Education. Innovation. Diversity
Vol. 1, No. 12 (2026)
ISSN 2661-5401

https://doi.org/10.17770/eid2026.1.107

Programming Education for Students with Autism: A Scoping Review of Pedagogical Strategies and Tools Between 2017–2025

Anastasios Topaloglou[1], Evaggelia Tsiomi [2], Andromachi Nanou[3], Dimitris Karampatzakis[4]

[1, 2, 3, 4] Department of Informatics, Democritus University of Thrace, Greece

Abstract. This paper explores the educational landscape for children and youth with Autism Spectrum Disorder (ASD), specifically focusing on the efficacy of block-based and visual programming environments. Reviewing sixteen empirical studies published between 2017 and 2025, this analysis examines how programming interventions were designed, delivered, and evaluated for neurodivergent learners. Results indicated an overwhelming focus on such tools as Scratch and Blockly because their syntactic complexity is low and a visual interface can be made explicit to relate to cognitive profiles of autistic students. Successful interventions have included the use of scaffolding support, visual supports, adjusted pacing support, and inclusive group strategies leading to engagement and learning outcomes. Gaps of enormous magnitude are in educator training, accessibility design, and cultural diversity because most studies were situated within high-income countries. There is inadequate research on the attitudes of stakeholders and longitudinal impacts on their lives from learning how to program. The review prescribes a holistic ecosystem-oriented intervention inclusive of technology that is pedagogy-driven by evidence and structured teacher support for equitable access to develop computational thinking skills to be consciously piloted as future research beyond tools and activities that address the systemic barriers to participation in computing education by autistic learners.

Keywords: Autism Spectrum Disorder (ASD); block-based programming; computational thinking; inclusive education; programming education; visual programming.

Introduction

Programming and computational thinking are the core skills of the 21st century in an increasingly high-tech world. Programming education encourages problem-solving or logical reasoning, creativity, autonomy, and self-expression (Bers, 2020). As programming is initiated at increasingly earlier stages of education and has made its way into primary and secondary curricula across many countries in the world, questions regarding to what extent this process is accessible and inclusive have found growing space. In this regard, students with Autism Spectrum Disorders (ASD) have built up increasing interest due to their different learning profiles and more descried barriers that they face through traditional lessons of computing (Elliott, 2018; Ariza & Hernández Hernández, 2025).

Corresponding author Anastasios Topaloglou. E-mail: antopal@cs.duth.gr

ORCID
Anastasios Topaloglou https://orcid.org/0009-0002-1401-2523
Evaggelia Tsiomi https://orcid.org/0009-0004-7572-3323
Andromachi Nanou https://orcid.org/0009-0002-7294-6359
Dimitris Karampatzakis https://orcid.org/0000-0003-0203-0476

Scientific findings and clinical observations demonstrate that students with ASD often excel in activities that follow strict rules, as well as for their pleasure in understanding visual materials that match structured programming tasks (Baron-Cohen, 2006). Traditional methods of teaching programming, based on abstract syntax and rapid interaction and implicit social rules, hinder participation and the emergence of their strengths. Difficulties in communication, executive function, sensory processing skills and social interaction combined with the absence of appropriate teaching methods lead to the exclusion of autistic students from meaningful participation and progress in computer science subjects (Nanou & karampatzakis, 2023, Papavlasopoulou et al., 2019; Zubair et al., 2019; Pereira et al., 2023; Papavlasopoulou et al., 2020).

Researchers have studied block-based visual programming methods and educational games as more accessible ways to teach programming concepts to students who require different types of educational support. The system enables users to learn complex computational concepts through its visual syntax and drag-and-drop capabilities and its system provides direct feedback to users which helps them understand their progress through various stages of learning (Elshahawy et al., 2020; Elshahawy et al., 2022). The research studies demonstrate that these methods help students with ASD develop essential computational thinking abilities because they build skills needed for sequencing and loops and conditionals and they maintain student interest and motivation (Elshahawy et al., 2022; Özgüç & Altın, 2022; Madzhi et al., 2024).

Programming-based educational programs for autistic learners help students develop both technical skills and various life skills according to research findings (Elliott, 2018; Rogers et al., 2017; Özgüç & Altın, 2022, Nanou & Karampatzakis 2022, 2023). Self-efficacy and communication abilities and participation levels improved through game-based learning environments and structured programming tasks and technology-supported teaching methods according to research which showed that learning experiences need to be developed according to student interests while they need to be supported through proper learning frameworks which need to be integrated with their specific needs (Begel et al., 2021; Rogers et al., 2017; Madzhi et al., 2024). The research about inclusive educational environments and semi-structured educational programs demonstrates that students with ASD show better emotional control and persistence when they work within predictable task systems which educational systems establish through their teaching methods (Elliott 2018; Özgüç & Altın, 2022).

The movement towards inclusive education requires educational institutions to provide all students with access to educational programs which teach digital literacy and programming skills (Ariza & Hernández Hernández, 2025). The process of including students with ASD in mainstream computing classes presents educators with multiple difficulties. Educators face three main obstacles which include their limited training and low confidence level and their need for more institutional backing to create computing lessons that suit both technical requirements and social needs of neurodiverse students (Elliott, 2018; Rogers et al., 2017). Educational institutions need to carry out inclusive computing methods but they do so in a way that creates disjointed and inconsistent practices across different learning environments (Özgüç & Altın, 2022). However, research also shows that inclusive computing practices can be implemented effectively when they are structured and responsive to students' individual needs. For instance, (Özgüç & Altın, 2022) demonstrated that the use of explicit instruction, combined with guided and independent practice, enabled a student with ASD to progressively develop coding skills and achieve independence. The instructional process was carefully planned, continuously evaluated, and adapted based on the student's performance, ensuring consistency and coherence throughout the learning experience. This example highlights that inclusive computing does not have to be fragmented; rather, when supported by appropriate pedagogical strategies and systematic implementation, it can provide meaningful and effective learning opportunities for neurodiverse students.

The principles of Universal Design for Learning together with inclusive and accessible learning frameworks provide effective methods to assess how programming instruction should be tailored for various learner needs (Menziltsidou et al., 2025; Nanou & Karampatzakis 2022, 2023). The computing education framework requires multiple methods to show information and keep students engaged and allow them to demonstrate their knowledge (Zubair et al., 2019). Research about interest-driven activities that include game design and creative programming tasks shows that these methods help autistic learners maintain engagement while developing advanced thinking skills (Elshahawy et al., 2022). Despite increasing research about this topic programming education research for students with ASD remains disjointed. The current research studies show wide differences which include their teaching methods and their technological resources and their age range of students and their study environments that include special education classrooms and summer camps and online learning platforms (Begel et al., 2021; Elliott, 2018; Papavlasopoulou et al., 2019). Moreover, while several reviews have addressed related topics, relatively few have systematically synthesized recent empirical and review studies with a specific focus on programming pedagogy for autistic learners (Ariza & Hernández Hernández, 2025; Nanou & Karampatzakis, 2023).

Although the terms computing education and coding are often used interchangeably in educational discourse, this review adopts a more precise conceptual focus. In this study, programming education is understood as instructional interventions that explicitly engage learners in the design, construction, and execution of algorithms using programming languages or environments. Accordingly, the scope of the review is limited to programming-based educational interventions, rather than broader ICT, digital literacy, or general computing activities. More specifically, the review focuses on block-based and visual programming environments, which have been widely adopted in autism-inclusive educational contexts due to their reduced syntactic complexity, strong visual structure, and drag-and-drop interaction. The definition of programming-based educational interventions includes educational programs that teach students through structured programming activities which serve as their main learning method through programming activities. The review uses block-based programming pedagogy as its main focus which helps to define its research objective and maintain consistent understanding across all studied materials.

Across the reviewed literature, several consistent patterns emerge regarding the design and implementation of programming activities for students with ASD. Specifically, programming technologies are not used as standalone tools, but are embedded within structured pedagogical approaches that emphasize predictability, scaffolding, and individualized pacing. Additionally, these activities often serve multiple purposes beyond technical skill development, including supporting social communication, increasing engagement, and promoting participation in learning processes. At the same time, the literature reveals considerable variation in how these approaches are implemented, depending on the educational context, the technologies used, and the pedagogical frameworks adopted. These findings highlight both the potential of programming-based interventions and the need for a more systematic understanding of how such approaches can be effectively designed and applied in inclusive educational settings.

This paper addresses the research gap through a scoping review which analyzes 16 peer-reviewed studies that were published from 2017 until 2025 and investigate how students with ASD learn programming through block-based and visual programming methods. The study uses scoping review methodology to create a complete overview of existing research in the field by gathering all relevant information and creating a detailed research map. The studies under review evaluate various teaching methods which include serious games and explicit instruction and gamified learning platforms and social-skills-oriented curricula and design-based research in both inclusive and specialized educational environments. This review aims to discover instructional methods and educational tools that researchers use in classrooms with autistic students while describing their benefits and problems and showing how this research affects upcoming studies and educational methods in inclusive computing classrooms for autistic students.

Research Objectives and Questions

The aim of this scoping review is to explore implementation, and impact of programming activities and environments for children and adolescents with ASD, with a particular focus on block-based, visual, and game-based programming platforms. In order to guide the review process, two overarching research objectives were formulated, each supported by a set of specific research questions. These objectives also define the methodological categories for data extraction and synthesis.

Research Objective 1: To map the contextual characteristics that shape the design and application of programming interventions for autistic learners.

- Which programming platforms and tools are most employed in interventions for autistic students, and what adaptations or accessibility features make them suitable for addressing learners' cognitive, social, or sensory needs?
- In what educational contexts (e.g., formal classrooms, extracurricular programs, online environments) are programming activities implemented, and how do contextual factors such as infrastructure, teacher training, or cultural setting influence their effectiveness?

Research Objective 2: To investigate the pedagogical goals, methods, and outcomes of programming-based educational interventions for students with autism.

- What pedagogical goals and learning outcomes are pursued through programming interventions, and how are these supported by instructional strategies such as scaffolding, differentiated instruction, visual supports, or gamification?
- How do collaboration structures including peer interaction, teacher guidance, and participatory or co-design approaches shape autistic learners' engagement, and what benefits or challenges are reported in terms of cognitive, social, and emotional development?

