26 (2)	.12	- ^b	- ^b	0.0 %	0.6 %
Far transfer only	92	490	0.48	[0.35, 0.60]						
Near and far transfer	10	46	0.47	[0.10, 0.84]						
Matching [#]										
Matched	29	158	0.43	[0.21, 0.65]	0.58 (1)	.45	2922.38 (522)	< .001	0.0 %	0.0 %
Not matched	71	365	0.53	[0.39, 0.68]						
Student collaboration [#]										
Collaboration	49	303	0.55	[0.37, 0.74]	0.34 (1)	.56	2376.05 (400)	< .001	0.0 %	0.0 %
No collaboration	21	99	0.45	[0.16, 0.74]						
Programming tool [#]										
Visual tool	85	467	0.52	[0.38, 0.65]	0.50 (1)	.48	2966.95 (520)	< .001	0.0 %	0.0 %
Text-based tool	16	55	0.40	[0.08, 0.71]						
Study context										
Regular lessons	89	446	0.47	[0.34, 0.60]	0.98 (1)	.32	2931.26 (537)	< .001	0.2 %	0.0 %
Extracurricular activity	16	93	0.63	[0.33, 0.93]						
Type of outcome measure										
Standardized test	58	288	0.42	[0.28, 0.57]	2.59 (1)	.11	2964.88 (537)	< .001	0.0 %	3.9 %
Unstandardized test	56	251	0.57	[0.42, 0.71]						
Sample characteristics										
Educational level ^a										
Kindergarten	7	22	0.50	[0.01, 1.00]	2.21 (3)	.55	2958.63 (535)	< .001	0.0 %	0.0 %
Primary school	68	390	0.56	[0.20, 0.91]						
Secondary school	23	112	0.36	[0.09, 0.62]						
College and university	12	15	0.35	[0.04, 0.66]						
Publication status										
Published literature	62	355	0.60	[0.45, 0.75]	4.67 (1)	.03	2877.82 (537)	< .001	0.0 %	4.6 %
Grey literature	43	184	0.34	[0.15, 0.52]						

Note. m = Number of studies, k = Number of effect sizes, g = Hedges' g , 95 % CI = 95 % Wald confidence interval, Q_M = Q -statistic underlying the test of moderators, Q_E = Q -statistic underlying the test for residual heterogeneity, df = degrees of freedom, R_2^2 = Level-2 variance

explanation, R_3^2 = Level-3 variance explanation. Values of variance explanations are based on the reduction of level-2 or level-3 variance after introducing moderators (Snijders & Bosker, 2012). For moderators, the number of studies may exceed $m = 105$, because some of the moderators were coded at the effect size rather than the study level.

^a The overall number of effect sizes may exceed $k = 539$, because some studies contained samples from multiple educational levels.

^b The test for residual heterogeneity failed to converge for this study-level moderator.

[#] Missing data in moderator variables occurred.

