education sciences

MDPI

Article

Computational Thinking in Grade 1: An Educational Robotics Study Using the intelino Smart Train

Raphael Fehrmann

Institute of Education, University of Münster, 48143 Münster, Germany; raphael.fehrmann@uni-muenster.de

Abstract

Computational thinking is increasingly regarded as an important component of digital education in primary school. Educational robotics is often discussed as a promising way to introduce computational thinking and promote problem-solving skills, which are key for the future, in early learning settings. However, empirical evidence on the extent to which computational thinking can be systematically fostered in Grade 1 students through short robotics-based instructional units remains limited. This study therefore investigates whether the computational thinking of first graders can be supported through an educational robotics intervention using the intelino Smart Train. A quasi-experimental pre-test/post-test design with an intervention group and a control group was employed. Students completed TechCheck-1 at two measurement points (before and after the intervention) to assess their basic computational thinking skills. The intervention group participated in a six-unit instructional intervention in which they controlled the intelino Smart Train through colour-coded commands. The findings indicate that the participating children already showed comparatively high computational thinking skills at the beginning of the intervention. No significant gender differences were found, and the intervention group did not demonstrate a significant advantage over the control group, which may also be related to ceiling effects. However, both groups showed learning gains across the measurement period. The results suggest that computational thinking can already be meaningfully addressed in Grade 1, but its systematic promotion may require longer-term curricular integration rather than a single short, isolated intervention.

Keywords: computational thinking; educational robotics; problem-solving skills; digital literacy

Check for updates

Academic Editor: Sarika Kewalramani

Received: 25 February 2026
Revised: 21 April 2026
Accepted: 23 April 2026
Published: 25 April 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. The Influence of Digitalization on Early Childhood

Digital technologies are increasingly shaping children's everyday lives from an early age. Children encounter digital systems in the form of voice assistants, smart toys, and learning apps, but they often do not understand how such technologies function or how algorithmic and data-based processes influence their behaviour and outputs (Flanagan et al., 2023; Fehrmann & May, 2025). For this reason, early digital education should not be limited to merely using digital tools. Rather, it should help children gradually understand technological structures, reflect on their effects, and engage with them in critical, problem-solving ways (GI, 2016; Fehrmann, 2024; Fehrmann & May, 2025).

In this context, computational thinking is increasingly discussed as a key component of digital literacy. Computational thinking refers to forms of thinking and problem solving that involve analyzing problems, breaking them down into sub-problems, recognizing

patterns, and developing step-by-step solution strategies (Wing, 2006, 2008; Bollin, 2016; Aho, 2012). In early childhood and early primary education, computational thinking can help children to understand rule-based and algorithmic systems in age-appropriate ways. This allows them to actively shape these systems rather than merely interact with them. In this way, computational thinking can contribute to children's ability to engage with digital technologies reflectively and purposefully.

Previous research shows that even young children can demonstrate basic forms of computational thinking in structured and playful learning settings (Angeli & Valanides, 2020; Bahr, 2025; Bati, 2021; Bers, 2018a, 2018b; Yang et al., 2022; Zeng et al., 2023). Studies indicate that children in preschool and early primary school can work with sequencing, pattern recognition, simple algorithms, and debugging processes when these are presented in age-appropriate ways (Bers, 2018a, 2018b; Bati, 2021; Yang et al., 2022; Zeng et al., 2023). At the same time, the existing literature also clarifies that there are still only a few empirically validated instruments for assessing computational thinking in this age group. Moreover, there is only limited evidence on how such skills develop over time, particularly in German-speaking contexts (Relkin et al., 2020; Zeng et al., 2023).

This is the point of departure for the present study. The focus of this study is first-grade students, and TechCheck-1 was used, as it is one of the few standardized instruments for measuring computational thinking in early primary education (Relkin et al., 2020). In addition, the study examines whether a short, structured intervention with the intelino Smart Train learning robot contributed to the development of computational thinking skills over six lessons spread out over four weeks. Educational robotics is of particular interest in this context because it allows children to engage with computational thinking through concrete, observable, and action-oriented problem-solving tasks (Bahr, 2025; Bati, 2021; Falloon, 2024; Komis & Misirli, 2011, 2012; Yang et al., 2022).

Considering this background, the present study investigates computational thinking skills in Grade 1 within a German-speaking educational environment. The intervention uses the TechCheck-1 assessment tool and a short educational robotics intervention employing the intelino Smart Train. The specific research questions guiding this study are formulated at the conclusion of the literature review and operationalized in the Methodology section.

2. Theoretical Background

2.1. Computational Thinking as a Fundamental Skill

Computational thinking, defined as the ability to formulate problems and design solutions that can be effectively executed by an information-processing entity, whether human or machine (Wing, 2006, 2008), has been well-established in the scientific literature since the 1980s (Pollak & Ebner, 2019). A key aspect of computational thinking is its close link to the development and analysis of algorithms, which help systematically structure and solve problems (Aho, 2012). The multi-step approach to problem solving, during which options for action are generated, tested, and evaluated, is conceptually aligned with the structure of an algorithm, and the phases of problem identification, solution strategy development, and solution analysis are formally represented in algorithmic form (Bollin, 2016). The goal is to systematically formalize and algorithmically structure the problem-solving process and the developed solution approach through a step-by-step modelling process (Denning, 2017). This approach aims to describe both components so precisely and comprehensively that their structure and internal logic correspond to that of an algorithm (Fehrmann & May, 2025). Key actions performed during computational thinking, compiled from the categorizations of various authors, include the following:

- Collecting, analyzing, and presenting data (data literacy);
- Recognizing recurring structures and relationships (pattern recognition);

- Breaking down complex problems into smaller, manageable sub-problems (decomposition);
- Filtering out essential information (abstraction);
- Developing step-by-step solutions (algorithmic thinking);
- Evaluating, testing, and improving solutions (evaluation);
- Transferring solutions and patterns to other problems (generalization)

(Cansu & Cansu, 2019; Chakraborty, 2024; Kallia et al., 2021; Palts & Pedaste, 2020; Shute et al., 2017; Tsai et al., 2020; Weintrop et al., 2016; Yadav et al., 2016). Computational thinking encompasses analytical and logical thinking and has also been linked to creativity (Kallia et al., 2021; Shute et al., 2017). Although computational thinking is based on concepts from computer science, it can be applied to numerous problems in different domains and disciplines and is considered a fundamental skill for the 21st century (Fadel et al., 2024). Computational thinking is also discussed as a foundation for understanding how algorithmic systems and, at later stages of education, more complex forms of artificial intelligence function (Teuber et al., 2022). Tedre (2022) proposes the notion of "Computational Thinking 2.0" as an extension of classical computational thinking. This extension explicitly addresses data-driven machine learning systems and associated issues such as model behavior, uncertainty, and ethical questions like bias or fairness. At the same time, a number of authors link computational thinking to Education for Sustainable Development, as well as a selection of Sustainable Development Goals, particularly SDG 4 on quality education and broader forms of digital participation (Buitrago-Flórez et al., 2021; Ramírez-Montoya et al., 2023; Erwinsyah et al., 2025). In the context of the present study, these broader perspectives on artificial intelligence and sustainability serve only as an educational frame that motivates why computational thinking is considered an important long-term goal.

Two questions therefore arise. First, how can computational thinking be understood as a relevant educational objective in primary education? Second, how can it be introduced fundamentally in developmentally appropriate ways from the early school years onward?

2.2. Computational Thinking in Primary Education: Measurement Methods and Findings

Current reviews indicate that children around three years of age possess rudimentary computational thinking abilities (Valdes et al., 2025), and studies on interventions show that children of this age can understand basic computational thinking concepts and can significantly increase their skills with targeted support (Angeli & Valanides, 2020; Bahr, 2025; Bati, 2021; Bers, 2018a, 2018b; Relkin et al., 2020; Yang et al., 2022; Zeng et al., 2023). These skills can be adequately activated, developed, and expanded in systematically designed, age-appropriate, playful learning settings that specifically stimulate logical thinking, creativity, and problem-solving skills (ibid.). Computational thinking in young children has been assessed using a range of methodological approaches, including tests, project- or task-based assessments, classroom observations, interviews, and self-evaluations. As summarized by Zeng et al. (2023), the existing literature includes different forms of CT measurement, but the approaches used remain heterogeneous across studies. Nevertheless, the availability of well-validated and standardized instruments for this age group remains limited (Sung, 2022; Jiménez et al., 2024). Prior research has often relied on unstructured or ad hoc methods, and rigorous assessment procedures for preschool and early childhood settings are still comparatively scarce (ibid.). One example of a validated instrument for this age group is *TechCheck-1*[1] (Relkin et al., 2020), which is used to assess computational thinking in the early primary years. Findings based on this instrument suggest that first-grade children already demonstrate measurable computational thinking skills, and that these skills may be fostered through age-appropriate interventions. However, country-specific evidence for Germany remains limited.

Despite the importance of interactive and collaborative learning opportunities in developing computational thinking skills, the widespread integration of computational thinking interventions in everyday life is lacking (Valdes et al., 2025), due, in part, to the insufficient training and continuing education of teaching staff (Valdes et al., 2025; Weber & Greiff, 2023; Xing et al., 2025) and the absence or poor standardization of teaching materials (Valdes et al., 2025; Weber & Greiff, 2023). Additionally, evidence-based research is limited due to a lack of longitudinal studies (Paraskevopoulou-Kollia et al., 2025) and assessments of computational thinking in early education, particularly among preschoolers (Jiménez et al., 2024; Relkin et al., 2020; Su & Yang, 2023; Valdes et al., 2025). Against this background, the question is not only how computational thinking can be assessed in primary education but also which pedagogical approaches are particularly suitable for fostering these skills in age-appropriate and engaging ways.