Materials and Methods

A scoping review was conducted to create a comprehensive overview which establishes the range and types of available research studies about programming interventions for children and adolescents with Autism Spectrum Disorder who use block-based and visual programming methods. A scoping review was selected as the most suitable research method because the emerging interdisciplinary field exhibits diverse study designs together with different participant groups and various research results. Scoping reviews provide a comprehensive overview of current evidence in addition to identifying research gaps which help establish future research paths according to (Arksey & O'Malley, 2005) and (Levac et al., 2010).

The review methodology uses the (Arksey & O'Malley, 2005) five-stage process which includes the following five steps: (1) identifying the research questions, (2) identifying relevant studies, (3) selecting studies, (4) charting the data, and (5) collating, summarizing, and reporting the results. The process received further refinement through (Levac et al., 2010) recommendations which guided the team to develop deeper analysis capabilities while implementing their iterative teamwork system. The research used the Preferred Reporting Items for Systematic Reviews and Meta-Analyses extension for Scoping Reviews (PRISMA-ScR) checklist (Tricco et al., 2018) to maintain complete transparency and consistent reporting throughout all phases of the review process.

Methodology

A literature search for this review was conducted during March 2024. Three major electronic databases were searched, which included IEEE Xplore, Scopus and Google Scholar. To ensure a comprehensive and inclusive mapping of the existing literature, studies were considered regardless of publication type or geographical context. This approach is consistent with the aims of a scoping review, which seeks to capture the breadth and diversity of research activity across different settings and study designs. A comprehensive search strategy was designed which used various combinations of terms that matched autism educational programs and learning environment content. The search terms used were: (block AND based AND programming AND autism) OR (block AND based AND programming AND inclusion) OR (block AND based AND programming AND inclusion AND scratch) OR (block AND based AND programming AND scratch) OR (autism AND microbit). These terms were applied either individually or in grouped combinations, connected through the Boolean operator OR to ensure a comprehensive retrieval of relevant studies. The search string needed to be adapted for each database which required different indexing systems and advanced search functions. The three databases provided access to 544 articles for researchers to examine. The process left 416 unique records which remained after 128 duplicate records were eliminated from the database. The two reviewers conducted their screening process by applying specifically defined inclusion criteria which they marked as evident.

(1) the article had to be published in English,
(2) include participants diagnosed with ASD, and
(3) describe programming-related interventions, tools, or educational strategies involving programming or computational thinking.

The two reviewers conducted their screening process by independently reviewing the titles, abstracts, and keywords of each article according to the eligibility criteria which they had defined before. To ensure the reliability of the screening process, inter-rater agreement between the two reviewers was assessed using Cohen's kappa coefficient, which indicated a substantial level of agreement ($\kappa = 0.78$). The reviewers maintained consistent selection standards because they worked together to apply the criteria while using a programmed system to document their selection choices. The team reached agreement on all conflicting points after both parties brought their differences to discussion, and they used a third reviewer to decide which articles should be included in the study. A total of 354 records were excluded during the title and abstract screening stage, as they did not meet the predefined eligibility criteria. Following this initial screening, 62 articles were retained for full-text assessment. After a detailed full-text review, 46 studies were excluded for specific reasons: 29 studies were deemed not directly relevant to programming instruction or computational thinking, while 17 studies did not focus on educational research involving participants diagnosed with ASD. Ultimately, 16 studies met all inclusion criteria and were incorporated into the final synthesis, as illustrated in Figure 1. As this study follows a scoping review design with a descriptive objective, no formal qualitative assessment of study quality was conducted.

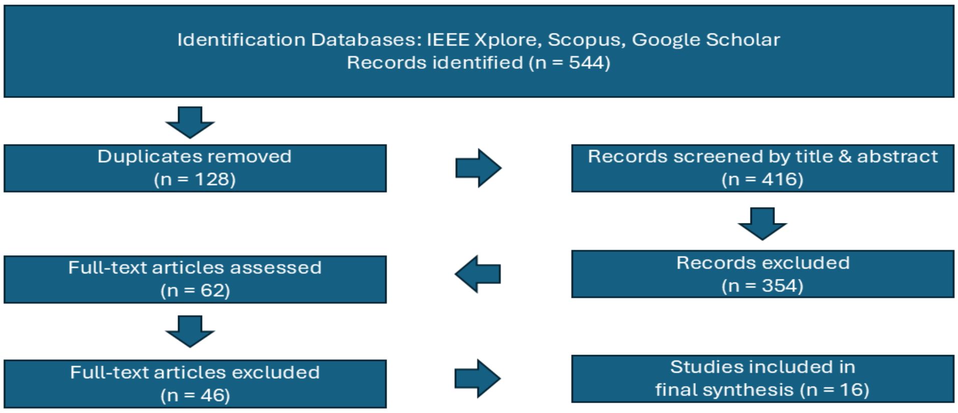

Figure 1. ***The selection process and each stage of the review***

Table 1 presents an overview of the sixteen studies included in this scoping review. The table summarizes key characteristics and analytical categories of the selected studies, including study type, participant groups, programming environments or technologies, educational contexts, primary pedagogical focus, and reported outcomes. This overview provides a structured mapping of the existing literature on programming-based educational interventions for students with Autism Spectrum Disorder (ASD) and serves as a reference framework for the thematic analysis presented in the subsequent subsections.

Table 1. ***Overview of the selected studies and analytical categories examined in the scoping review***

Study	**Participants**	**Research methodology**	**Study type**	**Primary pedagogical focus**	**Programming environment / technology**
Ariza & Hernández Hernández, 2025	39 studies involving students with disabilities (SWDs)	Systematic Literature Review (SLR) based on PRISMA	Systematic review	Inclusive STEM and programming interventions	Mixed programming and STEM technologies
Anderson & Donehower Paul, 2024	Elementary students with ASD and Special Education teachers	Mixed-methods design-based research (experience report with iterative curriculum development, observations, and focus groups)	Intervention study	Coding integrated with social skills development	Block-based programming
Begel et al., 2021	7 autistic first-year university students	Primarily qualitative (interviews, observations, reflections) with limited quantitative measures	Intervention (camp-based)	Communication and self-efficacy through coding	Game-based programming

Study	Participants	Research methodology	Study type	Primary pedagogical focus	Programming environment / technology
Madzhi et al., 2024	≈30–39 students with disabilities (13–15 y.o.)	Quantitative (survey-based)	Intervention study	STEM learning through robotics, motivation & collaboration	Zoom:Bit Robot, Micro:Bit, MakeCode
Elliott, 2018	68 teachers from SEND schools in England (with some follow-up interviews)	Mixed-methods (survey-based study)	Practice-based study	Pedagogical adaptation in computing education	Multiple computing tools
Elshahawy et al., 2020a	4 children with Autism Spectrum Disorder (mean age ≈ 10.25 years)	Experimental user study (qualitative evaluation)	Intervention study	Sequential programming through serious games	Block-based serious game
Elshahawy et al., 2022	6 children with autism (aged 10–14), divided into two groups	Experimental (comparative study using serious game vs block-based platform, with pre/post tests, engagement and usability evaluation)	Intervention study	Development of computational thinking skills	Gamified block-based platform
Elshahawy et al., 2020b	8 children with Autism Spectrum Disorder (aged 7–14 years)	Mixed-methods (design-based study with focus groups and survey evaluation)	Intervention study	Computational thinking and engagement	Serious game
Özgüç & Altın, 2022	1 student with ASD (10-year-old boy)	Qualitative (case study with explicit instruction, observations, interviews, and content analysis)	Single-case study	Explicit instruction in programming skills	Block-based programming
Nanou & Karampatzakis, 2023	45 studies (total n = 267 participants across studies)	Qualitative (scoping review with systematic search and narrative synthesis)	Scoping review	Participation and inclusion in educational robotics	Educational robotics

Study	Participants	Research methodology	Study type	Primary pedagogical focus	Programming environment / technology
Pereira et al., 2023	193 secondary students, 19 university students, plus teachers, researchers, and users with disabilities	Mixed methods (quantitative and qualitative, project-based and service-learning approach)	Case-based study	Inclusive STEM and programming projects	Programming projects
Papavlasopoulou et al., 2019	157 children aged 8–17 years (across three cycles in Norway)	Design-Based Research (DBR) with mixed methods	Design-based research	Learning experience and engagement	Block-based programming (Scratch)
Papavlasopoulou et al., 2020	149 K-12 students (aged 8–17 years)	Mixed-methods (eye-tracking, tests, interviews, observations)	Experimental study	Cognitive and behavioral learning processes	Visual programming
Rogers et al., 2017	Children (6–16 years), school students in workshops/classrooms	Mixed-methods (design-based study)	Conceptual study	Physical computing and learner motivation	micro:bit
Shah et al., 2024	36 computing teachers (6 interviews) with experience teaching autistic students	Mixed methods (literature review, survey, ethnography, and interviews)	Conceptual / empirical study	Pedagogical barriers and inclusive computing practices	Multiple computing tools
Zubair et al., 2019	7 experts (teachers and researchers in autism and special education)	Qualitative (semi-structured interviews and thematic analysis, with validation phase)	Design study	Accessibility and usability in programming tools	Visual programming tools

Results

Mapping the Contextual Dimensions That Affect the Participation of Students with Autism in Programming Interventions

The figures presented in this section are aligned with the research objectives. Figures 2, 3, and 4 address Objective 1 by illustrating the contextual and technological characteristics of the reviewed studies, while Figures 6, 7, and 8 address Objective 2 by focusing on pedagogical goals, collaboration structures, and instructional strategies.

Year and Country of Publication

16 studies were reviewed, all published from 2017 to 2025. It shows that even though it's pretty new, more and more research is coming out about teaching programming to kids with Autism Spectrum Disorder (ASD). As illustrated in Figure 2, early research activity between 2017 and 2019 was limited, reflecting an exploratory phase in the field. There's been a clear jump in publications starting in 2020. This increase suggests growing scholarly attention toward accessible and inclusive programming education for autistic learners. 2020 and 2024, where we managed to put out three studies in each of those years. It's 75 %. Most of the studies we looked at came from 2020 to 2024. This growth happened alongside bigger moves toward digital and

remote learning, which probably made more people interested in programming education that can change to fit what different students need. Research activity has remained relatively stable in recent years, indicating sustained interest rather than a short-term research trend. There are two studies come out in both 2021 and 2022. Then, there were two more in 2023, and the latest one popped up in 2025. Overall, the field of programming education for students with ASD appears to be both active and evolving, reflecting ongoing theoretical and pedagogical development.

