Integrating educational robotics in primary school classrooms: Exploring teacher and pupil perspectives.

Amy A. E. Hughes

A thesis submitted to the School of Psychology, Cardiff University, in partial fulfilment of the requirement for the degree of Doctor of Philosophy.

July 2024

Under the supervision of Dr Sarah Gerson and Dr Johanna van Schaik

Summary of Thesis

This thesis investigated integrating educational robotics in early primary school classrooms to enhance children's computational thinking (CT) skills and did so by exploring both teacher and child perspectives. In Chapter 1, I outlined recent changes in primary school computing curriculums and the significance of CT in early education. I also explored how CT skills can be targeted within programming education and reviewed methods for teaching CT and programming to children under the age of 8. Finally, I emphasised the importance of teachers' beliefs in shaping educational practices and outcomes.

In Chapter 2, I explored primary school teachers' beliefs about CT, programming, and robotics through a focus group, revealing insights into the digital landscape in Wales from practising teachers' perspectives. In Chapter 3, I extended this work by using a mixed-methods survey to gather broader insights from a more diverse sample of teachers. Findings showed that while teachers valued CT, programming, and robotics, they often lacked confidence in teaching these subjects but believed they could learn. The chapter also highlighted barriers such as lack of resources, training, and support. Recommendations for improving teacher education programs included incorporating developmentally appropriate content, providing hands-on robotics experiences, and discussing cross-curricular integration ideas. These recommendations informed the design of a teacher education workshop discussed in Chapters 5 and 6.

In Chapter 4, I examined how children (aged 4 to 7) learned with an educational robot, investigating the relationship between visual perspective taking (VPT) skills, programming performance, and executive functioning. I also explored the potential benefits of embodied learning for programming performance. I found no significant correlations between children's programming performance and their VPT or executive functioning skills. Furthermore, findings suggested that embodied learning methods did not significantly enhance performance on algorithm writing tasks or transfer of learning to other programming-related tasks.

In Chapters 5 and 6, I evaluated a school-based robotics intervention and its effects on children's CT abilities and teachers' beliefs. This study was teacher-led, used a control group and quantitative methods to assess the impact of a 6-week robotics curriculum and teacher education workshop. Results showed significant improvements in children's debugging and algorithm prediction skills and enhanced teachers' relevance, enjoyment, and self-efficacy beliefs when the workshop was included. Additionally, teachers' post-intervention self-efficacy was linked to pupil improvements in debugging and prediction tasks, demonstrating the importance of teacher confidence in student outcomes.

In summary, this thesis demonstrated (1) how educational robotics can be integrated within early primary school classrooms to benefit children's CT skills and teachers' beliefs and (2) the importance of considering the role of the teacher in education research. Implications for primary education and future research are discussed in Chapter 7.

Contents

List of Tables

List of Figures

Acknowledgements

Firstly, I would like to express my deepest thanks to my primary supervisor, Dr. Sarah Gerson. I am extremely grateful for your advice, enthusiasm, and belief in me. Your dedication and the countless hours you invested in data collection were instrumental in realising the full potential of this research. To Dr. Johanna van Schaik, despite being in a different country, your guidance and support have never wavered. Your advice and insights were crucial to the completion of this thesis, and I am profoundly thankful for your dedication and words of encouragement. To Dr. Carina Girvan, thank you for generously sharing your extensive knowledge of qualitative analysis. Your expertise has greatly enriched my research and skillset. To Dr. Dominic Guitard, thank you for providing your technical expertise and assisting with the analysis of my school intervention data.

Of course, this research would not have been possible without the enthusiastic participation of the schools, teachers, families, and children. Your cooperation and willingness to engage in these studies were vital to the success of each project. I am deeply appreciative of your time, effort, and the valuable insights you provided. I would also like to thank Primo Toys/ MORAVIA Consulting for donating the robotics equipment that made much of my research possible.

I could not have gotten through the last four years without the support of my CUCHDS office mates. Kai, Jen, Rhys, Kate, Ellie, and Becca thank you for creating a supportive and enjoyable work environment. A special thank you to Kelsey, I value your friendship and advice greatly and I am immensely grateful for your patience while I was learning to use R. To my Cardiff dance girls, your friendships helped me get through the low and challenging points of this journey, so thank you.

I am also grateful to the students who assisted with these projects over the years: Vicky, Lloyd, Alexandra, Grace, Hannah, Chara, Georgie, Zoe, Matt, Ellie, and Jamie. Your help was greatly appreciated and contributed significantly to the success of this research.