2.3. Educational Robotics as a Learning Approach

Valdes et al. (2025) confirm that children can perceive and create simple sequences and patterns and formulate initial planning steps, even at an early age. They can then develop simple algorithms, create programs, transfer patterns, and solve problems. Programming is particularly well-suited for designing learning opportunities that promote computational thinking in early education because it provides access to creative media design by enabling the creation of digital products (Lepold & Ullmann, 2018). Approaches such as educational robotics, which is enactive and playful, enable children to experience computational thinking in an active and socially embedded way while developing their initial understanding of algorithmic structures; for example, they can program learning robots as a didactic medium (Bahr, 2025; Bati, 2021; Falloon, 2024; Fehrmann, 2024; Fehrmann & May, 2025; Komis & Misirli, 2011, 2012; Screpanti et al., 2021; Yang et al., 2022). These approaches underscore the importance of interactive technology for early childhood learning and development (Yang et al., 2022). Specifically encouraging learners to program learning robots allows them to explore how actuators and sensors work in situational contexts (Fehrmann & May, 2025), and in doing so, they solve realistic problems to better understand and optimize the control logic of child-friendly robots using computational thinking. Children generate meaningful actions and adaptive reactions from learning robots through precise programming (Misirli & Komis, 2014), and they use "pseudocodes," such as colour combinations, to elicit specific reactions from the robots, which makes abstract programming and problem-solving concepts tangible and experiential (Ching & Hsu, 2023; Misirli & Komis, 2023). Through direct interaction with the robots (usually without the use of a digital device), children receive immediate feedback and can independently identify and correct errors, stimulating key computational thinking processes, such as debugging and iterative improvement. Falloon (2024) demonstrates that students who follow a structured, problem-based coding curriculum progress significantly in learning and applying concrete computational thinking strategies, such as sequencing, debugging, and recognizing patterns in code. Moreover, they begin using recognized code patterns to solve new tasks, which fosters the development of programming skills, thereby stimulating algorithmic thinking and the ability to provide sequential instructions and conditions (Ching & Hsu, 2023; Misirli & Komis, 2023). Preschool children find working with robots motivating, and robotics-based interventions lead to significantly better sequencing, pattern recognition, and debugging results (Papadakis & Kalogiannakis, 2020). Following interventions, the computational thinking skills of students can become so well developed that they can create and analyze more complex sequences and abstract results and transfer strategies between tasks (Paraskevopoulou-Kollia et al., 2025). Working with robots also fosters social interaction and teamwork skills in learners (Ching & Hsu, 2023).

2.4. Computational Thinking in the German Primary Curriculum

An obstacle to designing learning settings is the inconsistent curricular anchoring of promoting computational thinking skills (Valdes et al., 2025). Previous research on computational thinking in children has predominantly focused on structured learning settings and targeted interventions (Paraskevopoulou-Kollia et al., 2025), suggesting that the visible development of computational thinking skills depends heavily on structured educational approaches and underscoring the need for structured, curriculum-based learning opportunities to promote computational thinking, which remain largely absent to date (Das & Mitra, 2024). Data from the European Commission (2016) show that computational thinking has been incorporated into the curricula of several European and non-European countries, including Denmark, France, Croatia, Malta, Turkey, and the United Kingdom. Bocconi et al. (2022) confirm this country-specific consolidation, noting that the subject is primarily incorporated into secondary education but is increasingly being incorporated into primary education as well. Das and Mitra (2024) demonstrate the heterogeneity of this incorporation and its implementation, highlighting the United Kingdom as a pioneer that has integrated computing and computational thinking at all education levels. The authors similarly describe Finland's approach, noting a direct link to acquiring technical skills and critical thinking/problem-solving abilities. Using Singapore as an example, Das and Mitra (2024) demonstrate that it is possible to link the acquisition of computational thinking skills to STEM-oriented basic education. Regarding the international perspective, Pollak and Ebner (2019) note that there has been no internationally coordinated frame of reference nor comprehensive, systematic implementation in regular classroom teaching, despite a wide variety of decentralized initiatives, meta-reviews, and draft curricula.

The following initial situation in Germany, where the study presented in the Methodology Section was conducted, can be presented to promote computational thinking in early childhood education. In early education, which ranges from daycare facilities to the first years of primary school, computational thinking should be integrated into everyday action-oriented learning arrangements rather than understood as an abstract preliminary programming exercise. Although the promotion of computational thinking is not explicitly stated in the North Rhine-Westphalian curriculum, the educational principles (MSB & MKFFI, 2018) and the Landesanstalt für Medien recommendations for promoting media literacy in daycare centres and primary schools (Eder et al., 2013) establish problem-solving skills as a consistent educational principle in digital and non-digital contexts. Students should be encouraged to develop their own questions, experiment with different solutions, verbally reflect on them, and use mistakes productively in further thinking and learning processes (MSB & MKFFI, 2018, pp. 73, 116 ff.). There is a close connection between promoting creativity and computational thinking: Children "engage with a variety of materials, tools, and technical processes through hands-on activities and experimentation. They discover cause-and-effect relationships and use them to solve problems and engage in creative activities. They can assess the significance of technical achievements and their impact on their everyday lives and form their own opinions on these matters" (MSB & MKFFI, 2018, p. 120, translated). The curricular starting point also clarifies that media production by children fosters creativity (Eder et al., 2013, p. 47). The interdisciplinary media competence framework for elementary school in North Rhine-Westphalia defines the "Problem Solving and Modelling" perspective as follows: children should understand the basic principles of digital systems, recognize and reflect on algorithmic patterns, describe problems formally, and plan structured, algorithmic sequences (Medienberatung NRW, 2018, 2020).

The practice of integrating IT into the curriculum, as illustrated by the example of North Rhine-Westphalia, can be observed in other federal states throughout Germany. The

KMK strategy, "Education in the Digital World" (KMK, 2016), is a reference framework covering the importance of basic IT skills that is to be integrated into all federal state curricula. Several federal states of Germany have developed their own media literacy frameworks based on the KMK strategy and have explicitly anchored basic computer literacy and computational thinking. Similar to the curricular integration in North Rhine-Westphalia, Bavaria has explicitly incorporated learning area 6 at their LehrplanPLUS, "Algorithmic Thinking and Visual Programming" into the digital education of special schools (grades 1–12). This learning area includes key computational thinking concepts, such as decomposition, sequential description, and algorithmic problem solving (ISB, 2022). Berlin-Brandenburg has integrated media education into the basic curriculum, emphasizing that necessary IT skills and subject-specific ways of thinking, including computational thinking, are promoted in all subjects (LIBRA, n.d.). Although the term "computational thinking" is not explicitly used, the core competencies in computational thinking are indirectly addressed via "informatic thinking," "algorithmic thinking," or "basic computer science education." Eickelmann et al. (2019, 2024) demonstrate the importance of promoting computational thinking early on, particularly in Germany. The International Computer and Information Literacy Study (ICILS) results show that, in 2018 and 2023, the computational thinking skills of German eighth graders (ages 13–15) were significantly lower than the international average. Moreover, according to the ICILS study, around 40 percent of eighth graders only achieve lower levels of skills and are therefore unable to adequately navigate digital spaces (Eickelmann et al., 2024).

Beyond curricular considerations, previous research also suggests that individual learner characteristics should be taken into account when examining computational thinking development in the early years, including gender as one possible analytical dimension.

2.5. Gender and Computational Thinking

Research on gender differences in computational thinking in early and primary education has so far produced a mixed and not yet fully consistent picture. The existing studies suggest that possible differences may depend on the specific age group, the instructional context, or the particular dimensions of computational thinking being assessed (Bati, 2021). At the same time, several findings do not indicate clear or robust gender-related skill differences in the early years of schooling. For example, Relkin et al. (2020) report comparable TechCheck-1 scores for boys and girls in the first grade, with no statistically significant gender differences. Likewise, Bati's (2021) systematic review does not point to a uniform pattern of gender effects in early computational thinking learning. Against this background, it seemed appropriate in this study not to assume gender differences a priori but to examine them empirically as a relevant analytical dimension.

2.6. Research Gap and Study Aim

Although computational thinking is increasingly recognized as an important educational objective in the early years of schooling, important gaps remain in the empirical literature. In the German primary education context, there is still limited evidence on the computational thinking skills of first-grade children, the extent to which these skills can be supported through educational robotics interventions in regular classroom settings, and whether gender plays a role at this stage. Moreover, previous research has produced mixed findings regarding gender and has only provided limited evidence for the early primary years in Germany. Against this background, the present study investigates the computational thinking skills of first-grade children using a quasi-experimental design, compares the development of an intervention group and a control group, and examines whether gender-related differences can be observed. The aim of this study is to contribute

context-specific empirical evidence to research on computational thinking in German primary education. The following section presents the main objectives, research design, and methodological procedures through which these questions are empirically examined.

3. Methodology

3.1. Research Questions and Study Aim

Given the research findings on early computational thinking development and the importance of structured learning settings in early education, the question arises as to how computational thinking skills in young children can be measured and supported through educational robotics. International studies show that children as young as preschool and early primary school age can perform tasks related to computational thinking and suggest that skill gains can be achieved through targeted interventions. However, according to the existing research, valid, empirically tested computational thinking assessments suitable for early education are rarely available. Furthermore, there are few empirical findings on computational thinking levels and development trajectories among early elementary school-age children in German-speaking countries.

In order to address the research gap identified in Section 2.6, this study utilizes the TechCheck-1 assessment to evaluate fundamental computational thinking skills in Grade 1. It employs a quasi-experimental pre-test/post-test design to ascertain whether a short, structured intervention utilizing the intelino Smart Train further enhances these skills within a four-week period.

The study is guided by the following three research questions:

Research question 1: What level of computational thinking skills do German first graders demonstrate without prior intervention, and how does their performance vary?

Research question 2: How do the computational thinking skills of first graders change during participation in a six-unit intervention using the intelino Smart Train learning robot, and how does this change compare with that observed in a control group receiving regular instruction?

Research question 3: Do the computational thinking test results show any indications of an increase in computational thinking skills depending on gender?

With this focus, this study contributes to the empirical assessment of computational thinking skills among first-grade children in a German-speaking context and examines whether the intelino Smart Train can serve as a low-threshold approach to promoting computational thinking in regular classroom settings.

3.2. Study Design

A quantitative, quasi-experimental, longitudinal intervention study with a pre-test and post-test was conducted (Döring, 2023). Four first-grade classes at a primary school served as the participant groups, with two classes serving as the experimental group and the other two serving as the control group. Both groups completed TechCheck-1, which assesses computational thinking skills, at two measurement points: before and after the intervention.