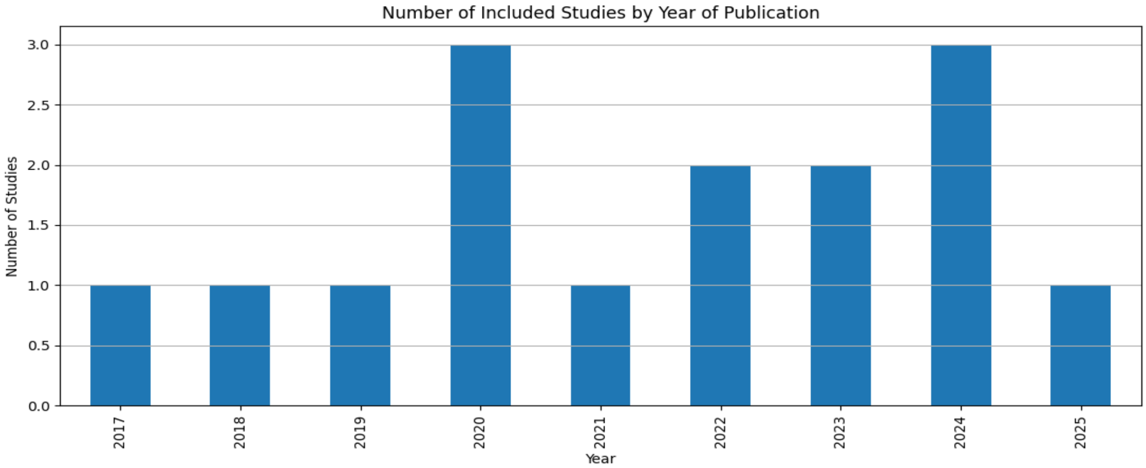

Figure 2. ***Distribution of Included Studies by Publication Year (2017–2025)***

Geographically, the studies show a variety of countries represented with either more than one publication or less than one publication (see Figure 3). The largest number of studies reported was from the United Kingdom with 4 studies (*n* = 4; 25.0 %), and the second largest was Egypt with 3 studies (*n* = 3; 18.75 %). 2 studies were reported from the United States and Norway (*n* = 2: Canada; 12.5%). The other countries represented by only 1 study included Greece, Spain, Turkey, Malaysia, and Colombia.

Although the majority of research on programming for autistic students was conducted within high-income countries, there is a considerable amount of research missing from the literature from low- and middle-income regions, which may suggest a lack of research or the ability to support these individuals in these regions (Ariza & Hernández Hernández, 2025; Nanou & Karampatzakis, 2023; Ehsan et al., 2018). Some factors that may limit inclusive programming include access to effective technology, quality teacher training, and lack of inclusive education-related behaviours (Elliott, 2018; Rogers et al., 2017; Zubair et al., 2019). The inclusive education-related policies must have been created and supported by national governments (Ariza & Hernández Hernández, 2025). Finally, although the literature indicates increasing numbers of scholarly inquiries over time, the geographic distribution of those inquiries continues to represent skewed distributions and many additional studies will still be needed to support individuals living in diverse socio-economic and cultural settings (Nanou & Karampatzakis, 2023; Papavlasopoulou et al., 2020).

However, most research purportedly exists within Europe and high resource regions. Research output across all countries or locations is often inconsistent, with only a few studies produced in middle-income regions. These situations illustrate the need for more research on programming interventions for autistic students in many different educational or socioeconomic situations. The degree of access to technology, teacher preparation, and national policy priorities will likely play a role in the implementation of inclusive programming practices and the amount of literature published related to those interventions. Overall, while the temporal trend represents an emerging and promising area of inquiry, it is necessary to include more geographic diversity within the literature to ensure future research findings are not disproportionately influenced by high resource educational systems.

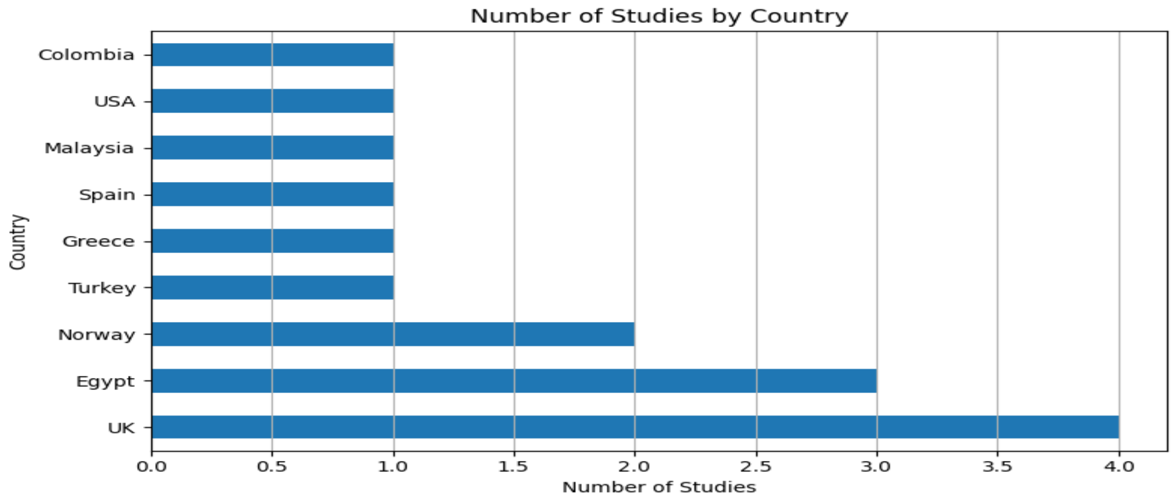

Figure 3. ***Geographical Distribution of Included Studies by Country of Publication***

Programming Technologies as Pedagogical Enablers in Autism-Inclusive Studies

The studies reviewed show that different programming technologies serve multiple instructional purposes; these include supporting structured and predictable learning environments, enhancing student engagement and motivation, facilitating communication and collaboration skills, and improving accessibility for learners with diverse needs (Elshahawy et al., 2020a; Anderson & Donehower Paul, 2024; Zubair et al., 2021; Rogers et al., 2017). As shown in Table 2 and Figure 4, the programming technologies documented in the literature do not point to a single generally accepted programming platform. Each technology has been chosen individually and adapted accordingly for the purpose of achieving various educational objectives: to increase access, to enhance interaction and participation, or to introduce participants to basic programming principles and concepts.

Studies where a single instance of instruction takes place use block-based environments to provide direct instruction and to enable individualized pacing based on the learners' needs (Özgüç & Altın, 2022). Studies that are based upon providing a combination of coding interventions and social skills interventions have been shown to utilize visual programming and robotics-based activities to support attainment of programming-based goals, as well as, to support the socio-communicative developmental skills (Anderson & Donehower Paul, 2024). Alternatively, studies located within the realm of physical computing and robotics develop, primarily, through the means of using tangible interaction (i.e., manipulation), in order to promote motivation, engagement, and collaboration among users, rather than through any systematic measure of programming / computational learning outcomes (Rogers & Bibby, 2017; Elliott, 2018; Nanou & Karampatzakis, 2023; Madzhi et al., 2024).

Further research focused on design and pedagogy indicates that programming education effectiveness for individuals with autism cannot be determined by technology choice alone. It is critical to include access needs, interface design principles, and uses of technology driven by pedagogy, with a focus on aligning technological capabilities with autism-informed instructional strategies (Zubair et al., 2019; Shah et al., 2024). Overall, it appears that programming technologies used in autism-focused education primarily serve as means to enable learning, with their pedagogical role varying by type of study and context, rather than being a determinant of learning outcomes on their own.

Table 2. ***Patterns of Technology Use in ASD-Focused Programming Studies***

Study type / pattern	Technologies involved	Role of technology	Evidence from studies
Autism-specific programming interventions	Custom serious games; block-based platforms	Structured task execution; predictable interaction; reduction of sensory load	Elshahawy et al., 2020a; Elshahawy et al., 2020b
Comparative technology studies	Serious games vs. Blockly-based platforms	Examination of feasibility and engagement; no clear superiority of tool	Elshahawy et al., 2022
Single-case instructional studies	Code.org; block-based environments	Explicit instruction; individualized pacing; functional skill development	Özgüç & Altın, 2022
Integrated coding & social skills interventions	Blockly-based robotics; visual programming	Parallel development of programming and social communication skills	Anderson & Donehower Paul, 2024
Physical computing in SEND contexts	micro:bit; simple robotics	Motivation; tangible interaction; engagement rather than CT assessment	Rogers & Bibby, 2017; Elliott, 2018
Robotics participation studies	Educational robots	Social participation; collaboration; not programming mastery	Nanou & Karampatzakis, 2023
Design and accessibility research	Visual programming tools (general)	Identification of accessibility requirements; interface design principles	Zubair et al., 2021
Pedagogical framework studies	Multiple computing tools	Pedagogy-driven use of technology; tool choice secondary	Shah et al., 2024