Table 4

Selection of Models to Estimate Near and Far Transfer Effects

<i>Model</i>	<i>g</i>	95 % CI	<i>z</i>	σ_2^2 [95 % CI]	σ_3^2 [95 % CI]	<i>-2LL (df)</i>	AIC	BIC	Model comparison	LRT
Near Transfer—Computer programming skills (<i>m</i> = 13, <i>k</i> = 19)										
1	0.75	[0.39, 1.11]	4.1*	0.037 [0.000, 0.300]	0.337 [0.063, 1.119]	32.5 (3)	38.5	41.2	-	-
2	0.75	[0.39, 1.11]	4.1*	0	0.367 [0.122, 1.151]	33.9 (2)	37.9	39.6	1 vs. 2	$\chi^2(1) = 1.3, p = .25$
3	0.66	[0.37, 0.95]	4.4*	0.318 [0.114, 0.847]	0	38.6 (2)	42.6	44.3	1 vs. 3	$\chi^2(1) = 6.0^*$
4	0.77	[0.66, 0.88]	13.7*	0	0	76.3 (1)	78.3	79.2	1 vs. 4	$\chi^2(2) = 47.7^*$
Far Transfer—Cognitive skills other than programming (<i>m</i> = 102, <i>k</i> = 520)										
1	0.47	[0.35, 0.59]	7.8*	0.203 [0.162, 0.252]	0.273 [0.184, 0.404]	1087.2 (3)	1093.2	1105.9	-	-
2	0.47	[0.36, 0.58]	8.1*	0	0.308 [0.225, 0.430]	1492.9 (2)	1496.9	1505.4	1 vs. 2	$\chi^2(1) = 405.8^*$
3	0.45	[0.36, 0.49]	13.3*	0.417 [0.353, 0.494]	0	1224.6 (2)	1228.6	1237.1	1 vs. 3	$\chi^2(1) = 137.4^*$
4	0.33	[0.31, 0.36]	28.9*	0	0	2606.6 (1)	2608.6	2612.9	1 vs. 4	$\chi^2(2) = 1519.5^*$

Note. *g* = Effect size Hedges' *g*, 95 % CI = 95 % Wald confidence intervals, σ_2^2 = Level-2 variance, σ_3^2 = Level-3 variance, *-2LL (df)* = -2

×Loglikelihood value with *df* degrees of freedom, AIC = Akaike's Information Criterion, BIC = Bayesian Information Criterion,

LRT = Likelihood-Ratio Test, *m* = Number of studies, *k* = Number of effect sizes.

* $p < .01$

Table 5

Categorical Moderator Analyses of the Near Transfer Effects ($m = 13$, $k = 19$)

<i>Moderator variables</i>	<i>m</i>	<i>k</i>	<i>g</i>	95 % CI	$Q_M(df)$	<i>p</i>	$Q_E(df)$	<i>p</i>	R_2^2	R_3^2
Study characteristics										
Study design (<i>coded at the levels of effect sizes and studies</i>)										
Pretest-posttest control group design	8	11	0.70	[0.21, 1.19]	0.14 (1)	.71	87.05 (17)	< .001	0.0 %	0.0 %
Posttest-only design	5	8	0.84	[0.24, 1.45]						
Randomization										
Random group assignment	4	8	0.29	[-0.27, 0.86]	3.51 (1)	.06	56.63 (17)	< .001	0.0 %	25.3 %
Non-random group assignment	9	11	0.95	[0.56, 1.34]						
Treatment of control group(s) (<i>coded at the level of effect sizes</i>)										
Treated controls	1	1	0.23	[-1.09, 1.54]	0.68 (1)	.41	81.99 (17)	< .001	0.0 %	0.0 %
Untreated controls	12	18	0.80	[0.41, 1.19]						
Treatment of control group(s) (<i>coded at the level of studies</i>)										
Treated controls	1	1	0.23	[-1.16, 1.61]	1.02 (2)	.60	79.57 (16)	< .001	0.0 %	0.0 %
Untreated controls	10	12	0.86	[0.41, 1.31]						
Mixed controls	2	6	0.52	[-0.43, 1.46]						
Matching										
Matched	2	5	0.22	[-0.70, 1.13]	1.53 (1)	.22	72.25 (17)	< .001	0.0 %	6.7 %
Not matched	11	14	0.84	[0.46, 1.22]						
Student collaboration [#]										