Between these measurement points, the experimental group participated in a six-part series of lessons with the intelino Smart Train learning robot, which was integrated into regular classroom instruction. During the same period, the control group received regular instruction without robotics-related computational thinking support. This design allowed changes in the computational thinking skills within the groups to be analyzed over time, as well as differences in the development processes between the intervention groups and control groups.

To strengthen the internal validity of the quasi-experimental design, existing classes within the same school setting were used, comparable testing conditions were established, and the procedures for both the intervention and assessments were standardized across measurement points.

3.3. Sample

Participants were recruited from a four-track elementary school in North Rhine-Westphalia, Germany, which allow the study to be conducted within a relatively uniform school context. Four first-grade classes participated (two as the experimental group and two as the control group). With a school social index of 4 (range: 1 to 9; MSB, 2023; Schräpler & Jeworutzki, 2021), the student body of the participating school can be described as socially mixed, with neither particularly privileged nor severely disadvantaged conditions.

The sample comprised a total of N = 66 first-grade students aged M = 6.8 years (SD = 0.6). Of these, n = 37 children were assigned to the experimental group and n = 29 children to the control group. The gender distribution was almost balanced, with 32 male students (48.5%) and 34 female students (51.5%), and it remained balanced within the groups. The experimental group consisted of 17 male (45.9%) and 20 female (54.1%) students, while the control group consisted of 15 male (51.7%) and 14 female (48.3%) students.

3.4. Measurement: TechCheck-1

The TechCheck-1 test (See Note 1) (Relkin et al., 2020; translation into German: Fehrmann, 2026) was used to assess computational thinking skills. As intended, the test administrator read all items aloud so children with little reading experience or underdeveloped written language skills could participate. TechCheck-1 comprises 15 multiple-choice items in a paper-and-pencil format. For each task, children select one of four fully illustrated answer options without written text. One point is awarded for each correct answer, resulting in a maximum score of 15 points. The items include child-friendly task formats, such as sequencing tasks, symbol series, object decomposition tasks, obstacle mazes, shape and symbol puzzles, object recognition, and symmetry problems, which operationalize computational thinking skills in a way that is appropriate for first-grade children. Two ungraded practice tasks introduce the format, and, in this study, the test took about 15 min to complete. According to Relkin et al. (2020), the instrument shows high reliability and construct validity, particularly for differentiating computational thinking skills in the lower and middle skills ranges of the first year of elementary school. The instrument was designed for first graders, and Relkin et al. (2020) report no significant gender-specific differences given an average level of proficiency in this age group. The authors also show that second graders perform significantly better than first graders, which supports the suitability of the test for differentiating performance across lower grade levels. While TechCheck-1 is used in several European countries according to the instrument's website, country-specific published findings for Germany are not yet available.

3.5. Intervention Material: intelino Smart Train

The intelino Smart Train (Figure 1) is a learning robot for children aged 3 to 10 that offers different approaches to programming and a variety of opportunities for computational thinking processes and problem solving.



Figure 1. The intelino Smart Train learning robot (photo: Raphael Fehrmann).

The intelino Smart Train (model INT-J1-SS1; manufacturer Innokind/intelino) is a programmable educational robot in the form of a train that runs on rails and is controlled via haptic colour codes or Bluetooth commands. The intelino Smart Train offers different levels of complexity through the interaction of various programming modes. However, it is limited in terms of the complexity of the problems it can solve due to its dependence on a rail system.

The intelino Smart Train is made of plastic and has colour sensors on the underside, an optical speedometer, an acceleration sensor, a microphone, and a Bluetooth connection. In addition to the motor, its actuators include a loudspeaker, various LEDs, and magnetic coupling that can be used to attach a wagon shaped like a passenger car. According to the manufacturer, the train can run for up to 100 min on its rechargeable micro-USB battery and reach a maximum speed of 100 cm/s. The modular rail system, which includes straight tracks, curves, switches, crossings, tunnels, and bridges, is expandable and compatible with standard wooden train sets and rail systems (e.g., Brio®), enabling flexible learning scenarios and allowing for combination with existing play materials.

The intelino Smart Train can be controlled using coloured tiles, known as *Action Snaps*, which are inserted into the tracks and come in green, red, blue, pink, yellow, and white. By combining these tiles by colour and number, children can visually and tactilely program the train without an additional digital device (Figure 2). The train runs over the tiles, recognizes their colour and arrangement with colour sensors, and executes the corresponding commands stored as default settings. For instance, children can use the colour tiles to control the train's direction and turning, adjust its speed, stop it, or couple and uncouple a carriage. Through direct interaction with the intelino Smart Train, children receive immediate feedback on whether the programmed action sequence produces the intended result. If the train reacts differently than expected, then the colour code has to be analyzed and modified, which activates computational thinking processes such as evaluation and decomposition. When routes or goals change, previously successful sequences must be adapted or transferred, which relates to abstraction, generalization, and transfer.



Figure 2. The intelino Smart Train can be programmed from first grade onwards with the help of Action Snaps (photo: Raphael Fehrmann).

Basic control concepts can be introduced, and computational thinking skills can be promoted by creating stimulating, diverse, and cognitively activating problem-solving tasks.

In the present study, the intelino Smart Train was used exclusively via its colour-coded Action Snaps on the modular rail system. The app-based control options and the extended programming environments shown below were deliberately not used in this intervention, as the age group of first graders had not yet acquired the reading, writing, and digital skills needed to engage meaningfully with app-, block-, or text-based programming. The detailed description of these additional levels is nevertheless included here, to show teaching applications and options for future studies: In addition to coding with colour tiles, the *intelino Smart Train Play app* provides various digital control options, ranging from full remote control of the train to defining custom commands with the *Snap Editor*, seamlessly combining unplugged and digital programming. There are also interfaces with various programming platforms, such as Scratch (the *intelino Smart Train Scratch app*), which enable the transition to structured programming concepts. Scratch allows users to logically link commands using graphical elements. Specific intelino Smart Train blocks are available in an English-language command library. Additionally, the *intelino Smart Train Python app* enables the transition to a text-based programming language, and the *intelino Smart Train Central app* includes digital explanations and short learning modules for using the trains and tracks.

Different programming approaches make it possible to accommodate children's varying levels of prior knowledge, learning speeds, and areas of interest. According to Bruner's (1971) EIS principle, the progression of the four programming approaches can be interpreted as a transition from enactive (colour tiles) to iconic (the *intelino Smart Train Scratch app*) to symbolic (the *intelino Smart Train Python app*/Python 3.7) representations. These modes allow differentiation based on prior experience, learning pace, and interests and are suitable for developing computational thinking from a didactic perspective.

According to Resnick's (2017) concept of "low floor—wide walls—high ceiling," the intelino Smart Train offers low barriers to entry for initial problem solving and programming in early childhood. Neither reading nor programming skills are required to control it, and initial successes through the simple placement of colour tiles are immediately visible

in the robot's reaction. At the same time, modular tracks and a wide range of tasks in subject-specific and interdisciplinary teaching enable a broad scope of content, in the sense of wide walls (e.g., letter paths in German lessons, route optimization, or geometry tasks in mathematics). In contrast, the high-ceiling potential is more moderate. Although connections to Scratch and Python allow for more complex control and problem-solving tasks (e.g., modelling more sophisticated actions and reactions), they remain tied to the predefined driving and action options on the rail system. Therefore, for advanced learners, the intelino Smart Train may lose its didactic value in the medium term if additional challenges and an expanded hardware setup are not created.

3.6. Intervention and Instructional Design

The intervention consisted of six 45 min lessons using the intelino Smart Train learning robot in a problem-oriented learning environment. In this environment, first graders could actively expand their computational thinking skills by programming the robot. The lesson series followed a two-phase didactic structure, progressing from initial familiarization with the robot to increasingly complex problem-solving tasks. Table 1 provides an overview of the contents and computational thinking focuses of the six intervention units.

Table 1. The structures and computational thinking focus of the six intervention units.

Unit	Topic	Core Activities	Computational Thinking Focus
1	Getting to know robots	Activating prior knowledge about robotics; discussing rules for safe use; exploring the basic functions of the intelino Smart Train	Understanding that robots require control and programmed instructions
2	Learning the robot language	Building simple tracks; operating the robot; introducing Action Snaps and colour code programming	Sequencing commands; understanding basic input–output logic; carrying out first programming actions
3	Learning the problem-solving steps	Introducing additional commands; learning the five problem-solving steps; solving a first structured task	Stepwise planning; algorithmic thinking; testing; improving solutions
4	Becoming problem solvers	Solving a new task independently; working with switches and direction changes; reflecting on solution strategies	Applying structured problem solving; planning routes; debugging; reflecting on procedures
5	Becoming code crackers	Interpreting existing programs; predicting robot behaviour; adapting given code sequences to new tasks	Pattern recognition; analyzing sequences; modifying existing solutions; transfer
6	Becoming problem-solving experts	Solving complex tasks by combining learned commands; presenting and justifying solutions; reflecting on learning progress	Combining commands strategically; decomposition; abstraction; generalization; communicating solutions

As shown in Table 1, the intervention combined technical familiarization, explicit problem-solving procedures, and increasingly complex programming tasks to support the gradual development of computational thinking skills.

During the first two units, students explored how the robot works by testing the basic technical features, observing how the Action Snaps function, and interpreting the robot's responses to specific snap colour commands. In these units, they established basic control concepts: "What happens when I place this colour combination? How do I make the train stop at a specific point?" The learners worked mainly in pairs and small groups, exchanging observations and adapting their programming based on observed and interpreted robot behaviour. The teacher took on a moderating role to promote computational thinking,

encouraging the children to verbalize their observations, break down problems, present ideas for solving problems in different ways, and highlight differences in approach. The productive handling of errors was particularly encouraged (e.g., "Which snap do you think is responsible for the train going the wrong way?").