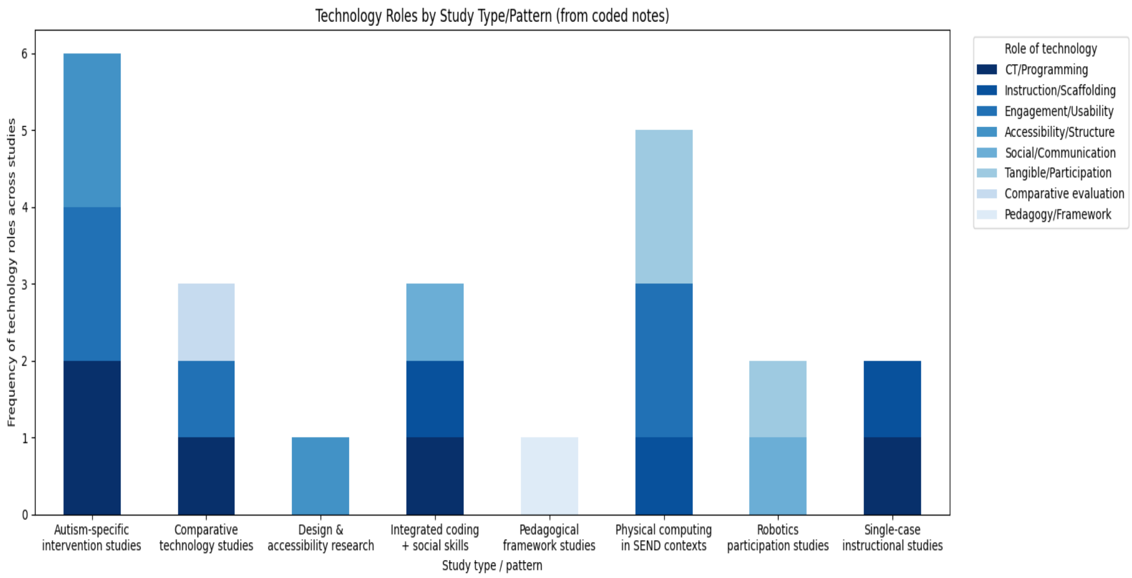

Figure 4. ***Overview of Programming Technologies and Platforms Employed in the Reviewed Studies***

In addition to the intervention-based evidence presented, six studies within the corpus provided critical conceptual and contextual insight to this review of programming education for autistic learners by articulating the pedagogical, institutional, and design conditions under which programming education for autistic learners is implemented and interpreted. Whereas these studies do not evaluate specific tools used to deliver programming interventions, they highlight the larger educational ecosystem in which programming interventions are situated.

Both foundational and practice-oriented research provide rationale for the implementation of programming interventions with autistic learners being based on pedagogical approaches, teacher practices, and learning structures that are dependent upon the use of physical computing, unplugged activities, task chunking, personalisation, and multimodal feedback (Rogers & Bibby, 2017; Elliott, 2018). The evidence from these studies indicates that instructional design has a much greater impact on both accessibility and engagement than the actual programming platform does. In addition, design-based research establishes a list of accessibility requirements and interface design principles that help create an autism-friendly visual programming environment focusing on predictability, reduced cognitive load and clear interaction (Papavlasopoulou et al., 2019; Zubair et al., 2019). Thus, programming technologies can be understood as enabling media whose effectiveness is mediated through alignment with design and pedagogy.

In summary, there has been an overall lack of knowledge regarding both programming and computational thought outcomes for people with autism as indicated by the disparities found in the literature across related fields pertaining to robotics education and inclusive education in STEM (Nanou & Karampatzakis, 2023; Ariza et al., 2025; Pereira et al., 2023). These findings were instrumental in guiding the focus of the current scoping review, which aims to analyze studies related to programming-based intervention. Collectively, these six studies strengthen the analytical foundation of the review by situating intervention findings within a wider pedagogical and research context, ensuring that interpretations of technology use are grounded not only in empirical outcomes but also in established principles of inclusive computing education.

Dimensions of Participation in Autism-Inclusive Programming Studies

The reviewed literature has emerged to support the idea that participation for students with Autism Spectrum Disorder (ASD) who participate in programming education is a multi-faceted construct as a result of both participating in direct instruction as well as other broader pedagogical and contextual factors. The sixteen studies from this scoping review demonstrate three inter-related dimensions of participation: 1) Direct participation through programming intervention; 2) Pedagogical and instructional supports to participate in programming activity; and 3) Contextual or access-related conditions that impact on student participation / engagement and sustainability.

Direct participation through programming interventions constitutes the core focus of ten studies. These intervention-oriented papers report the design and implementation of programming activities in which autistic learners actively engage with block-based environments, serious games, or integrated programming curricula. Interventions in this category commonly emphasize structured task sequences, visual representations, predictable interaction patterns, and incremental progression to support engagement and reduce cognitive and sensory demands (Elshahawy et al., 2020a; Elshahawy et al., 2020b; Elshahawy et al., 2022; Özgüç & Altın, 2022; Anderson & Donehower Paul, 2024; Begel et al., 2021). While some studies report preliminary improvements in computational thinking or programming-related understanding, the predominant outcomes relate to feasibility, motivation, sustained participation, and learners' confidence when engaging with programming activities (Madzhi et al., 2024).

An additional aspect of participation is related to pedagogical methods (also called didactic methods) designed to further enable student engagement with programming, even though they fall outside the main focus of intervention programme design. Several authors have identified forms of teacher mediation (Rogers & Bibby, 2017; Elliott, 2018; Shah et al., 2024), instructional scaffolding, personalisation, physical computing / unplugged computing activities and / or other supports as ways of facilitating participation by students on the autism spectrum to participate more fully in programming. This indicates that children's participation in programming does not just depend on the availability of programming technologies, but rather on how these activities are presented, adjusted and supported in their instructional environments.

The third dimension of contextual conditions and accessibility conditions indirectly shape how individuals participate. Design-centred research indicates a number of important considerations such as interface predictability, controlled complexity, multi-modal feedback and cognitive load that have been identified as prerequisites for engaging with visual programming environments in meaningful ways (Zubair et al., 2019). Furthermore, in reviewing educational robotics and inclusive STEM, there exists an abundance of literature that shows technology enhanced learning initiatives focused on social engagement and motivation while in many cases not paying attention to developing the programming skills of autistic individuals (Nanou & Karampatzakis, 2023; Ariza et al., 2025; Pereira et al., 2023). These perspectives help to contextualize the findings of the intervention and illustrate the need for programming focused interventions that specifically address issues of accessibility and participation. Evidence from the studies reviewed supports that programming participation of autistic individuals cannot solely be a function of technology-based interventions.

Mapping the Nature of Programming Activities in Relation to the Characteristics of Students with Autism That Participate

Programs and activities that are designed for students with autism spectrum disorder (ASD) have to take into consideration students' unique cognitive, social and sensory differences that affect how students participate in learning. Many students with autism spectrum disorder will also benefit from structured environments, visual supports and predictable routines and from being involved in learning activities where instructional activities are low in verbal complexity and do not create a lot of sensory overload compared to other students. In order to address these unique needs of students with ASD, studies included in this review use a variety of instructional approaches and program environments that are based on the principles of inclusive and neurodiverse aware pedagogy.

In this section, are summarized some of the programming activity types found in the literature reviewed so we can understand the nature and organization of the programming activity being described in the literature. The programming activity can be assessed by the educational objectives for which programming activities are conducted, collaboration type of programming activity implementation, types of programming technology used in a given program activity, and the psychoeducational / pedagogical strategies used to promote participatory involvement. We will also explore how programming environments / activities were adapted or customized for each unique autistic learner according to their cognitive and emotional profiles through visual supports, gamified task structure increments, and scaffolding techniques. In this part of the literature review, the authors identify consistent themes and emerging "best practices" that we can use to promote programming curricula for individuals with Autism Spectrum Disorder and document where empirical evidence is lacking to support inclusive programming practices. Examples presented in this section will further document research and innovative methods to provide meaningful learning opportunities to all individuals with Autism Spectrum Disorder and to promote equitable opportunities for all individuals on the Autism Spectrum to participate and be involved in their learning activities.

The Goals of Programming Activities

Within this study of multiple reviews, the programming activities of children with Asperger's syndrome were not solely intended to improve their skills in programming, but rather to achieve a number of other educational objectives beyond that of developing a child's technical ability. These objectives can be classified in four areas that have varying degrees of overlap with one another: (1) Computational thinking; (2) Academic skill growth; (3) Social skills communications and collaborations; and (4) Engagement and determination of the learner.

Most responses reflected interest in acquiring an understanding of how to use programming languages to create software. This was reflected in the majority of studies particularly as they involved a structured programming task targeting the development of computational thinking applied via block-based or visual environments such as Scratch, Blockly-based systems or custom made more serious types of videogames. These environments were intentionally selected to reduce syntactic complexity and cognitive load, allowing learners to focus on algorithmic thinking rather than textual syntax. Scaffolding mechanisms were generally included within instructional designs so that concept acquisition happened progressively through the use of a succession of increasingly complex tasks, visual cues, and restricted options to help learners with their gradual acquisition of a given concept (Elshahawy et al., 2020a; Elshahawy et al., 2020b; Elshahawy et al., 2022; Özgüç & Altın, 2022).

Many studies have focused on developing academic skills related to computation thinking. Specific skills included developing foundational digital literacy, logical reasoning, numeracy, and for some learners, these skills were directed towards an older group due to the presence of an impairment. Some studies embedded programming into curricula to enhance executive functions associated with independent thinking, attention, and task completion by way of goal-based / problem-solving activities through predictable activity structures (Özgüç & Altın, 2022; Elliott, 2018).

The final goal set by different participants was to help facilitate social communication and collaboration, which can often be difficult for students with ASD. Five studies contained explicit planning considerations for collaborative programming arrangements such as pair programming, working in small groups, and opportunities for facilitated interactions during programming tasks. In these cases, programming was used not only as an educational goal, but also as a catalyst for developing turn-taking, joint attention, joint problem solving, and expressive means of communication. Goals related to social interaction were frequently noted in camp-based, remote, and integrated programming and social skilling interventions, where socialisation and programming skills have been purposefully scaffolded on top of each other (Begel et al., 2021; Anderson & Donehower Paul, 2024).