Collaboration	6	15	0.68	[0.24, 1.12]	0.18 (1)	.67	51.55 (12)	< .001	0.0 %	0.0 %
No collaboration	2	3	0.48	[-0.33, 1.29]						
Programming tool[#]										
Visual tool	10	14	0.82	[0.36, 1.27]	0.01 (1)	.92	80.34 (16)	< .001	0.0 %	0.0 %
Text-based tool	2	4	0.76	[-0.21, 1.73]						
Study context										
Regular lessons	9	12	0.69	[0.23, 1.15]	0.28 (1)	.60	84.85 (17)	< .001	0.0 %	0.0 %
Extracurricular activity	4	7	0.91	[0.22, 1.61]						
Type of outcome measure										
Standardized test	1	1	0.71	[-0.60, 2.03]	0.01 (1)	.95	87.01 (17)	< .001	0.0 %	0.0 %
Unstandardized test	12	18	0.76	[0.36, 1.16]						
Facets of Programming										
Programming skills	11	14	0.70	[0.29, 1.12]	0.44 (1)	.51	82.46 (17)	< .001	0.0 %	0.0 %
Programming knowledge	2	5	1.08	[0.05, 2.11]						
Sample characteristics										
Educational level^a										
Kindergarten	1	1	0.41	[-1.10, 1.92]	0.43 (3)	.93	74.06 (15)	< .001	0.0 %	0.0 %
Primary school	6	11	0.90	[0.24, 1.57]						
Secondary school	5	6	0.69	[-0.02, 1.39]						
College and university	2	3	0.72	[-0.23, 1.67]						
Publication status										
Published literature	10	15	0.72	[0.28, 1.15]	0.13 (1)	.72	86.27 (17)	< .001	0.0 %	0.0 %
Grey literature	3	4	0.88	[0.10, 1.66]						

Note. m = Number of studies, k = Number of effect sizes, g = Hedges' g , 95 % CI = 95 % Wald confidence interval, Q_M = Q -statistic underlying the test of moderators, Q_E = Q -statistic underlying the test for residual heterogeneity, df = degrees of freedom, R_2^2 = Level-2 variance explanation, R_3^2 = Level-3 variance explanation. Values of variance explanations are based on the reduction of level-2 or level-3 variance after introducing moderators (Snijders & Bosker, 2012).

^a The overall number of effect sizes may exceed $k = 19$, because some studies contained samples from multiple educational levels.

[#] Missing data in moderator variables occurred.

Table 6

Categorical Moderator Analyses of the Far Transfer Effects ($m = 102, k = 520$)

<i>Moderator variables</i>	<i>m</i>	<i>k</i>	<i>g</i>	95 % CI	$Q_M(df)$	<i>p</i>	$Q_E(df)$	<i>p</i>	R_2^2	R_3^2
Study characteristics										
Study design (<i>coded at the level of effect sizes</i>)										
Pretest-posttest control group design	76	352	0.48	[0.35, 0.62]	0.13 (1)	.72	2820.4 (518)	< .001	0.0 %	0.0 %
Posttest-only design	32	168	0.45	[0.27, 0.63]						
Study design (<i>coded at the level of study</i>)										
Pretest-posttest control group design	71	324	0.44	[0.30, 0.59]	1.06 (2)	.59	- ^b	- ^b	0.0 %	0.0 %
Posttest-only design	25	109	0.50	[0.26, 0.74]						
Mixed design	6	87	0.69	[0.23, 1.16]						
Randomization										
Random group assignment	46	296	0.57	[0.40, 0.75]	2.29 (1)	.13	2698.60 (518)	< .001	0.0 %	3.0 %
Non-random group assignment	56	224	0.39	[0.23, 0.55]						
Treatment of control group(s) (<i>coded at the level of effect sizes</i>)										
Treated controls	41	162	0.15	[-0.02, 0.31]	39.33 (1)	< .001	2767.84 (518)	< .001	17.2 %	0.0 %
Untreated controls	76	358	0.64	[0.50, 0.77]						
Treatment of control group(s) (<i>coded at the level of studies</i>)										
Treated controls	25	87	0.59	[0.34, 0.84]	1.95 (2)	.38	- ^b	- ^b	0.0 %	0.0 %
Untreated controls	61	256	0.48	[0.32, 0.63]						