In the following four learning units, students were encouraged to solve specific, predefined problems using the intelino Smart Train, applying their computational thinking skills and previously acquired programming knowledge via Action Snaps to solve these problems. Based on real-life challenges relevant to them, the students independently developed creative solution strategies, reflecting on their approach and deepening their programming skills in the process. The number of available colour combinations gradually increased, and specific parts, such as switches, were introduced. Students solved increasingly complex problems by programming the robot to reach specified destinations within different rail networks as efficiently and reliably as possible, and they were encouraged to take a structured approach: identifying the problem, planning routes, integrating colour code sequences, observing robot journeys, and identifying and correcting errors. Moreover, students were encouraged to structure longer command sequences, plan sections separately, and, in the event of malfunctions, change individual elements rather than recreate the entire program. Reflection phases during the plenary session allowed them to identify typical strategies, such as "always set the start and finish first" and "mark critical points," and link them to computational thinking. The teacher ensured that all students were engaged by mixing groups, distributing role cards with specific tasks, and assigning observation tasks.

For example, the sixth lesson focused on a more complex, narratively embedded problem, and in this phase, students applied and combined all the problem-solving steps they had learned in various ways. They were tasked with planning a "zoo visit" for the intelino Smart Train, solving the complex problem by purposefully combining different colour codes and developing creative, efficient solution strategies. Various stations were marked on a predefined or jointly sketched map of a zoo (e.g., the entrance, the giraffe enclosure, the monkey house, and the elephant enclosure). The train visited these stations in a specific order and performed specific actions at each one (e.g., "Drive to the giraffes. Stop there for 10 s to feed them."). The objective was to program the train to follow a specific route, ensuring that it visited all the designated stations, stopped at certain points, reacted accordingly, and ultimately returned to the starting point. The challenges were twofold: not only did specific snap commands have to be integrated at appropriate points, but double or reverse travel on individual sections also had to be planned and accounted for (e.g., "There is a tree on the track, so the intelino Smart Train must turn around. How will the robot behave when it runs the codes you used before in reverse?") (Figure 3).

Working in small groups, the children first analyzed the situation, asking questions such as "Which stations are important?" and "Where do we start?" Then, they determined the order of visits on the corresponding sections of the route and planned the associated programming step by step. The teacher reminded the students of the key steps (identify the problem, gather ideas, write a plan, enter the program, and test and improve) to promote consistent work in line with computational thinking. The robot was only provided after the students had put together a route and integrated the programming. In the final step, the students tested their programming with the intelino Smart Train and revised it if necessary. Providing the train only after the fourth problem-solving step was completed prevented hasty experimentation and encouraged thoughtful, planned action. During the debugging phase, the teacher drew attention to the relationship between the robot's behaviour and the colour code by asking questions such as "At what point does the train behave differently than expected?", "Which snap could be interfering here?", and "Is your code correct?" The children were then encouraged to formulate hypotheses and test them through small,

targeted modifications. Successful groups described how they tested the entire route piece by piece to systematically identify errors. Other groups developed a strategy of marking critical points on the map with symbols to keep track of corrections and modifications. Through repeated planning, testing, and targeted code modification, the children experienced computational thinking processes, such as decomposition (breaking down the overall task and route into sections), abstraction (focusing on relevant snaps), and generalization (transferring a functioning sequence to another route), in a meaningful narrative context, and they then reflected upon tried-and-true courses of action in a plenary session.

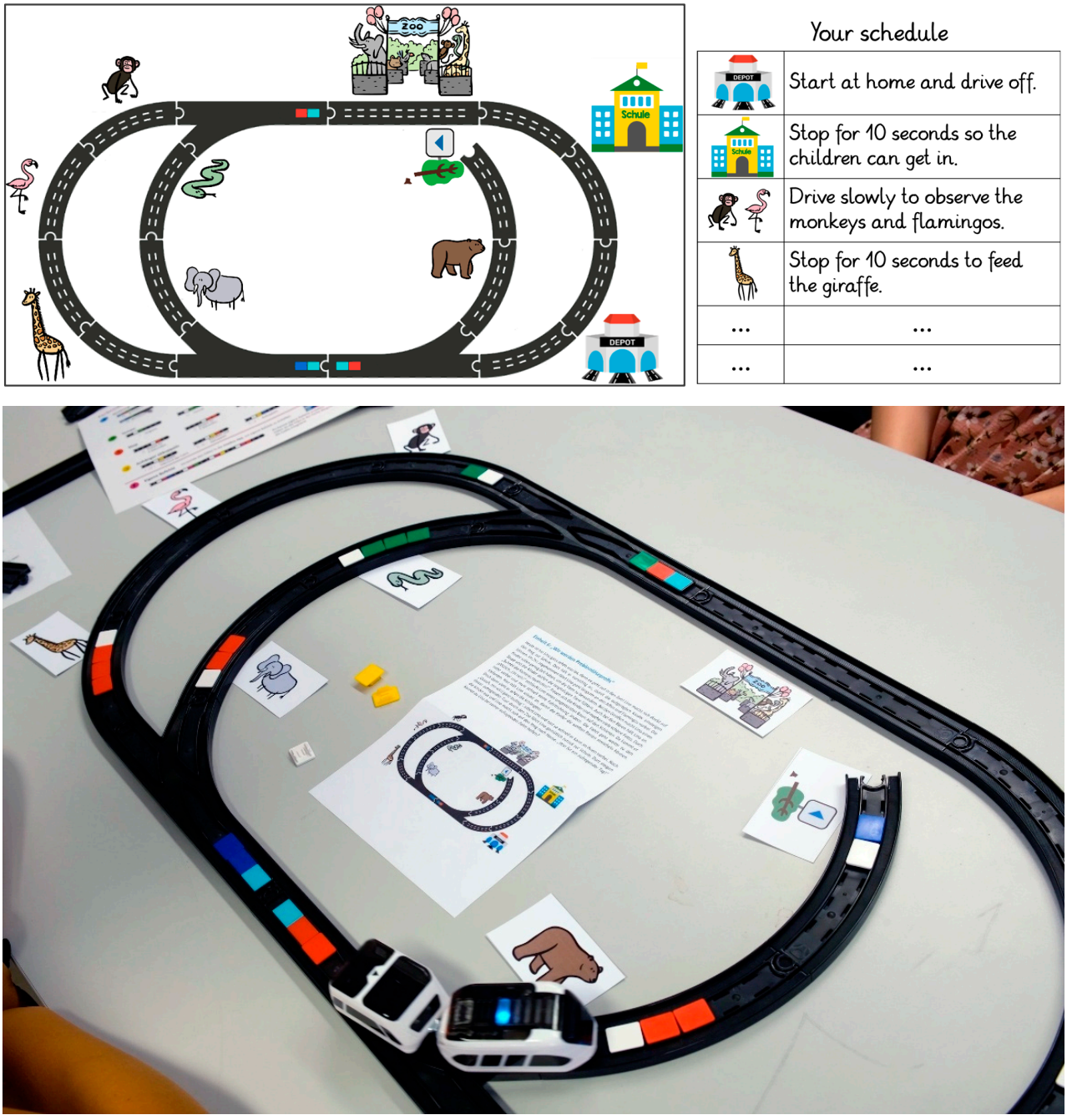


Figure 3. Setting up the track with picture cards and route instructions (task: Ina Selter; photos: Raphael Fehrmann; integrated icons and symbols: Cocomaterial (CC0) and intelino Smart Train; tasks available for download at https://go.rfehrmann.de/intelino25) (accessed on 14 February 2026).

3.7. Procedure

The pre- and post-tests were administered to the class groups in two survey phases in Februaryy 2025. The two phases were separated by a four-week intervention period, and the test dates at the beginning and end of Februaryy were integrated into the regular lesson schedule for both groups. The teacher remained present while an outside person administered the tests. At the beginning of each test session, the purpose and procedure were explained to the students in simple language. The children were told that this was not a class test and that mistakes were allowed to reduce performance pressure.

TechCheck-1 was administered using the same standardized procedure at both measurement points. On the cover sheet, the class, gender, and group membership were already indicated, and the children added the personal code. First, the two practice tasks were explained in a plenary session and solved using examples to ensure that all students understood the answer format. The students then worked through the 15 test items independently while the test supervisor read each task aloud and the children marked their answers in the respective answer fields. Questions about understanding the instructions were permitted, but no assistance with the content of the tasks was provided. Possible collaboration between students was prevented by placing learning office walls between them. The average completion time for both tests was 15 to 20 min. After completion, the test booklets were collected and coded so that the results from both measurement points could be matched for later analysis.

3.8. Data Analysis

To evaluate the data, the children's answers were digitally recorded, and the correct answers were added up for each student, with each correct answer receiving one point. The pre- and post-test data were combined. Descriptive statistics (mean values, standard deviations, minimum, and maximum) were calculated separated by measurement points, groups (experimental vs. control group), and gender. One-sample *t*-tests were conducted to examine differences between the sample values and reference values reported by Relkin et al. (2020). Group and gender differences were analyzed using independent-sample *t*-tests. To examine developmental changes between the two measurement points, difference scores between the pre- and post-test results were calculated and compared between groups. In addition, supplementary one-factor analyses of variance were conducted to examine the influence of group membership and gender on skills development. Effect sizes were determined to assess the practical significance of the observed differences.

4. Presentation of Key Results

The results are presented in three steps. First, pre-test scores are reported for the full sample and compared by gender and group membership. Second, post-test scores are presented by group and gender. Third, changes between the two measurement points are analyzed.

4.1. Results at First Measurement Point

In the pre-test, the students ($N = 66$) achieved a mean score of $M = 10.65$ ($SD = 2.18$) on the TechCheck-1 measure of computational thinking (Table 2), which was significantly higher than the mean reported for all first graders[2] (($\mu = 9.35$), $t(65) = 4.85$, $p < 0.001$, $d = 0.60$ with a medium effect) (Gollwitzer et al., 2017) and also significantly higher than the mean reported for white first-grade students (($\mu = 9.27$), $t(65) = 5.15$, $p < 0.001$, $d = 0.63$ for a medium effect) (Gollwitzer et al., 2017).

Table 2. Results of one-sample *t*-tests.

	Min	Max	*M*	*SD*	*t*(65)	*p*	*d*
SumPre (all), comparison to μ = 9.35 [a]	6.00	15.00	10.65	2.18	4.85	<0.001 *	0.60
SumPre (all), comparison to μ = 9.27 [b]	6.00	15.00	10.65	2.18	5.15	<0.001 *	0.63

Note: $N = 66$, two-tailed *p*-values, * $p < 0.05$; [a] all/Relkin et al. (2020); [b] white first graders/Relkin et al. (2020).