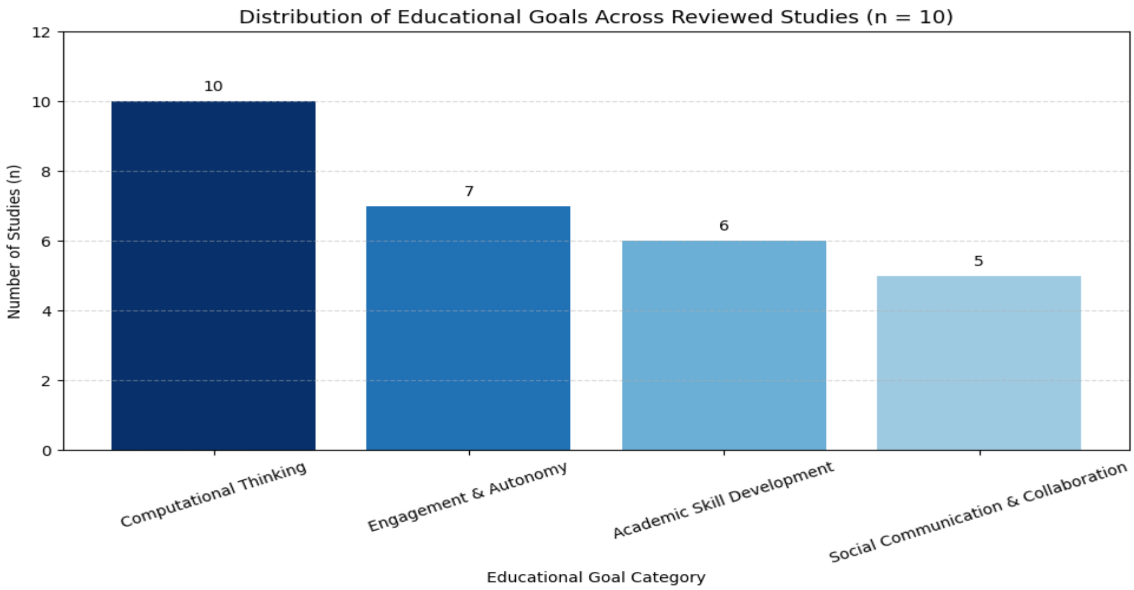

Figure 5. ***Distribution of Primary Educational Goals Across Reviewed Intervention Studies (n = 10)***

The distribution of primary learning goals across the studies is shown in Figure 5. The last few years a considerable number of studies have focused on learner engagement and learner autonomy as the primary objectives of programming activities. Specifically, it has often been reported that the interactive, visual, and creative capabilities of block-based tools enhance learners' motivation, retain their focus on the task at hand, and provide them with a sense of ownership over the learning process. As a result, there is an emphasis on customization opportunities, immediate response, and playful exploration as critical tools for supporting participation among learners with sensory sensitivities or limited verbal skills (Elshahawy et al., 2020a; Kaliouby et al., 2006). It should be noted that the three goals (Cognitive, Social and Affective) could cross over into each other. Some interventions employed a multi-dimensional design which integrated their cognitive, social, and affective outcomes at one time, indicating a growing awareness of the need for effective software development education for autistic / ASD students to balance technical skills with engagement, inclusion and personal development.

The Collaboration Context in Relation to the Goals of Programming Activities

The learning environment in which programming learning activities occur is an important factor influencing learning outcomes and experiences for learners with Autism Spectrum Disorder (ASD). For the ten core intervention studies, programming learning activities were performed in three formats (individual, dyadic, and group) that were empirically aligned to both intended pedagogical outcomes and learner characteristics. Of the ten core studies, the individual learning setting format was utilized the most frequently (8 studies), typically when the focus was on building foundational skills in programming and / or computational thinking such as on sequencing, debugging, and recognizing control structures in block-based coding (Elshahawy et al., 2020a; Elshahawy et al., 2020b; Özgüç & Altın, 2022). Individual learning settings provide personalized pacing, fewer social and sensory demands, and greater instructional control than dyadic and group-based formats and therefore may be particularly helpful for learners with increased sensory sensitivities and / or limited verbal communication (Rogers & Bibby, 2017; Zubair et al., 2019).

In four studies where paired programming and peer-supported activities were used, dyadic collaboration formats were used to support social communication and / or joint problem-solving as learning objectives. For example, in these studies, paired programming and peer-mediated activities were used to facilitate turn-taking, sharing / attention, and verbal communication through the use of a programming task (i.e., paired programming, peer-mediated activities; Begel et al., 2021; Anderson & Donehower Paul, 2024). These activities included both structured social engagements and predictable programming tasks, both of which are essential for autistic learners. Likewise, group or classroom-based collaborations were reported in four studies, usually within inclusive or semi-inclusive education settings. Activities contained within larger classroom routines or camps that included both autistic learners and neurotypical peers contained programming activities (e.g., Rogers & Bibby, 2017; Papavlasopoulou et al., 2019). Group-based programming collaborations did have barriers to these collaborations such as the need for group management and differentiated instruction, as well as the need for additional instructional support (e.g., Elliott, 2018; Anderson & Donehower Paul, 2024).

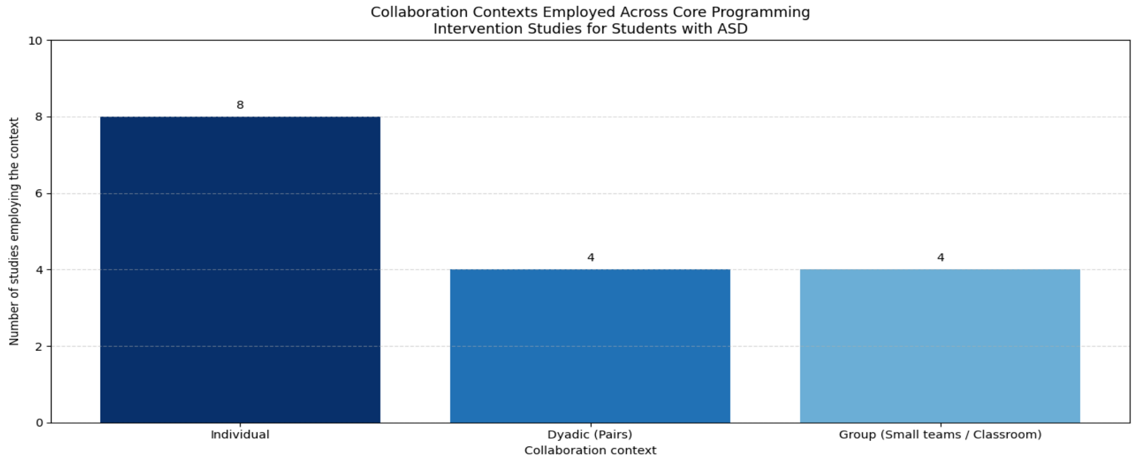

Figure 6. ***Collaboration Settings in Programming Activities for Students with ASD***

The distribution of the collaboration contexts from this review of intervention studies is shown in Figure 6. Several studies used multiple collaboration formats to change the amount of social interaction between learners and teachers based on task difficulty, learners' readiness for learning, or instructional phase (Papavlasopoulou et al., 2019; Begel et al., 2021). This indicates an evolving view of collaboration as a pedagogical strategy rather than something that is simply required, and that the ability to participate effectively in collaboration is linked to coordinating the social demands of collaborative programming with the cognitive and sensory profiles of learners. Studies in this review indicate that collaboration in autism-inclusive programming education is most successful when it is deliberately structured / scaffolded, with defined roles, predictable patterns of interaction, and supported through explicit instructions. When these criteria are met, collaborative programming environments can help learners achieve both technical and broader social participation goals simultaneously.

Programming Technologies That Have Been Used in Programming Activities

The studies that were reviewed were focused on using various programming technologies, mainly visual and block-based programming paradigms, to reduce syntax complexity and increase accessibility for students with Autism Spectrum Disorder (ASD). The studies used for both intervention and context were consistent in their selection of technology with the reported sensory, cognitive, and communication characteristics of students with autism, favoring technology with easy-to-use interfaces or drag and drop and predictable behavior of the system.

The majority of studies reported that Scratch is the most utilized programming environment, with 8 out of 16 studies using Scratch (e.g., Rogers & Bibby, 2017; Papavlasopoulou et al., 2020; Begel et al., 2021; Elliott, 2018). Scratch has a visual interface and is user-friendly to allow for flexibility in developing both games, animations and story-based content, making it a favorite choice for inclusion in programming. The interventions utilizing Scratch were primarily aimed at developing computational thinking skills, providing opportunities for creative expression and supporting high levels of learner engagement through exploratory and project-based learning.

Blockly-based environments were the second-most used type of programming technologies (5 total studies) and were most commonly integrated with independent learning platforms or combined with robotics and gamified systems (Elshahawy et al., 2022; Özgüç & Altın, 2022). The modularity of Blockly enabled teachers and researchers to change task complexity and accommodate learner differences for each student, making Blockly a good fit for structured approaches to learning and design of comparison interventions.

In different research projects, many used elementary forms and companion applications (or devices) such as ScratchJr, MakeCode, Snap!, and MakeyMakey between 1 and 2 times each. Specifically, ScratchJr was designed to support young children and / or those who have limited spoken language. Therefore, it makes sense that this platform was used predominantly with younger / less verbal children because it uses icon-based design principles and has less written language (Özgüç & Altın, 2022). MakeCode and Snap! allowed the introduction of programming while they also helped develop logical reasoning skills Mapey and MakeyMakey offered hands-on or creative computer programming through their ability to connect a physical action to the result of a visual program (Rogers & Bibby, 2017).

In six of the studies reviewed here, educational robotics and physical computing were incorporated into many coding / programming experiences either as intended programming environments or as supporting tools. This meant that participants engaged with micro:bit activities, LEGO WeDo kits, and simple programmable robots (typically controlled via a Scratch or Blockly interface) (Rogers & Bibby, 2017; Elliott, 2018; Anderson & Donehower Paul, 2024). These technologies were often used to promote student motivation, engagement, and embodied learning rather than to evaluate programming proficiency and computational learning outcomes independently.

The reviewed articles consistently showed that technology selection includes either High / Low floor design or High / Low ceiling design in the design of your technology. An example would be allowing the entry of a beginning programmer into a programming environment and progressively advancing him or her to develop more complex programs. (Bers, 2020). Usability, adaptability, and flexibility were repeatedly emphasized as critical design criteria, particularly for students with high sensory sensitivity or executive functioning challenges (Zubair et al., 2019).