Mixed controls	16	177	0.32	[0.04, 0.60]						
Matching [#]										
Matched	29	153	0.45	[0.22, 0.67]	0.17 (1)	.68	2770.22 (502)	< .001	0.0 %	0.0 %
Not matched	68	351	0.50	[0.35, 0.65]						
Student collaboration [#]										
Collaboration	48	292	0.56	[0.36, 0.75]	0.34 (1)	.56	2281.99 (386)	< .001	0.0 %	0.0 %
No collaboration	21	96	0.45	[0.15, 0.75]						
Programming tool [#]										
Visual tool	82	453	0.50	[0.36, 0.63]	0.47 (1)	.50	2821.09 (502)	< .001	0.0 %	0.0 %
Text-based tool	16	51	0.38	[0.07, 0.69]						
Study context										
Regular lessons	87	434	0.45	[0.32, 0.58]	0.53 (1)	.47	2798.06 (518)	< .001	0.0 %	0.0 %
Extracurricular activity	15	86	0.58	[0.27, 0.88]						
Type of outcome measure										
Standardized test	58	287	0.41	[0.26, 0.55]	2.60 (1)	.11	2831.09 (518)	< .001	0.0 %	3.2 %
Unstandardized test	51	233	0.55	[0.40, 0.70]						
Sample characteristics										
Educational level ^a										
Kindergarten	6	21	0.51	[-0.02, 1.04]	2.45 (3)	.49	2800.63 (516)	< .001	0.0 %	0.0 %
Primary school	67	379	0.53	[0.38, 0.68]						
Secondary school	23	106	0.33	[0.06, 0.59]						
College and university	11	42	0.31	[-0.01, 0.62]						
Publication status										
Published literature	59	340	0.58	[0.42, 0.73]	4.38 (1)	.04	2738.92 (518)	< .001	0.0 %	4.9 %
Grey literature	43	180	0.32	[0.02, 0.49]						

Note. m = Number of studies, k = Number of effect sizes, g = Hedges' g , 95 % CI = 95 % Wald confidence interval, Q_M = Q -statistic underlying the test of moderators, Q_E = Q -statistic underlying the test for residual heterogeneity, df = degrees of freedom, R_2^2 = Level-2 variance explanation, R_3^2 = Level-3 variance explanation. Values of variance explanations are based on the reduction of level-2 or level-3 variance after introducing moderators (Snijders & Bosker, 2012). ^a The overall number of effect sizes may exceed $k = 520$, because some studies contained samples from multiple educational levels. ^b The test for residual heterogeneity failed to converge for this study-level moderator. [#] Missing data in moderator variables occurred.

Table 7

Continuous Moderator Analyses of the Near and Far Transfer Effects

<i>Moderator variables</i>	Intercept	<i>SE</i>	<i>B</i>	<i>SE</i>	$Q_M(df)$	<i>p</i>	$Q_E(df)$	<i>p</i>	R_2^2	R_3^2
Near Transfer—Computer programming skills ($m = 13, k = 19$)										
Study characteristics										
Intervention length (in hours) #	0.81	0.25	-0.05	0.23	0.04 (1)	.84	66.55 (13)	< .001	–	0.0 %
Publication year	0.74	0.20	0.06	0.19	0.10 (1)	.76	84.13 (17)	< .001	–	0.0 %
Sample characteristics										
Average age (in years) #	1.00	0.39	-0.29	0.39	0.55 (1)	.46	38.94 (5)	< .001	–	0.0 %
Proportion of female students #	0.91	0.32	-0.24	0.35	0.46 (1)	.50	34.52 (6)	< .001	–	0.0 %
Far Transfer—Cognitive skills other than programming ($m = 102, k = 520$)										
Study characteristics										
Intervention length (in hours) #	0.48	0.07	0.01	0.06	0.02 (1)	.88	2237.58 (479)	< .001	0.0 %	0.1 %
Publication year	0.46	0.06	0.07	0.06	1.55 (1)	.21	2823.00 (518)	< .001	0.3 %	0.0 %
Sample characteristics										
Average age (in years) #	0.53	0.09	-0.06	0.07	0.72 (1)	.40	920.74 (227)	< .001	0.0 %	0.0 %
Proportion of female students #	0.40	0.08	-0.04	0.07	0.33 (1)	.57	1727.84 (325)	< .001	0.2 %	0.4 %