Male students ($n = 32$) achieved a mean pre-test score of $M = 10.78$ ($SD = 2.35$), whereas female students ($n = 34$) achieved $M = 10.53$ ($SD = 2.03$). The difference was not statistically significant ($t(64) = 0.47$, $p = 0.643$, $d = 0.12$). No significant differences were found within either group (Table 3).

Table 3. Comparison of pre-test scores by gender.

	m			*f*						
	n	*M*	*SD*	*n*	*M*	*SD*	*t*	*df*	*p*	*d*
SumPre (all), Male vs. Female	32	10.78	2.35	34	10.53	2.03	0.47	64	0.643	0.12
SumPre (EG), Male vs. Female	17	10.76	2.14	20	10.50	2.16	0.37	35	0.711	0.13
SumPre (CG), Male vs. Female	15	10.80	2.65	14	10.57	1.91	0.27	27	0.793	0.10

Note: two-tailed *p*-values.

The experimental group ($n = 37$) achieved a mean pre-test score of $M = 10.62$ ($SD = 2.13$), and the control group ($n = 29$) achieved $M = 10.69$ ($SD = 2.29$). The difference was not statistically significant ($t(64) = -0.13$, $p = 0.901$, $d = 0.03$) (Table 4).

Table 4. Comparison of pre-test scores by group membership.

	EG			CG						
	n	*M*	*SD*	*n*	*M*	*SD*	*t*	*df*	*p*	*d*
SumPre	37	10.62	2.13	29	10.69	2.29	−0.13	64	0.901	0.03

Note: two-tailed *p*-values.

4.2. Results at Second Measurement Point

In the post-test, the experimental group ($n = 37$) achieved a mean score of $M = 12.27$ ($SD = 2.33$), and the control group ($n = 29$) achieved $M = 11.93$ points ($SD = 1.96$). The difference between the groups was not statistically significant ($t(64) = 0.63$, $p = 0.532$, $d = 0.16$) (Table 5).

Table 5. Comparison of post-test scores by group membership.

	EG			CG						
	n	*M*	*SD*	*n*	*M*	*SD*	*t*	*df*	*p*	*d*
SumPost	37	12.27	2.33	29	11.93	1.96	0.63	64	0.532	0.16

Note: two-tailed *p*-values.

Within the experimental group, male students ($n = 17$) achieved a mean post-test score of $M = 12.24$ ($SD = 2.14$), whereas female students ($n = 20$) achieved $M = 12.30$ points ($SD = 2.54$). This difference was not statistically significant ($t(35) = 0.08$, $p = 0.934$, $d = 0.03$).

No statistically significant gender difference was found in the control group ($t(27) = 0.01$, $p = 0.995$, $d < 0.01$) (Table 6).

Table 6. Comparison of post-test scores by gender.

	m			*f*						
	n	***M***	***SD***	***n***	***M***	***SD***	***t***	***df***	***p***	***d***
SumPost (EG), Male vs. Female	17	12.24	2.14	20	12.30	2.54	0.08	35	0.934	0.03
SumPost (CG), Male vs. Female	15	11.93	2.12	14	11.93	1.86	0.01	27	0.995	<0.01

Note: two-tailed *p*-values.

4.3. Changes Between Two Measurement Points

Difference scores were calculated by subtracting pre-test scores from post-test scores to examine the changes between the two measurement points. The experimental group ($n = 37$) showed a mean increase of $M = 1.65$ ($SD = 2.06$), while the control group ($n = 29$) showed a mean increase of $M = 1.24$ ($SD = 1.77$). The difference between the groups was not significant ($t(64) = 0.85$, $p = 0.399$, $d = 0.21$) (Table 7).

Table 7. Comparison of change scores between experimental and control groups.

	EG			**CG**						
	n	***M***	***SD***	***n***	***M***	***SD***	***t***	***df***	***p***	***d***
Diff	37	1.65	2.06	29	1.24	1.77	0.85	64	0.399	0.21

Note: two-tailed *p*-values.

A supplementary one-factor analysis of variance likewise showed no significant effect of group membership on the development of computational thinking ($F(1,64) = 0.720$, $p = 0.399$, $\eta_p^2 = 0.011$). Figure 4 illustrates the score development for both groups across the two measurement points.

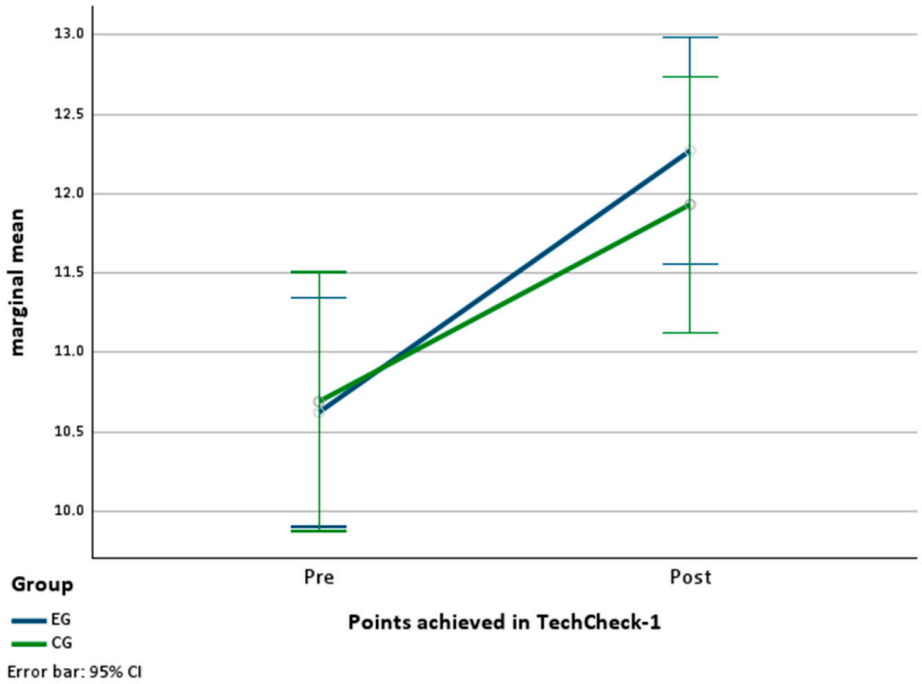


Figure 4. Points achieved on TechCheck-1 at both measurement points per group.

Within the experimental group, female students (n = 20) showed a mean gain of M = 1.80 (SD = 2.19), whereas male students (n = 17) showed a mean gain of M = 1.47 (SD = 1.94), but this difference was not statistically significant ($t(35) = 0.48$, $p = 0.634$, $d = 0.16$) (Table 8).

Table 8. Comparison of change scores in experimental group by gender.

	m			*f*						
	n	***M***	***SD***	***n***	***M***	***SD***	***t***	***df***	***p***	***d***
Diff (EG)	17	1.47	1.94	20	1.80	2.19	0.48	35	0.634	0.16

Note: two-tailed p-values.

A supplementary one-factor analysis of variance showed no significant effect of gender on the development of computational thinking within the experimental group ($F(1,35) = 0.997$, $p = 0.634$, $\eta_p^2 = 0.007$).

5. Discussion

5.1. Classification of Results and Answers to Research Questions

The reported results are discussed in relation to the research questions formulated in Section 3.1. The limitations are then outlined, and implications for future research are identified.

Research question 1: What level of computational thinking skills do German first graders demonstrate without prior intervention, and how does their performance vary?

The pre-test results show that the first graders achieved a significantly higher mean score on TechCheck-1 than the overall mean score reported by Relkin et al. (2020) in the international TechCheck-1 validation study. The first graders' mean score was also higher than the mean score reported by Relkin et al. (2020) for white students in their first year of school. This finding aligns with the observations of reviews indicating that early childhood STEM experiences and informal learning opportunities may foster to the development of computational thinking skills in this age group (Lavigne et al., 2023), particularly in comparatively well-resourced contexts such as Germany, where diverse and individualized learning opportunities may be available from an early age. At the same time, the results indicate that male and female first graders showed comparable computational thinking skills at the beginning of this study, and that no gender-specific difference was detectable in the pre-test. In addition, no statistically significant differences in computational thinking skills were found between the experimental and control group before the intervention.

According to these findings, the sample of German first graders already demonstrated a comparatively high initial level of computational thinking relative to the reported reference values, even without specific educational robotics support. This finding aligns with the initial research presented in this article, which demonstrates that children as young as preschool and early elementary school age can grasp fundamental computational thinking concepts and perform related tasks (Valdes et al., 2025). In structured, playful learning settings, children can plan sequences, recognize patterns, form simple algorithms, and perform debugging steps (Angeli & Valanides, 2020; Bahr, 2025; Bati, 2021; Bers, 2018a, 2018b; Relkin et al., 2020; Yang et al., 2022; Zeng et al., 2023).

Given this, it is plausible that the comparatively high initial scores in this sample reflect early media experiences, informal learning opportunities, or indirect school influences. At the same time, the TechCheck-1 data reported by Relkin et al. (2020) should not be interpreted as pure baseline scores without support because they were collected in the

context of ongoing interventions. The present study therefore adds empirical evidence on computational thinking levels among first graders in a German-speaking context.

This study builds on the existing research by showing that first graders in the present sample achieved computational thinking scores above published reference values without prior educational robotics intervention. However, the dispersion of the pre-test results shows that the sample included children with both very high and low levels of computational thinking, underscoring the need for diagnostically sensitive instruments and differentiated support concepts.

Research question 2: How do the computational thinking skills of first graders change during participation in a six-unit intervention using the intelino Smart Train learning robot, and how does this change compare with that observed in a control group receiving regular instruction?

Overall, the results show that both groups demonstrated an increase in performance on TechCheck-1 between the pre- and post-tests. The control group achieved a slightly lower mean post-test score than that of the experimental group, but the difference between the groups was not statistically significant. Overall, the findings indicate that students in both groups continued to show comparable levels of performance in computational thinking after the intervention and did not differ significantly from one another on the post-test.