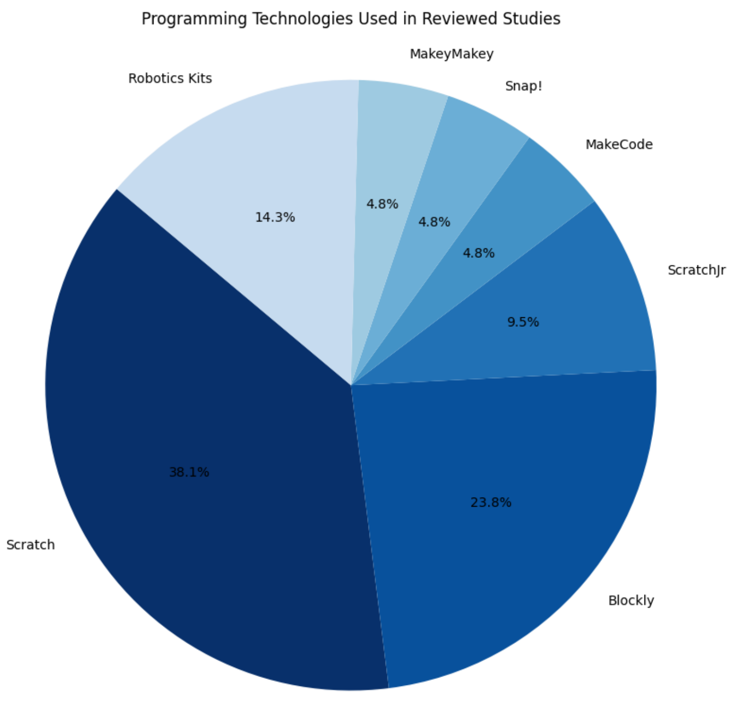

Figure 7. ***Distribution of Programming Tools and Platforms Across Reviewed Educational Studies***

Figure 7 shows the distribution of the programming technologies assessed and the technology categories (software, hardware, and hybrid) in the reviewed studies. The numerous studies of the same well-known programming platforms (Scratch and Blockly in particular) show a limited number of studies examining alternative programming technologies and new and different types of programming tools. For example, very few studies assessed programming technologies that include adaptive systems, AI-supported tools, and new types of human-computer interaction for programming (e.g., voice interfaces, augmented reality). This limited focus on technology suggests the need for more extensive research to explore the wider universe of programming tools and to assess how emerging programming technologies can be developed in conjunction with the use of autistic students and can accommodate the diverse needs of students with autism through collaborative design and longitudinal research.

The Recommended Psychoeducational Methodologies and Strategies That Support the Participation of Students with Autism in Programming Activities

Psychoeducational and instructional strategies are frequently seen as enabling participation by reducing both cognitive barriers and sensory barriers as a means to increase engagement and progression in the programming for each task or activity. Also, the number of studies showing scaffolding and structuring of tasks are most frequently cited (*n* = 9) in how they can be used to structure the experience. The use of scaffolding and structuring of tasks is the method of systematically approaching activities through smaller, predictable steps, routine use of some type of prompts, guide or sequence. These systematic processes have been commonly used in previous studies of serious games / gamifying interventions related to autistic individuals (Elshahawy, 2020a; Elshahawy, 2020b; Elshahawy, 2022) and in the single case studies of systematic / stepwise teaching and skill building (Özgüç & Altın, 2022).

The overall results of the research conducted on structured teaching approaches suggest that teachers use a variety of methods to support the needs of students on the AS, including chunking information and establishing routines. The majority of practices are focused on visually representing the information being taught / learned and using multiple modalities of learning to do so (*n* = 7). The specific types of visual supports included visual instructions, visual representations that create less load for students (e.g., using block programming in place of traditional programming), and visual feedback mechanisms that are both on-screen and through physical computing (e.g., physically manipulating an object to see an outcome on the computer). These practices have been supported through research that has been done primarily with design studies, highlighting the importance of having a clear interface; having predictable interfaces; and having a low cognitive load for the student has been shown to have a positive impact on the ability of autistic children to pay attention and learn through the use of computers. Additionally, research done using physical computing supports the idea that being able to interact with something physically will increase engagement and understanding by the student (Rogers & Bibby, 2017; Elliott, 2018).

Many studies (*n* = 6) included individualized pacing and differentiation, highlighting that autistic students require flexible progression, more time to process information, and changes based on their unique communication / sensory profiles. This theme was evident among both structured interventions and classroom-based studies, as pacing and personalisation were emphasized in both cases as core facilitators of ongoing participation (Özgüç & Altın, 2022; Elliott, 2018; Shah et al., 2024). There were also several (*n* = 6) studies that identified motivational supports (i.e., gamification, rewards, and interest-based tasks). In the autism-specific intervention literature, motivation was not treated as an “add-on,” but as an integral design feature that helps maintain persistence and reduces attrition (Elshahawy et al., 2020a; Elshahawy et al., 2020b; Elshahawy et al., 2022). Motivation-oriented design was also evident in settings where programming activities were embedded in camp or socially meaningful formats, supporting sustained participation and confidence (Begel et al., 2021; Anderson & Donehower Paul, 2024).

Less frequently, studies reported co-design and participatory development (n = 5) and peer-mediated or inclusive learning structures (n = 4). Participatory approaches were mainly discussed as a way to improve contextual fit and accessibility by aligning tools and activities with learner needs and stakeholder realities (Zubair et al., 2019; Elliott, 2018; Shah et al., 2024). Peer-mediated and inclusive configurations were most evident in interventions that explicitly integrated communication or social participation goals alongside programming (Anderson & Donehower Paul, 2024; Begel et al., 2021), and in broader discussions of participation-oriented technology use in autism-related robotics contexts (Nanou & Karampatzakis, 2023).

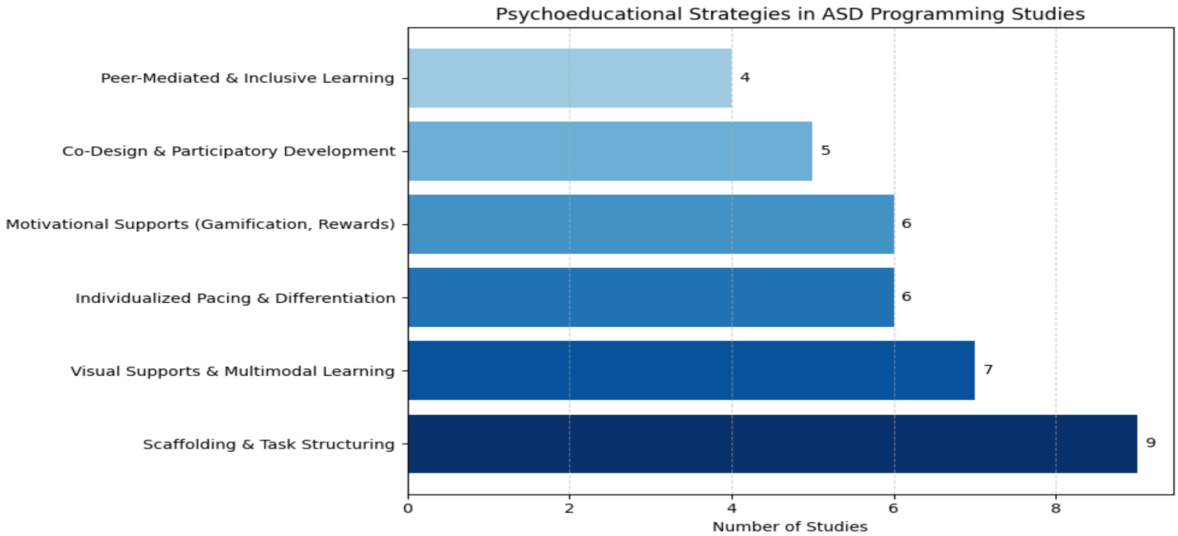

Figure 8. ***Distribution of psychoeducational strategies employed to support the participation of students with ASD in programming-based learning interventions***

Figure 8 presents the frequency with which the above psychoeducational strategies were explicitly reported across the included studies. Overall, the reviewed studies converge on the idea that effective participation is produced through combinations of strategies, rather than a single universal method. The literature consistently suggests that technology choice (e.g., block-based tools, serious games, physical computing) becomes meaningfully inclusive only when paired with structured scaffolding, visual / multimodal supports, differentiated pacing, and motivation-sensitive design (Elshahawy et al., 2020a; Elliott, 2018; Zubair et al., 2019; Shah et al., 2024). At the same time, variability in reporting and implementation across studies indicates a need for more systematic evaluation of "which strategy works best, for whom, and under what conditions," particularly across diverse educational contexts.

In addition to general scaffolding and differentiation practices, one study proposed a highly structured, role-based collaboration strategy explicitly designed to support the participation of children with ASD in programming-related activities: the Search and Share Strategy (SaSS). The SaSS is a structured psychoeducational strategy developed within inclusive educational robotics contexts to reduce participation barriers and foster active involvement of autistic learners during collaborative construction and programming tasks (Nanou et al., 2021; Tsiomi et al., 2023; Tsiomi et al., 2025). Grounded in principles of structured teaching and collaboration scripting, the strategy decomposes participation into a sequence of clearly defined, visually represented steps (Come, Find, Look, Look For, Give), enabling students with ASD to engage meaningfully in teamwork while assuming a specific functional role.

Empirical evidence from intervention-based research indicates that the SaSS contributed to increased engagement, reduced social withdrawal, and improved autonomy of children with ASD during both construction and programming phases of LEGO Mindstorms activities. The strategy was particularly effective in supporting children with moderate support needs (ASD

Level 2), for whom unstructured collaboration often poses significant challenges. Importantly, the SaSS aligns with several psychoeducational approaches identified across the reviewed literature, including scaffolding, visual supports, role assignment, and peer-mediated collaboration. However, unlike more general recommendations, the SaSS provides a concrete, operationalized framework that can be replicated, adapted, and evaluated in inclusive programming contexts. Although applied in a limited number of studies, the SaSS illustrates the potential of structured collaboration strategies to bridge the gap between social participation and programming engagement for autistic learners. Its inclusion highlights the need for further research on explicit collaboration scripts and role-based participation models within block-based and visual programming interventions (Nanou et al., 2021; Tsiomi et al., 2023; Tsiomi et al., 2025).