Note. Q_M = Q -statistic underlying the test of moderators, Q_E = Q -statistic underlying the test for residual heterogeneity, df = degrees of freedom, R_2^2 = Level-2 variance explanation, R_3^2 = Level-3 variance explanation. Values of variance explanations are based on the reduction of level-2 or level-3 variance after introducing moderators. All moderators were z -standardized.

Missing data in moderator variables occurred.

Table 8

Models to Estimate Far Transfer Effects Differentiated by Cognitive Skills

<i>Cognitive Skills</i>	<i>m</i>	<i>k</i>	<i>g</i>	95 % CI	<i>z</i>	σ_2^2 [95 % CI]	σ_3^2 [95 % CI]	-2LL (<i>df</i>)	AIC	BIC
Reasoning	59	229	0.37	[0.23, 0.52]	5.1**	0.067	0.246	399.8 (3)	405.8	416.1
						[0.035, 0.110]	[0.157, 0.393]			
Creative thinking	10	77	0.73	[0.27, 1.20]	3.1**	0.257	0.505	160.4 (3)	166.4	173.4
						[0.159, 0.417]	[0.159, 1.733]			
Metacognition	10	48	0.44	[0.01, 0.88]	2.0*	0	0.433	87.1 (2)	91.1	94.8
							[0.156, 1.467]			
Spatial skills	19	38	0.37	[0.08, 0.67]	2.5*	0.152	0.265	77.4 (3)	83.4	88.2
						[0.033, 0.444]	[0.001, 0.888]			
Mathematical skills	36	102	0.57	[0.34, 0.80]	4.8**	0.263	0.321	228.7 (3)	234.7	242.5
						[0.163, 0.420]	[0.147, 0.650]			
Literacy	9	19	-0.02	[-0.12, 0.08]	-0.4	0	0	21.3 (1)	23.3	24.2
School achievement	6	7	0.28	[0.14, 0.42]	3.9**	0	0	6.3 (1)	8.3	9.3

Note. *g* = Effect size Hedges' *g*, 95 % CI = 95 % Wald confidence intervals, σ_2^2 = Level-2 variance, σ_3^2 = Level-3 variance, -2LL (*df*) = -2

×Loglikelihood value with *df* degrees of freedom, AIC = Akaike's Information Criterion, BIC = Bayesian Information Criterion, *m* = Number of studies, *k* = Number of effect sizes.

* $p < .05$, ** $p < .01$

Table 9

Subskill Moderator Analyses of the Transfer Effects by Cognitive Skills

<i>Moderator variables</i>	<i>m</i>	<i>k</i>	<i>g</i>	95 % CI	$Q_M(df)$	<i>p</i>	$Q_E(df)$	<i>p</i>	R_2^2	R_3^2
Reasoning skills										
Subskills ^a										
Intelligence, attention, perception, and memory	29	97	0.25	[0.08, 0.43]	4.29 (2)	.12	756.02 (226)	< .001	0.0 %	8.5 %
Problem solving	30	112	0.47	[0.30, 0.65]						
Critical thinking	9	20	0.37	[0.07, 0.67]						
Skills assessment [#]										
Verbal	15	22	0.24	[0.06, 0.42]	0.08 (1)	.78	185.43 (91)	< .001	0.0 %	0.0 %
Non-verbal	24	71	0.21	[0.06, 0.36]						
Creative thinking skills										
Subskills ^a										
Flexibility	8	20	0.39	[-0.10, 0.88]	48.09 (3)	< .001	414.31 (73)	< .001	52.0 %	2.7 %
Fluency	10	24	0.44	[-0.04, 0.92]						
Originality	10	25	1.28	[0.80, 1.76]						
Elaboration	9	18	0.56	[0.06, 1.06]						
Skills assessment [#]										
Verbal	3	13	0.71	[0.25, 1.19]	0.01 (1)	.96	679.28 (67)	< .001	2.8 %	0.6 %
Non-verbal	10	56	0.72	[0.25, 1.19]						
Spatial skill										
Subskills ^a										
Spatial understanding and reasoning	18	34	0.28	[0.03, 0.52]	4.74 (1)	.03	116.25 (36)	< .001	0.0 %	66.3 %
Spatial memory	1	4	1.25	[0.41, 2.10]						
Mathematical skills										