Regarding skills development, students in both groups showed positive average performance gains, meaning that both groups achieved higher post-test than pre-test scores. The analysis of the difference scores shows that both the experimental and control groups improved their computational thinking skills between the pre-test and post-test. However, the gains of the students who participated in the intervention with the intelino Smart Train robot did not differ significantly from those of the control group. The graphical representation in Figure 4 illustrates this pattern, showing measurable gains in both groups across the two measurement points, while the developmental trajectories of the groups did not differ significantly.

This result can be interpreted in different ways considering the literature. Studies on educational robotics and early childhood programming have repeatedly shown that robotics-based learning environments can effectively stimulate computational thinking skills, such as sequencing, algorithmizing, and debugging, and this is especially true when the environments are problem-oriented, collaborative, and playful (Bahr, 2025; Bati, 2021; Falloon, 2024; Komis & Misirli, 2011, 2012; Paraskevopoulou-Kollia et al., 2025; Yang et al., 2022). The series of lessons used in this study followed this approach. Children worked in small groups to solve real-world problems, plan routes, arrange colour codes, create sequences, observe the robot's behaviour, and correct errors. The zoo task described in the Intervention Section illustrates how decomposition, abstraction, evaluation, and generalization were addressed in a narrative context.

It should be noted, however, that the intervention was limited to six units and was embedded in an already learning-intensive first school year. Studies using structured, playful robotics interventions with children in preschool and Grades 1–3 have frequently targeted and, in several cases, improved in computational thinking-related sub-skills (systematic overview: Paraskevopoulou-Kollia et al., 2025). In comparison with other intervention studies in this area (e.g., Constantinou & Ioannou, 2018), the intervention implemented in this study was intentionally short (six 45 min units over four weeks) and integrated into regular first-grade instruction, situating it at the shorter and more ecologically realistic extreme of the of intervention spectrum reported in the literature.

In this study the control group received regular instruction in which problem solving, pattern recognition, and basic cognitive strategies were addressed in subjects such as mathematics (current topic: addition in the twenties, number decomposition, and structured

problem solving using discovery packs). Thus, computational thinking-related skills may have been implicitly strengthened in these subjects as well. The observed performance increase in both groups can therefore be understood as an expression of a general increase in skills over the course of the survey period. This increase was not exceeded by the specific robotics intervention to a statistically verifiable extent; longer intervention periods and broader samples would be required for that. It is also important to consider possible ceiling effects related to the test instrument used (Döring, 2023), as the students achieved high scores already on the pre-test and the instrument may therefore have had limited sensitivity for further gains.

Regarding intervention effects, the present finding that the experimental group showed a marginally, but not statistically significant, larger gain in comparison to the control group aligns with certain aspects of extant literature. A limited number of short-term robotics interventions in analogous age groups have documented substantial impacts on particular computational thinking subskills in select cases (e.g., Constantinou & Ioannou, 2018; Knight et al., 2019), while systematic reviews also indicate that the overall evidence base is characterized by rather short and small-scale studies and that robust conclusions about sustained gains in computational thinking require longer, more strongly curriculum-embedded, and longitudinally investigated programs (Paraskevopoulou-Kollia et al., 2025). The present study aligns more closely with the latter interpretation, particularly in the light of the comparatively high baseline level observed in the sample and the limited duration of the intervention.

Overall, the findings suggest that the series of lessons using the intelino Smart Train stimulated computational thinking-related ways of thinking and acting. However, in the present duration, scope, and sample constellation, it did not lead to a higher increase in performance on the test than regular lessons. To promote computational thinking sustainably and measurably, it appears advisable to integrate educational robotics-based learning opportunities into the curriculum over a longer period, as discussed in the Theoretical Background Section in relation to international developments and curriculum recommendations.

Research question 3: Do the computational thinking test results show any indications of an increase in computational thinking skills depending on gender?

The results show that there are no statistically significant differences in performance between male and female students on either the pre- or post-test. In addition, the increase in performance between the two measurement points shows no gender-specific differences in the overall sample or within the intervention group. These results therefore indicate that no statistically significant differences in computational thinking between male and female students were detectable after the intervention. Both genders showed comparable skill levels in the tested skills in both the experimental and control groups.

These results confirm the international findings reported in the Theoretical Background Section: there are no systematic gender differences in computational thinking skills in early childhood (Relkin et al., 2020; Bati, 2021). This finding is particularly relevant given widespread gender stereotypes. Specifically, the assumption that male students are more interested in technology and therefore perform better in computational thinking is not supported by international studies (Master et al., 2021, 2023) or by the available data. The present study can thus be read as adding a German primary education data point to a broader international pattern. Rather, the results suggest that, in early elementary school, computational thinking is an area in which male and female students perform similarly and benefit equally from learning settings.

5.2. Study Limitations

Several limitations should be considered when interpreting the findings of this study.

First, the sample was limited to 66 first graders from a single elementary school with an average school social index. In addition, the classes were not randomly assigned to the experimental and control groups, which limits the generalizability of the findings and means that class- or school-related confounding variables cannot be fully ruled out (Döring, 2023). At the same time, the pre-test results show no significant differences between the experimental and control groups, indicating comparable initial conditions within the scope of this study.

Second, a quasi-experimental pre-test/post-test design with only two measurement points was used in this study, allowing for conclusions about short-term changes in computational thinking skills; however, it does not permit any statements about the stability of these changes or about medium- and long-term skill development. A follow-up study with an additional measurement point would therefore be desirable (Döring, 2023).

Third, the findings must be interpreted considering the assessment instrument used. TechCheck-1 captures computational thinking through standardized test performance, but it does not directly assess process-related aspects such as problem-solving strategies, reflection processes, or collaborative negotiation. However, such dimensions are central to a broader understanding of skill-oriented assessment and should ideally be complemented by additional formats, such as observations or performance-based tasks (Baartman et al., 2007). As a result, differences in how the children approached tasks during the intervention may not be fully reflected in the test scores. In addition, the relatively high pre-test scores and limited point range of the instrument may have produced ceiling effects, which may help explain why no statistically significant advantage of the intervention over regular instruction was found, although both groups improved between the pre- and post-tests. The observed pattern therefore points less to an absence of learning and more to the difficulty of detecting intervention-specific effects under the given design, sample, and measurement conditions.

Fourth, the intervention was limited to six teaching units and was implemented by a single teacher–researcher. Under these conditions, only a relatively short intervention period could be examined, and possible teacher-related effects cannot be excluded. At the same time, computational thinking development in early primary education is generally understood as a longer-term learning process that requires repeated opportunities for problem solving, reflection, and structured practice. The non-significant difference between the intervention and control groups should therefore be interpreted cautiously and should not be understood as evidence that the intelino Smart Train is ineffective in general. Rather, the findings reflect the effects of a short-term intervention conducted under specific school-based conditions and in the context of potential ceiling effects.

6. Conclusions

This study examined the computational thinking skills of German first graders using TechCheck-1 and investigated whether a six-unit intervention with the intelino Smart Train learning robot led to stronger skill gains than regular instruction. The findings show that the participating children achieved pre-test scores that exceeded published reference values, with no statistically significant differences between male and female students or between the experimental and control groups at the beginning of the study.

Throughout the study period, both the experimental and control groups improved their test performance. However, the intervention group did not show a statistically significant advantage over the control group in the post-test performance or skills development between the two measurement points. The results therefore suggest that the short-term robotics-based intervention did not lead to measurable gains beyond those observed in regular classroom instruction under the given conditions.

At the same time, the findings indicate that computational thinking can be taught meaningfully in early primary education and that first graders are able to work with age-appropriate tasks related to sequencing, problem solving, and debugging. In addition, the absence of gender-related differences in the pre-test, post-test, and change scores supports the assumption that computational thinking learning opportunities in the first years of schooling can be designed in ways that are accessible to all children. Overall, this study contributes empirical evidence on the computational thinking skills of first graders in a German-speaking context and highlights important challenges for future intervention research, particularly regarding the sample size, duration of intervention, and sensitivity of assessment instruments. The present study contributes the extant literature on educational robotics interventions in early primary education in three ways. First, it provides one of the initial quasi-experimental data points for first graders' computational thinking in a German-speaking context, employing an instrument that has been validated on an international scale. Second, the study illustrates the limitation of short, isolated robotics interventions when initial levels are already comparatively high. Third, it contributes to the existing body of evidence that early gender differences in computational thinking are not a robust phenomenon and should not be assumed in pedagogical design. The findings therefore support a cautious but constructive view: educational robotics may offer meaningful learning opportunities in early primary education; however, stronger evidence for intervention-specific effects will require longer-term studies, broader samples, and more differentiated assessment approaches.

7. Future Research Perspectives

The findings of this study suggest several directions for future research. First, future studies should contribute to the further development of computational thinking assessment in early primary education. The present results show comparatively high pre-test scores and performance gains in both groups, while no statistically significant difference in skills development between the experimental and control groups could be detected. Against this background, future research should design and use instruments that differentiate more clearly across both the lower and upper performance ranges and are sensitive enough to capture smaller intervention-related changes. In addition, future instruments should reflect different facets of computational thinking more adequately, including more recent conceptual developments in the field (CT 2.0/Tedre, 2022), and should be complemented by process-oriented data, such as observations, learning products, or interviews, in order to gain a deeper understanding of children's strategies and learning processes (Baartman et al., 2007).

Second, future studies should investigate longer-term and more systematically embedded interventions. In this study, the intervention comprised only six teaching units, and the findings indicate that this short-term robotics-based learning arrangement did not lead to measurable gains beyond those observed in regular classroom instruction. Future research should therefore examine whether computational thinking develops more clearly when educational robotics activities are implemented over longer periods of time, or even across several school years, and are more closely integrated into the curriculum (Spörer & Glaser, 2010). It would be worthwhile to examine teaching arrangements in which age-appropriate robotics, unplugged activities, and other forms of programming-related learning are linked over a longer developmental period.