Discussion

This scoping review examined how programming education particularly, block-based, visual, and game-based approaches, has been designed and implemented for children and adolescents with Autism Spectrum Disorder (ASD). By synthesizing evidence from sixteen peer-reviewed studies published between 2017 and 2025, the review offers an integrated perspective on the pedagogical roles of programming technologies, the conditions that shape participation, and the broader educational ecosystem required to support inclusive computing practices.

A central insight emerging from the reviewed literature is that programming technologies function primarily as enabling media rather than as independent determinants of learning outcomes. This finding is also reflected across multiple intervention studies that emphasize the supportive role of programming environments when combined with appropriate instructional design (Elshahawy et al., 2020a; Özgüç & Altın, 2022). Block-based and visual environments, most notably Scratch and Blockly, are consistently selected because they reduce syntactic burden, provide strong visual structure, and support predictable interaction patterns that align with autism-informed pedagogical principles (Papavlasopoulou et al., 2019; Elliott, 2018; Elshahawy et al., 2020a). Across intervention-focused studies, these environments facilitated access to foundational computational thinking concepts such as sequencing, loops, and conditional logic, while simultaneously supporting engagement and task persistence. Similar outcomes have been reported in studies using both game-based and block-based environments, where learners demonstrated improved engagement and understanding of programming concepts (Elshahawy et al., 2022; Begel et al., 2021).

However, the review indicates that the effectiveness of these tools is highly contingent on how they are embedded within instructional design. Studies that reported positive outcomes consistently observed technology use with structured scaffolding, individualized pacing, and visual or multimodal supports (Özgüç & Altın, 2022; Elshahawy et al., 2022). These instructional patterns are consistently observed across intervention-based studies, particularly in single-case and small-scale implementations that prioritize individualized learning approaches (Özgüç & Altın, 2022; Elshahawy et al., 2020b). This finding reinforces the argument that accessibility in programming education is not an intrinsic property of platforms alone, but an emergent property of pedagogical alignment between tools, learners, and instructional practices. However, the findings across studies are not entirely consistent. While several interventions report improvements in engagement and participation, evidence regarding the development of sustained programming skills remains limited. In addition, variations in study design, sample size, and intervention duration make it difficult to draw definitive conclusions regarding the comparative effectiveness of different programming approaches.

Across the corpus, pedagogical strategies played a more decisive role than platform choice in shaping students' participation and learning experiences. Scaffolding and task structuring were the most consistently emphasized approaches, enabling learners with ASD to engage with complex programming tasks through incremental progression and predictable routines (Papavlasopoulou et al., 2019; Elshahawy et al., 2020b). Visual supports and multimodal interaction further capitalized on the visual processing strengths often reported in autistic learners, while also mitigating cognitive and sensory overload (Rogers & Bibby, 2017).

Individualized pacing and differentiation emerged as particularly important in single-case and small-scale interventions, where learner variability was explicitly addressed through flexible task design and adaptive progression (Özgüç & Altın, 2022). In contrast, peer-mediated and collaborative learning approaches were applied more selectively, typically in contexts where social communication goals were integrated alongside programming objectives (Begel et al., 2021; Anderson & Donehower Paul, 2024). This selective use of collaboration has also been noted in studies integrating social skill development with programming activities, where structured support is required to ensure meaningful participation (Anderson & Donehower Paul, 2024; Begel et al., 2021). These findings suggest that collaboration in programming activities should be viewed as a context-sensitive pedagogical choice, rather than a universally applicable strategy.

Table 3. ***Pedagogical strategies, evidence, and remaining gaps in autism-inclusive programming studies***

Pedagogical strategy	Evidence from reviewed studies	Remaining gaps
Scaffolding & task structuring	Strong, consistent use across interventions (Papavlasopoulou et al., 2019; Elshahawy et al., 2022)	Lack of standardized, transferable models
Visual & multimodal supports	Frequently reported, especially with physical computing (Rogers & Bibby, 2017)	Limited evaluation of sensory-specific impact
Individualized pacing & differentiation	Common in single-case and small-scale studies (Özgüç & Altın, 2022)	Rarely scalable or systematically evaluated
Peer-mediated & inclusive collaboration	Applied selectively (Begel et al., 2021; Anderson & Donehower Paul, 2024)	Risk of uneven participation without structured facilitation
Participatory co-design	Present in design-oriented studies (Zubair et al., 2019)	Limited evidence of learning impact
Structured Collaboration Scripts	Emerges as a unifying pattern across interventions that combine task structuring, visual supports, adaptive pacing, and educator mediation (Nanou et al., 2021; Tsiomi et al., 2023; Tsiomi et al., 2025)	The literature primarily frames Search and Share Strategy in terms of individual practices, rather than as a unified and systematically replicable pedagogical strategy

Despite frequent references to inclusion and accessibility, explicit empirical examination of sensory, cognitive, and motor accessibility remains limited across the reviewed studies. This limitation is also reflected in design-oriented studies, which tend to propose accessibility principles without systematically evaluating their impact in applied learning contexts (Zubair et al., 2019).

Design-oriented research provides valuable principles regarding predictability, reduced complexity, and interface clarity (Zubair et al., 2019), yet these considerations are rarely operationalized or evaluated within intervention studies. This gap is particularly concerning given that even visually accessible tools may become exclusionary without appropriate customization for learners with hypersensitivities or executive functioning challenges.

Similarly, educator training and preparedness emerged as underexplored but critical factors. While some studies acknowledged the role of teacher mediation or incorporated reflective practices (Rogers & Bibby, 2017; Shah et al., 2024), few provided evidence of structured, scalable professional development models. This finding underscores a systemic vulnerability: without adequately trained educators, the potential of inclusive programming technologies is unlikely to be realized consistently across contexts (Elliott, 2018).

Taken together, the reviewed studies conceptualize participation in programming education for students with ASD as a multi-layered phenomenon. Direct engagement in programming tasks constitutes only one dimension; equally important are pedagogical enablers (e.g., scaffolding, pacing, teacher support) and contextual conditions (e.g., accessibility, institutional readiness). Review-level evidence from educational robotics and inclusive STEM literature further indicates that participation is often framed in terms of motivation and social engagement, with comparatively less emphasis on sustained programming skill development (Nanou & Karampatzakis, 2023; Ariza et al., 2025). This trend is particularly evident in robotics-based and participation-focused studies, where engagement is prioritized over the development of advanced programming competencies (Nanou & Karampatzakis, 2023). This layered understanding reinforces the need to move beyond tool-centric perspectives toward ecosystem-oriented frameworks that account for instructional design, educator capacity, and learner diversity.

The findings of this review carry several implications. For practitioners, they highlight the importance of aligning programming technologies with autism-informed pedagogical strategies, rather than relying on platform accessibility alone. For curriculum designers and developers, the results emphasize the need for flexible, customizable environments that support diverse sensory and cognitive profiles. For policymakers, the lack of evidence on educator training and accessibility design signals an urgent need for investment in inclusive capacity-building.

Table 4. ***Priority directions for future research***

Research priority	Rationale
Longitudinal intervention studies	Current evidence is predominantly short-terms
Standardized outcome measures	Improve comparability across studies
Accessibility-focused design research	Address sensory and cognitive barriers explicitly
Educator training frameworks	Support sustainable, scalable implementation
LMIC-focused studies	Address geographic and socio-economic imbalance

This review is subject to limitations typical of scoping methodologies, including reliance on English-language, peer-reviewed sources and substantial heterogeneity in study design and outcome measures. Nevertheless, by integrating intervention-focused and contextual evidence, the review provides a coherent synthesis of the current state of autism-inclusive programming education. The findings suggest that inclusive programming education for students with ASD requires a shift toward holistic, systems-oriented approaches that integrate technology, pedagogy, educator support, and accessibility design. Such a perspective is essential if programming education is to become a genuinely inclusive and empowering pathway for autistic learners across diverse educational contexts.

Conclusions

This scoping review synthesized research published between 2017 and 2025 that examines programming education for students with Autism Spectrum Disorder (ASD), with particular emphasis on block-based, visual, and game-based programming approaches. By systematically mapping empirical intervention studies alongside contextual and pedagogical contributions, the review provides a structured overview of how programming activities have been designed, implemented, and supported for autistic learners across diverse educational contexts. The findings indicate that block-based and visual programming environments, most notably Scratch, Blockly-based platforms, and related gamified or serious-game systems, dominate the current landscape of programming interventions for students with ASD. These environments are consistently selected for their reduced syntactic complexity, visual clarity, and capacity to support step-by-step reasoning, making them well aligned with the cognitive and sensory profiles commonly reported among autistic learners. Across intervention studies, such tools were primarily used to introduce foundational programming and computational thinking concepts, including sequencing, conditionals, and iterative logic.

However, the review also demonstrates that technology alone is insufficient to ensure meaningful participation or learning outcomes. The effectiveness and feasibility of programming interventions were closely tied to the psychoeducational strategies employed alongside technological tools. Scaffolding, task structuring, visual and multimodal supports, individualized pacing, and motivational elements emerged as recurrent and influential design features across the literature. These strategies functioned not only to reduce cognitive load but also to enhance engagement, persistence, and learner confidence, reinforcing the importance of pedagogical intentionality in inclusive computing education. Despite these promising trends, the review identified several persistent gaps in the literature. Accessibility considerations particularly those related to sensory sensitivities, motor demands, and interface customization remain underexplored, even within studies explicitly targeting autistic learners. Similarly, educator training and preparedness received limited empirical attention, despite teachers' central role in mediating programming activities and adapting them to individual learner needs. The dominance of small-scale, short-term interventions further constrains conclusions regarding long-term learning trajectories, skill transfer, and sustainability. Another notable limitation concerns the uneven geographical distribution of research. The majority of studies were conducted in high-income countries, with minimal representation from low- and middle-income contexts. This imbalance raises concerns regarding the global applicability of current findings and highlights the need for culturally responsive and resource-sensitive programming models that extend beyond well-resourced educational settings. Taken together, the evidence suggests that programming education for students with ASD is an evolving but still fragmented research field. While substantial progress has been made in identifying accessible tools and supportive pedagogical strategies, future work must move toward more integrated, system-level approaches. Such approaches should combine accessible technology design, evidence-based instructional frameworks, structured educator training, and rigorous evaluation methodologies.