Subskills ^a	<i>m</i>	<i>k</i>	<i>g</i>	95 % CI	Q_M	Q_E	R_2^2	R_3^2
Mathematics achievement, modeling, and problem solving	20	38	0.45	[0.14, 0.76]	1.22 (1)	.27	800.91 (100)	< .001
Mathematical concepts	20	64	0.66	[0.38, 0.95]				

Note. *m* = Number of studies, *k* = Number of effect sizes, *g* = Hedges' *g*, 95 % CI = 95 % Wald confidence interval, Q_M = *Q*-statistic underlying the test of moderators, Q_E = *Q*-statistic underlying the test for residual heterogeneity, *df* = degrees of freedom, R_2^2 = Level-2 variance explanation, R_3^2 = Level-3 variance explanation. Values of variance explanations are based on the reduction of level-2 or level-3 variance after introducing moderators (Snijders & Bosker, 2012).

^a The overall number of effect sizes may exceed the overall number of effect sizes (*k*), because some studies administered tests targeted at multiple subskills.

Missing data in moderator variables occurred.

Table 10

Summary of Key Findings

<i>Research Questions (RQs)</i>	<i>Key Findings</i>
RQ1a. Overall transfer effect size	$g = 0.49$, 95 % CI = [0.37, 0.61]
RQ1b. Moderators	Treatment of control group(s), publication status
RQ2a. Near transfer effect size	$g = 0.75$, 95 % CI = [0.39, 1.11]
RQ2b. Moderators	Randomization of experimental groups
RQ3a. Overall far transfer effect size	$g = 0.47$, 95 % CI = [0.35, 0.59]
RQ3b. Moderators	Treatment of control group(s), publication status
RQ4a. Far transfer effects by cognitive skills	<ul style="list-style-type: none"> • Reasoning: $g = 0.37$, 95 % CI = [0.23, 0.52] • Creative thinking: $g = 0.73$, 95 % CI = [0.27, 1.20] • Metacognition: $g = 0.44$, 95 % CI = [0.01, 0.88] • Spatial skills: $g = 0.37$, 95 % CI = [0.08, 0.67] • Mathematical skills: $g = 0.57$, 95 % CI = [0.34, 0.80] • Literacy: $g = -0.02$, 95 % CI = [-0.12, 0.08] • School achievement: $g = 0.28$, 95 % CI = [0.14, 0.42]
RQ4b. Differences in effect sizes by cognitive skill	Significant differences existed—8.0 % variance explanation by differences between cognitive skills