Third, future research should focus not only on test performance but also on broader learning processes and possible transfer effects. The present study concentrated on computational thinking performance as measured by TechCheck-1; however, it remains unclear how robotics-based learning environments may influence related dimensions such as

problem-solving strategies, debugging practices, persistence, collaboration, and motivational orientations. Longitudinal studies would be particularly valuable for investigating whether early support in computational thinking has sustained effects beyond the immediate intervention period and whether such learning opportunities also affect related domains of learning and development.

Fourth, the absence of statistically significant gender differences in the pre-test, post-test, and change scores suggests that computational thinking learning opportunities in early primary education can be designed in ways that are accessible to all children. Future research could build on this finding by more closely examining which characteristics of learning environments support broad participation, such as narrative framing, task design, collaboration formats, or role distribution in group work. In this way, further studies could contribute to a more differentiated understanding of how computational thinking can be promoted without reproducing gender stereotypes.

Finally, future research should pay greater attention to the implementation conditions. Because the present study was conducted as a short-term, school-based intervention with one teacher and one specific learning robot, it remains unclear how robust such findings are across different teachers, school contexts, and didactic designs. Further studies should therefore investigate how teachers' professional knowledge, confidence, and classroom implementation shape the quality and effects of computational thinking learning opportunities in early primary education (Fehrmann, 2024; Fehrmann & May, 2025).

Funding: This research received no external funding.

Institutional Review Board Statement: Unlike in some other countries, there is no general legal obligation in German empirical educational research to obtain a formal ethics vote (IRB statement) for studies involving schoolchildren. Research in schools is primarily subject to the school approval procedures of the federal states, the research ethics guidelines of the DGfE, and, at the national level, to RdErl. d. Ministerium für Schule und Weiterbildung v. 15.07.1996 (GABl. NW. I S. 152)1, 10-45 Nr. 2 Wissenschaftliche Untersuchungen, Tests und Befragungen an Schulen gemäß § 120 Abs. 4 SchulG, Abs. 2 (https://bass.schule.nrw/102.htm) (accessed on 14 February 2026). For this knowledge study with elementary school students, the necessary approvals were obtained from the school authorities, and the informed consent of the legal guardians was requested by the school. Advice was also provided by the data protection officer at the University of Münster.

Informed Consent Statement: Informed consent was obtained from all subjects involved in this study.

Data Availability Statement: The dataset and raw data supporting the conclusions will be made available upon request to the authors.

Acknowledgments: Many thanks to Ina Selter for her support in designing and conducting the intervention and supporting the data collection as part of her master's thesis. The teaching materials generated by Ina Selter and Raphael Fehrmann, which were used in this intervention, can be found as Open Educational Resources at https://go.rfehrmann.de/intelino25 (ZIP) (accessed on 14 February 2026).

Conflicts of Interest: The author declares no conflict of interest.

Notes

[1] The test presented here is now called TechCheck-1 and is aimed explicitly at first-grade students. In its original version, it was called TechCheck (Relkin et al., 2020). The instrument can be downloaded at https://sites.bc.edu/devtech/assessments/downloads/techcheck-downloads/ (accessed on 14 February 2026), translation of the test into German: Fehrmann (2026).

[2] The values from Relkin et al. (2020) used for comparison here are not purely pre-level values but were collected during an ongoing intervention, according to the authors.

References

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, *55*(7), 832–835. [CrossRef]

Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in Human Behavior*, *105*, 105954. [CrossRef]

Baartman, L. K. J., Bastiaens, T. J., Kirschner, P. A., & van der Vleuten, C. P. M. (2007). Evaluating assessment quality in competence-based education: A qualitative comparison of two frameworks. *Educational Research Review*, *2*(2), 114–129. [CrossRef]

Bahr, T. (2025). Evaluating computational thinking practices in early childhood education: Insights from an intervention bridging research and practice. In *Proceedings of the 30th ACM conference on innovation and technology in computer science education, Nijmegen, The Netherlands, June 27–July 2* (pp. 653–659). ACM. [CrossRef]

Bati, K. (2021). A systematic literature review regarding computational thinking and programming in early childhood education. *Education and Information Technologies*, *27*, 2059–2082. [CrossRef]

Bers, M. U. (2018a). Coding and computational thinking in early childhood: The impact of ScratchJr in Europe. *European Journal of STEM Education*, *3*(3), 8. [CrossRef]

Bers, M. U. (2018b). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge. [CrossRef]

Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., Earp, J., Horvath, M. A., Jasutė, E., Malagoli, C., Masiulionytė-Dagienė, V., & Stupurienė, G. (2022). *Reviewing computational thinking in compulsory education*. Publications Office of the European Union. [CrossRef]

Bollin, A. (2016). COOLeInformatik. *OCG Journal*, *2*, 28.

Bruner, J. (1971). Über kognitive Entwicklung. Über Darstellung. In J. Broner, H. Aebli, & J. Hornsby (Eds.), *Studien zur kognitiven Entwicklung. Eine kooperative Untersuchung am center for cognitive studies der Harvard-Universität*. Ernst Klett Verlag.

Buitrago-Flórez, F., Danies, G., Restrepo, S., & Hernández, C. (2021). Fostering 21st century competences through computational thinking and active learning: A mixed method study. *International Journal of Instruction*, *14*(3), 737–754. [CrossRef]

Cansu, F., & Cansu, S. (2019). An overview of computational thinking. *International Journal of Computer Science Education in Schools*, *3*, 17–30. [CrossRef]

Chakraborty, P. (2024). Computer, computer science, and computational thinking: Relationship between the three concepts. *Human Behavior and Emerging Technologies*, *2024*, 5044787. [CrossRef]

Ching, Y., & Hsu, Y. (2023). Educational robotics for developing computational thinking in young learners: A systematic review. *Techtrends*, *68*, 423–434. [CrossRef] [PubMed]

Constantinou, V., & Ioannou, A. (2018, September 3–5). *Development of computational thinking skills through educational robotics*. 13th European Conference on Technology Enhanced Learning, Leeds, UK. Available online: https://ceur-ws.org/Vol-2193/paper9.pdf (accessed on 14 February 2026).

Das, S., & Mitra, A. (2024). Enhancing logic and problem-solving skills in the digital age. In M. Fonkam, & N. Vajjhala (Eds.), *Revolutionizing curricula through computational thinking, logic, and problem solving* (pp. 40–59). IGI Global Scientific Publishing. [CrossRef]

Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, *60*(6), 33–39. [CrossRef]

Döring, N. (2023). *Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften*. Springer. [CrossRef]

Eder, S., Lehmann, A., Lenich, A., Roboom, S., Seiler, G., & Wentzel, J. (2013). *Medienkompetenz-Kitas NRW. Ein Modellprojekt der Landesanstalt für Medien Nordrhein-Westfalen (LfM)*. Landesanstalt für Medien Nordrhein-Westfalen (LfM). Available online: https://www.medienanstalt-nrw.de/fileadmin/user_upload/materials_and_ordering_system/download/L155_Medienkompetenz-Kitas-NRW.pdf (accessed on 14 February 2026).

Eickelmann, B., Bos, W., Gerick, J., Goldhammer, F., Schaumburg, H., Schwippert, K., Senkbeil, M., & Vahrenhold, J. (Eds.). (2019). *ICILS 2018 #Deutschland: Computer- und informationsbezogene kompetenzen von schülerinnen und schülern im zweiten internationalen vergleich und kompetenzen im Bereich computational thinking*. Waxmann. Available online: https://www.pedocs.de/volltexte/2019/18166/pdf/Eickelmann_et_al_2019_ICILS_2018_Deutschland.pdf (accessed on 14 February 2026).

Eickelmann, B., Fröhlich, N., Bos, W., Gerick, J., & Goldhammer, F. (Eds.). (2024). *ICILS 2023 #Deutschland: Computer- und informationsbezogene Kompetenzen und Kompetenzen im Bereich computational thinking von Schüler*innen im internationalen Vergleich*. Waxmann. Available online: https://www.pedocs.de/volltexte/2024/32046/pdf/Eickelmann_et_al_2024_ICILS_2023_Deutschland.pdf (accessed on 14 February 2026).

Erwinsyah, A., Yusuf, F. M., Laliyo, L. A., Mursalin, R., & Riumkina, I. (2025). Research trends of computational thinking for advancing sustainable development goals (SDGs) in science learning: Bibliometric analysis. *Jurnal Pendidikan IPA Indonesia*, *14*(2), 23645. [CrossRef]

European Commission. (2016). *Developing computational thinking in compulsory education—Implications for policy and practice* (P. Kampylis, & Y. Punie, Eds.). Publications Office. Available online: https://data.europa.eu/doi/10.2791/792158 (accessed on 14 February 2026).

Fadel, C., Black, A., Taylor, R., Slesinski, J., & Dunn, K. (2024). *Education for the age of AI*. Center for Curriculum Redesign.

Falloon, G. (2024). Advancing young students' computational thinking: An investigation of structured curriculum in early years primary schooling. *Computers & Education*, *216*, 105045. [CrossRef]

Fehrmann, R. (2024). *Professionelle digitale Kompetenz bei Lehramtsstudierenden fördern! Wie kann computational thinking durch den Einsatz von Bildungsrobotik in der Hochschullehre vermittelt werden? Wissenschaftliche Schriften der Universität Münster VI, 26*. Tredition. [CrossRef]

Fehrmann, R. (2026). *Übersetzung des computational thinking-tests: TechCheck-1, im englischsprachigen Original entwickelt von E. Relkin, L. de Ruiter, M. U. Bers*. Available online: https://go.rfehrmann.de/2026a (accessed on 14 February 2026).

Fehrmann, R., & May, D. (2025). Die Grundlagen künstlicher Intelligenz durch computational thinking lernförderlich ab der Grundschule nachvollziehen. Wie der Lernroboter Ozobot Problemlösekompetenz fördern und ein Verständnis algorithmischer Funktionsweisen anbahnen kann. *Zeitschrift für Begabungsforschung und Talententwicklung*, *2*(1), 64–80. [CrossRef]

Flanagan, T., Wong, G., & Kushnir, T. (2023). The minds of machines: Children's beliefs about the experiences, thoughts, and morals of familiar interactive technologies. *Developmental Psychology*, *59*(6), 1017–1031. [CrossRef]

GI/Gesellschaft für Informatik. (2016). *Dagstuhl-Erklärung. Bildung in der digitalen vernetzten Welt*. Available online: https://gi.de/fileadmin/GI/Hauptseite/Themen/Dagstuhl-Erkla__rung_2016-03-23.pdf (accessed on 14 February 2026).