In conclusion, programming education holds considerable potential as a means of fostering not only computational skills but also agency, confidence, and inclusion for autistic learners. Realizing this potential requires sustained research efforts that extend beyond isolated interventions and address the broader educational ecosystems in which programming activities are embedded. By doing so, future research can contribute to the development of inclusive computing education practices that are equitable, scalable, and responsive to the diverse needs of students with ASD.

References

Anderson, E., & Donehower Paul, C. (2024). Integrated Computing (Coding + Social Skills) For Elementary Students with Autism Spectrum Disorder. *Proceedings of the 2024 on ACM Virtual Global Computing Education Conference V. 1, SIGCSE Virtual 2024,* 12–18. https://doi.org/10.1145/3649165.3690120

Ariza, J. Á., & Hernández Hernández, C. (2025). A Systematic Literature Review of Research-based Interventions and Strategies for Students with Disabilities in STEM and STEAM Education. *International Journal of Science and Mathematics Education, 23*(7), 2863–2893. https://doi.org/10.1007/s10763-025-10544-z

Arksey, H., & O'Malley, L. (2005). Scoping studies: Towards a methodological framework. *International Journal of Social Research Methodology, 8*(1), 19–32. https://doi.org/10.1080/1364557032000119616

Baron-Cohen, S. (2006). The hyper-systemizing, assortative mating theory of autism. *Progress in Neuro-Psychopharmacology and Biological Psychiatry, Includes a Special Topic on Evolutionary Psychiatry, 30*(5), 865–872. https://doi.org/10.1016/j.pnpbp.2006.01.010

Begel, A., Dominic, J., Phillis, C., Beeson, T., & Rodeghero, P. (2021). How a Remote Video Game Coding Camp Improved Autistic College Students' Self-Efficacy in Communication. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education, SIGCSE '21*, 142–148. https://doi.org/10.1145/3408877.3432516

Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education, 58*(3), 978–988. https://doi.org/10.1016/j.compedu.2011.10.006

Bers, M. U. (2020). *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom* (2nd ed.). Routledge. https://doi.org/10.4324/9781003022602

Ehsan, H., Rispoli, M., Lory, C., & Gregori, E. (2018). A Systematic Review of STEM Instruction with Students with Autism Spectrum Disorders. *Review Journal of Autism and Developmental Disorders, 5*(4), 327–348. https://doi.org/10.1007/s40489-018-0142-8

Elliott, C. (2018). *Computing in Special Educational Needs & Disabilities Settings: The Current Picture in England 2018*.

Elshahawy, M., Aboelnaga, K., & Sharaf, N. (2020). CodaRoutine: A Serious Game for Introducing Sequential Programming Concepts to Children with Autism. *2020 IEEE Global Engineering Education Conference (EDUCON)*, 1862–1867. https://doi.org/10.1109/EDUCON45650.2020.9125196

Elshahawy, M., Bakhaty, M., Ahmed, G., Aboelnaga, K., & Sharaf, N. (2022). Towards Developing Computational Thinking Skills Through Gamified Learning Platforms for Students with Autism. In M. E. Auer, A. Pester, & D. May (Eds.), *Learning with Technologies and Technologies in Learning: Experience, Trends and Challenges in Higher Education* (pp. 193–216). Springer International Publishing. https://doi.org/10.1007/978-3-031-04286-7_10

Elshahawy, M., Bakhaty, M., & Sharaf, N. (2020). Developing Computational Thinking for Children with Autism using a Serious Game. *2020 24th International Conference Information Visualisation (IV)*, 761–766. https://doi.org/10.1109/IV51561.2020.00135

Hourcade, J. P., Bullock-Rest, N. E., & Hansen, T. E. (2012). Multitouch tablet applications and activities to enhance the social skills of children with autism spectrum disorders. *Personal and Ubiquitous Computing, 16*(2), 157–168. https://doi.org/10.1007/s00779-011-0383-3

Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, 82, 263–279. https://doi.org/10.1016/j.compedu.2014.11.022

Kalashnikova, L. V., Lobanova, A. S., Hrabovets, I. V., Chernous, L. S., Chorna, V. A., & Davydenko, Y. O. (2025). Modern information and communication technologies in professional training of sociology students: Evolving needs and significance in the digital age. *CTE Workshop Proceedings*, 12, 381–398. https://doi.org/10.55056/cte.922

Kaliouby, R. el, Picard, R., & Baron-Cohen, S. (2006). Affective Computing and Autism. *Annals of the New York Academy of Sciences*, *1093*(1), 228–248. https://doi.org/10.1196/annals.1382.016

Kishida, Y., & Kemp, C. (2009). The Engagement and Interaction of Children With Autism Spectrum Disorder in Segregated and Inclusive Early Childhood Center-Based Settings. *Topics in Early Childhood Special Education*, *29*(2), 105–118. https://doi.org/10.1177/0271121408329172

Levac, D., Colquhoun, H., & O'Brien, K. K. (2010). Scoping studies: Advancing the methodology. *Implementation Science*, *5*(1), 69. https://doi.org/10.1186/1748-5908-5-69

Madzhi, N. K., Thamrin, N. M., Zulkifli, Z., & Herman, S. H. (2024). Exploring the Potentials of Robotic Inclusive Education in Supporting Students with Disabilities. *Journal of Creative Practices in Language Learning and Teaching (CPLT)*, *12*(2), 125–135. https://doi.org/10.24191/cplt.v12i2.3626

Menziltsidou, S., Tsiomi, E., Vasiloudi, E., Nanou, A., & Karampatzakis, D. (2025). Designing Inclusive Coding Display Cards for Micro:bit. *Proceedings of the 28th Pan-Hellenic Conference on Progress in Computing and Informatics, PCI '24*, 301–306. https://doi.org/10.1145/3716554.3716600

Nanou, A., & Karampatzakis, D. (2022). Collaborative educational robotics for the inclusion of children with disabilities. *Education. Innovation. Diversity*, *1*(4), 30–43. https://doi.org/10.17770/eid2022.1.6899

Nanou, A., & Karampatzakis, D. (2023). The Participation of Students with Autism in Educational Robotics: A Scoping Review. *Social Sciences*, *12*(12), 675. https://doi.org/10.3390/socsci12120675

Nanou, A., Tsiomi, E., Oikonomou, A., & Karampatzakis, D. (2021). The SAS strategy training for children with ASD in inclusive educational robotics activities. *Education. Innovation. Diversity*, *2*(3), 34–52. https://doi.org/10.17770/eid2021.2.6723

Özgüç, C. S., & Altın, D. (2022). Teaching Block-Based Coding to a Student with Autism Spectrum Disorder. *Ankara Üniversitesi Eğitim Bilimleri Fakültesi Özel Eğitim Dergisi*, *23*(3), 565–594. https://doi.org/10.21565/ozelegitimdergisi.822554

Papavlasopoulou, S., Giannakos, M. N., & Jaccheri, L. (2019). Exploring children's learning experience in constructionism-based coding activities through design-based research. *Computers in Human Behavior*, *99*, 415–427. https://doi.org/10.1016/j.chb.2019.01.008

Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2020). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*, *105*, 105939. https://doi.org/10.1016/j.chb.2019.03.003

Pereira, J., Castro, P. M., Dapena, A., & Penedo, M. G. (2023). Inclusive projects in science and technology to secondary and higher education. *International Conference on Education and New Developments*. https://doi.org/10.36315/2023v2end024

Rogers, Y., Shum, V., Marquardt, N., Lechelt, S., Johnson, R., Baker, H., & Davies, M. (2017). From the BBC micro to micro:bit and beyond: A British innovation. *Interactions*, *24*(2), 74–77. https://doi.org/10.1145/3029601

Shah, S. M., Elliott, C., & Nedungadi, P. (2024). Square Pegs and Round Holes: Pedagogy for Autistic Students in Computing Education. *IEEE Transactions on Education*, *67*(6), 919–930. https://doi.org/10.1109/TE.2023.3335395

Tricco, A. C., Lillie, E., Zarin, W., O'Brien, K. K., Colquhoun, H., Levac, D., Moher, D., Peters, M. D. J., Horsley, T., Weeks, L., Hempel, S., Akl, E. A., Chang, C., McGowan, J., Stewart, L., Hartling, L., Aldcroft, A., Wilson, M. G., Garritty, C., … Straus, S. E. (2018). PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and Explanation. *Annals of Internal Medicine*, *169*(7), 467–473. https://doi.org/10.7326/M18-0850

Tsiomi, E., Maistrelli, A., Nanou, A., & Karampatzakis, D. (2025). Enhancing Participation of Students with Autism in Inclusive Educational Robotics: An Action Research. In I. Kazanidis & A. Tsinakos (Eds.), *Research on E-Learning and ICT in Education: Technological, Pedagogical and Instructional Perspectives* (pp. 217–234). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-99961-1_13

Tsiomi, E., Nanou, A., & Karampatzakis, D. (2023). Remote training to support children with autism in inclusive educational robotics activities. *Conferences of the Greek Scientific Association for Information and Communication Technologies in Education*, 458–465.

Zolyomi, A., Begel, A., Waldern, J. F., Tang, J., Barnett, M., Cutrell, E., McDuff, D., Andrist, S., & Morris, M. R. (2019). Managing Stress: The Needs of Autistic Adults in Video Calling. *Proc. ACM Hum.-Comput. Interact.*, *3*(CSCW), 134:1-134:29. https://doi.org/10.1145/3359236

Zubair, M. S., Brown, D. J., Bates, M., & Hughes-Roberts, T. (2019). *Designing Accessible Visual Programming Tools for Children with Autism Spectrum Condition* (arXiv:1911.07624). arXiv. https://doi.org/10.48550/arXiv.1911.07624

Zubair, M. S., Brown, D. J., Hughes-Roberts, T., & Bates, M. (2021). Designing accessible visual programming tools for children with autism spectrum condition. *Univers. Access Inf. Soc.*, *22*(2), 277–296. https://doi.org/10.1007/s10209-021-00842-y