Figures

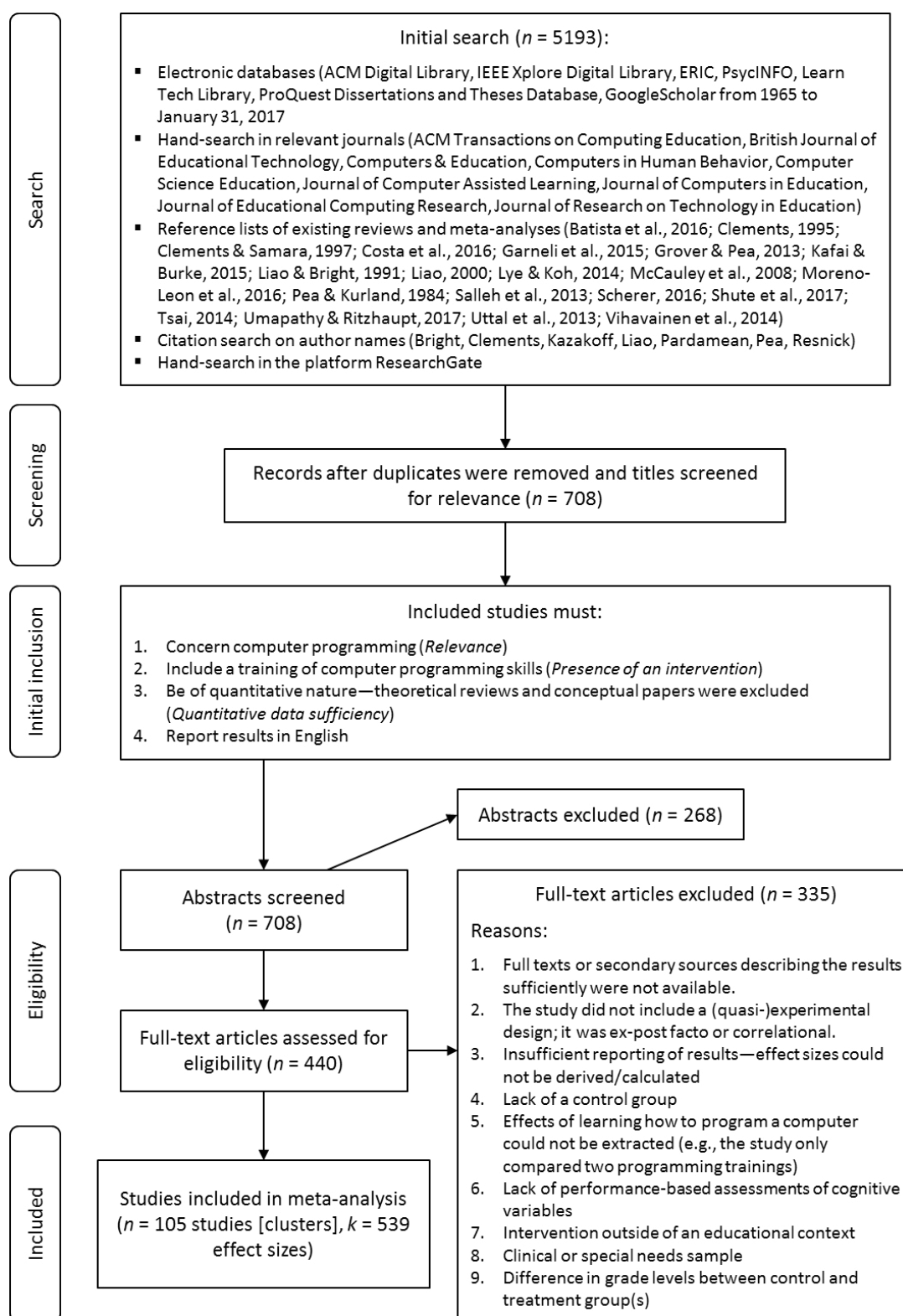
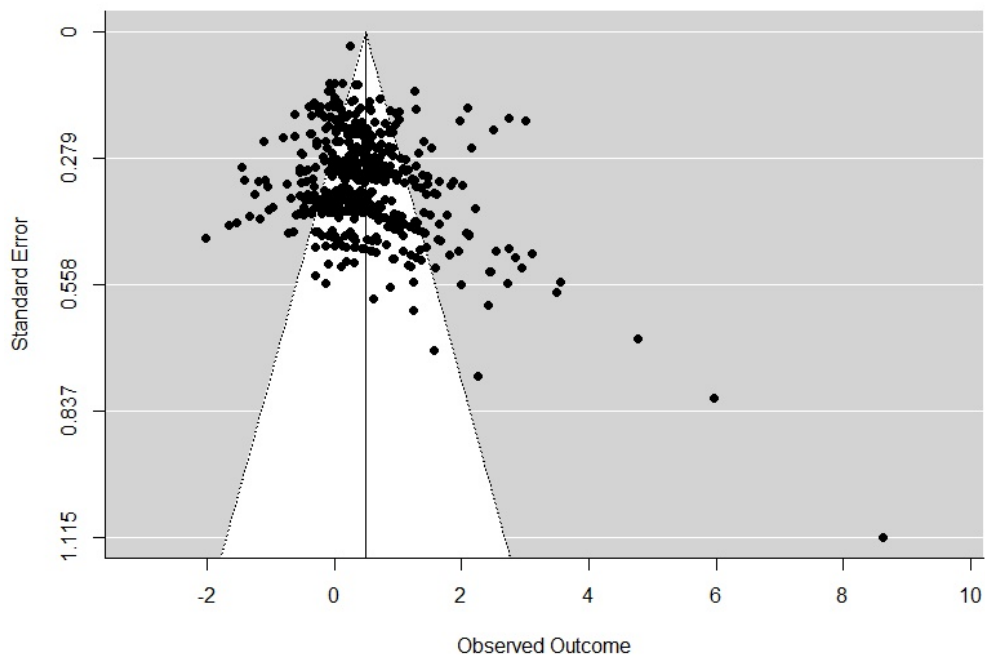
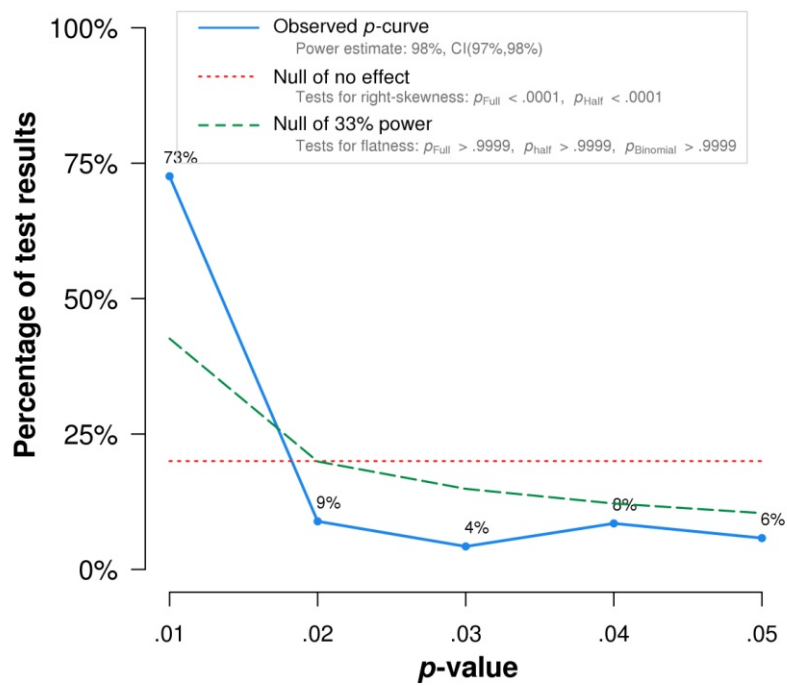


Figure 1. Flow diagram describing the literature search and the selection of eligible transfer effect studies (adapted from the PRISMA Statement; Moher, Liberati, Tetzlaff, Altman, & The PRISMA Group, 2009).

(a)



(b)



Note: The observed p -curve includes 259 statistically significant ($p < .05$) results, of which 218 are $p < .025$. There were 275 additional results entered but excluded from p -curve because they were $p > .05$.

Figure 2. (a) Funnel plot and (b) p -curve of the full data set ($m = 105, k = 539$).