Gollwitzer, M., Eid, M., & Schmitt, M. (2017). *Statistik und Forschungsmethode*. Beltz.

ISB/Staatsinstitut für Schulqualität und Bildungsforschung. (2022). *Digitale Bildung 1–12*. Available online: https://www.lehrplanplus.bayern.de/fachlehrplan/foerderschule/1/db/foerderschwerpunkt/geistige-entwicklung/db#296156 (accessed on 14 February 2026).

Jiménez, M., Zapata-Cáceres, M., Román-González, M., Robles, G., Moreno-León, J., & Martín-Barroso, E. (2024). Computational concepts and their assessment in preschool students: An empirical study. *Journal of Science Education and Technology*, *33*, 998–1020. [CrossRef]

Kallia, M., van Borkulo, S. P., Drijvers, P., Barendsen, E., & Tolboom, J. (2021). Characterising computational thinking in mathematics education: A literature-informed Delphi study. *Research in Mathematics Education*, *23*(2), 159–187. [CrossRef]

KMK/Kultusministerkonferenz. (2016). *Bildung in der digitalen Welt. Strategie der Kultusministerkonferenz*. Available online: https://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/2016/2016_12_08-Bildung-in-der-digitalen-Welt.pdf (accessed on 14 February 2026).

Knight, V. F., Wright, J., & DeFreese, A. (2019). Teaching robotics coding to a student with ASD and severe problem behavior. *Journal of Autism and Developmental Disorders*, *49*(6), 2632–2636. [CrossRef]

Komis, V., & Misirli, A. (2011). Robotique pédagogique et concepts préliminaires de la programmation à l'école maternelle—Une étude de cas basée sur le jouet programmable Bee-Bot. In G.-L. Baron, E. Bruillard, & V. Komis (Eds.), *Sciences et technologies de l'information et de la communication (STIC) en milieu éducatif—Analyse de pratiques et enjeux didactiques, Actes du quatrième colloque international DIDAPRO 4—Dida&STIC* (pp. 271–281). New Technologies Editions. Available online: https://edutice.archives-ouvertes.fr/edutice-00676143/document (accessed on 14 February 2026).

Komis, V., & Misirli, A. (2012). L'usage des jouets programmables à l'école maternelle—Concevoir et utiliser des scenarios pédagogiques de robotique éducative. *Revue Scholé*, *17*(1), 143–154.

Lavigne, H. J., Presser, A. L., Rosenfeld, D., Cuellar, L., Vidiksis, R., Ferguson, C., Wolsky, M., & Andrews, J. (2023). Computational thinking with families: Studying an at-home media intervention to promote joint media engagement between preschoolers and their parents. *Early Childhood Research Quarterly*, *65*, 102–114. [CrossRef]

Lepold, M., & Ullmann, M. (2018). *Digitale Medien in der Kita*. Herder.

LIBRA/Landesinstituts Brandenburg für Schule und Lehrkräftebildung. (n.d.). *Basiscurriculum Medienbildung*. Available online: https://bildungsserver.berlin-brandenburg.de/bcmedienbildung (accessed on 14 February 2026).

Master, A., Meltzoff, A. N., & Cheryan, S. (2021). Gender stereotypes about interests start early and cause gender disparities in computer science and engineering. *Proceedings of the National Academy of Sciences of the United States of America*, *118*(48), e2100030118. [CrossRef]

Master, A., Tang, D., Forsythe, D., Alexander, T., Cheryan, S., & Meltzoff, A. N. (2023). Gender equity and motivational readiness for computational thinking in early childhood. *Early Childhood Research Quarterly*, *64*(3), 242–254. [CrossRef]

Medienberatung NRW. (2018). *Medienkompetenzrahmen NRW (MKR)*. Kompetenzraster. Available online: https://medienkompetenzrahmen.nrw/fileadmin/pdf/LVR_ZMB_MKR_Rahmen_A4_2020_03_Final.pdf (accessed on 14 February 2026).

Medienberatung NRW. (2020). *Lehrkräfte in der digitalisierten Welt. Orientierungsrahmen für die Lehrerausbildung und Lehrerfortbildung in NRW*. Available online: https://www.schulministerium.nrw/system/files/media/document/file/lehrkraefte_digitalisierte_welt_2020.pdf (accessed on 14 February 2026).

Misirli, A., & Komis, V. (2014). Robotics and programming concepts in early childhood education. In K. Charalampos, P. Panagiotis, & K. Ilias (Eds.), *Research on e-learning and ICT in education* (pp. 99–118). Springer.

Misirli, A., & Komis, V. (2023). Computational thinking in early childhood education: The impact of programming a tangible robot on developing debugging knowledge. *Early Childhood Research Quarterly*, *65*(4), 139–158. [CrossRef]

MSB/Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen. (2023). *Der schulscharfe Sozialindex*. Available online: https://www.schulministerium.nrw/schulsozialindex (accessed on 14 February 2026).

MSB/Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen & MKFFI/Ministerium für Kinder, Familie, Flüchtlinge und Integration des Landes Nordrhein-Westfalen. (2018). *Bildungsgrundsätze für Kinder von 0 bis 10 Jahren in Kindertagesbetreuung und Schulen im Primarbereich in Nordrhein-Westfalen*. Available online: https://www.kita.nrw.de/system/files/media/document/file/Bildungsgrundsaetze_Stand_2018.pdf (accessed on 14 February 2026).

Palts, T., & Pedaste, M. (2020). A model for developing computational thinking skills. *Informatics in Education*, *19*, 113–128. [CrossRef]

Papadakis, S., & Kalogiannakis, M. (2020). Learning computational thinking development in young children with bee-bot educational robotics. In M. Kalogiannakis, & S. Papadakis (Eds.), *Handbook of research on tools for teaching computational thinking in P-12 education* (pp. 289–309). IGI Global Scientific Publishing. [CrossRef]

Paraskevopoulou-Kollia, E., Michalakopoulos, C., Zygouris, N., & Bagos, P. (2025). Computational thinking in primary and pre-school children: A systematic review of the literature. *Education Sciences*, *15*(8), 985. [CrossRef]

Pollak, M., & Ebner, M. (2019). The missing link to computational thinking. *Future Internet*, *11*(12), 263. [CrossRef]

Ramírez-Montoya, M. S., Buenestado-Fernandez, M., & Ibarra-Vazquez, G. (2023). Unlocking sustainable development goals through computational thinking: A search to inform computers education from citizen science dataset. In *The 8th international conference on information and education innovations (ICIEI 2023), Manchester, UK, April 13–15* (p. 7). ACM. [CrossRef]

Relkin, E., De Ruiter, L., & Bers, M. (2020). TechCheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology*, *29*, 482–498. [CrossRef]

Resnick, M. (2017). *Lifelong kindergarten—Cultivating creativity through projects, passion, peers, and play*. The MIT Press.

Schräpler, J. P., & Jeworutzki, S. (2021). *Konstruktion des Sozialindex für Schulen in Nordrhein-Westfalen*. ZEFIR. [CrossRef]

Screpanti, L., Miotti, B., & Monteriù, A. (2021). Robotics in education—A smart and innovative approach to the challenges of the 21st century. In S. David, G. Lorenzo, S. Margherita, M. Beatrice, M. Andrea, & B. Paulo (Eds.), *Makers at school, educational robotics and innovative learning environments* (pp. 17–27). Springer.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*(1), 142–158. [CrossRef]

Spörer, N., & Glaser, C. (2010). Förderung selbstregulierten Lernens im schulischen Kontext. *Zeitschrift für Pädagogische Psychologie*, *24*(3–4), 171–175. [CrossRef]

Su, J., & Yang, W. (2023). A systematic review of integrating computational thinking in early childhood education. *Computers and Education Open*, *4*, 100122. [CrossRef]

Sung, J. (2022). Assessing young Korean children's computational thinking: A validation study of two measurements. *Education and Information Technologies*, *27*, 12969–12997. [CrossRef]

Tedre, M. (2022). Computational thinking 2.0. In *Proceedings of the 17th workshop in primary and secondary computing education* (Vol. 1, pp. 1–2). ACM. [CrossRef]

Teuber, K., Dindarian, A., & Ekaterina, N. C. N. (2022). Künstliche Intelligenz und ihre Anforderungen an den Kompetenzerwerb. In R. Knackstedt, J. Sander, & J. Kolomitchouk (Eds.), *Kompetenzmodelle für den digitalen Wandel. Kompetenzmanagement in Organisationen*. Springer. [CrossRef]

Tsai, M., Liang, J., & Hsu, C. (2020). The computational thinking scale for computer literacy education. *Journal of Educational Computing Research*, *59*, 579–602. [CrossRef]

Valdes, G., Figueredo, Ó., & Sánchez, A. (2025). Integrating computational thinking in children aged 3 to 6: Challenges and opportunities in early childhood education. *Frontiers in Education*, *10*, 1535135. [CrossRef]

Weber, A., & Greiff, S. (2023). ICT skills in the deployment of 21st century skills: A (cognitive) developmental perspective through early childhood. *Applied Sciences*, *13*(7), 4615. [CrossRef]

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, *25*, 127–147. [CrossRef]

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

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A*, *366*(1881), 3717–3725. [CrossRef] [PubMed]

Xing, G., Cady, A., & Wang, X. (2025). Playful computational thinking learning in and beyond early childhood classrooms: Insights from collaborative action research of two teacher-researchers. *Education Sciences*, *15*(7), 840. [CrossRef]

Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for All: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, *60*, 565–568. [CrossRef]

Yang, W., Ng, D., & Gao, H. (2022). Robot programming versus block play in early childhood education: Effects on computational thinking, sequencing ability, and self-regulation. *British Journal of Educational Technology*, *53*, 1817–1841. [CrossRef]

Zeng, Y., Yang, W., & Bautista, A. (2023). Computational thinking in early childhood education: Reviewing the literature and redeveloping the three-dimensional framework. *Educational Research Review*, *39*, 100520. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.