Finally, I would like to express my profound gratitude to my family. To Mum and Dad, for your unwavering support and encouragement; to my brother and sister, for always being there for me; and to Rufus, our family dog, for the joy and comfort you bring even on the toughest days. To Grandad and Pauline, for always cheering me on. And a special thank you to James, for your constant love and support throughout this journey. You believe in me when I don’t believe in myself, and I’m forever grateful for your presence in my life.

Chapter 1. General Introduction

In recent years, changes have been made to primary school curriculums across the world to integrate computational thinking skills. Past research has identified several approaches and methods that can be used to foster computational thinking (i.e., problem solving) skills amongst children, one particularly suited to the youngest school children is educational robotics. Hence, the primary aim of this thesis was to explore how educational robotics could be used as a tool in early primary school classrooms to develop children's computational thinking skills. In this thesis, I present a collection of studies that investigated the integration of computational thinking, programming, and robotics from two perspectives: (1) the child and (2) the teacher.

In this introductory chapter, I explore the importance of incorporating computational thinking within primary school curriculums, not only for meeting the needs of the economy, but for also benefitting children's lifelong learning skills. I also provide an overview of how primary school curriculums have changed to incorporate these concepts. I then explore what computational thinking is, how it is linked to programming education and how these ideas can be introduced within early years settings. As educational robotics are commonly used with children under the age of 8, I then summarise the benefits of using robotics with young children, as illustrated by the findings of previous research. I also highlight the limitations of past research and emphasise how my research contributes to this body of literature. Finally, I explore why it is important that research investigates the role of the teacher and teachers' beliefs in this area (i.e., due to their impact on teaching practices and pupil outcomes).

Advancements Within the UK Job Market

The demand for STEM (Science, Technology, Engineering and Mathematics) skilled workers in the UK is on the rise. In fact, it has been noted that the number of STEM occupations is growing at a rate six times faster than non-STEM occupations (Noonan, 2017). As a result, there is a significant employer demand for digitally skilled workers. Digital skills range from basic skills such as those needed to carry out an internet search, to advanced digital skills required for specialised work. One in four UK employers (27%) report that most of their workers require skills at a more advanced level (WordSkillsUK, 2021). The term *"advanced digital skills"* refers to having good knowledge across basic digital skills, as well as in-depth knowledge in one or more areas (e.g., programming or specialist digital software; Kipster, 2018). Recently, over one in three (37%) employers surveyed by WorldSkillsUK (2021) reported that their current workforce lacks the advanced digital skills they require.

To combat this skills shortage and prevent it from damaging the country's economic competitiveness and productivity, urgent action is needed from governments, schools, and industry leaders with a focus on education (House of Commons, 2016). Consequently, there is now an emphasis on preparing young people with 21st-century skills through STEM-related teaching, starting as early as primary school. Officials hope that the delivery of a high-quality computing curriculum will help target digital skills, thus minimising the discrepancy between the advanced digital skills needed to adequately support the current economy and the skills young people take into the workplace following schooling.

Changes to Curriculums – International Initiatives

In the last decade, there has been a global push to move away from traditional Information and Communication Technology (ICT) curriculums to more specialised computing curriculums. In England, computing replaced ICT in 2014 and has since been mandatory in schools from the ages of 5 to 16. The previous ICT curriculum was criticised for not providing pupils with the computing skills required to meet the demands of industry and higher education (House of Commons, 2016). For instance, ICT curriculums previously focused on studying information and communication and how it is used within society. Thus, children learnt about different ways of communicating and collecting information using computers. In a 2012 speech, former Education Secretary Michael Gove criticised the ICT curriculum stating, *"Instead of children bored out of their minds being taught how to use Word and Excel by bored teachers, we could have 11-year-olds able to write simple 2D computer animations using a MIT tool called Scratch"* (Department for Education, 2012). Thus, the new computing curriculum has since focused on developing children's knowledge of *how* technology works so they can be *creators* rather than remaining *consumers* of technology and only using it at a basic level. As mentioned by Michael Gove, programming (through digital tools like Scratch) is one aspect of computing whereby children can learn more about the inner workings and processes of technology and begin to create their own content. Even with the emergence of artificial intelligence, skills like programming have very low rates of automation (PwC, 2021) compared to administrative programs like Word and Excel. Thus, Governments see programming as a skill which will remain valuable for years to come. Moreover, they see the value in the additional skills often developed alongside programming (i.e., problem solving) and envision how these broader skills can be applied to other areas of learning.

Consequently, many countries across Europe have integrated programming into school curriculums at the national, regional or local level, including: Austria, Belgium, Bulgaria, the

Czech Republic, Denmark, Estonia, Finland, France, Hungary, Ireland, Israel, Lithuania, Malta, Poland, Portugal, Slovakia, Spain, Turkey and the United Kingdom (Bers 2020, Balanskat & Engelhardt, 2015; European Schoolnet, 2015; Uzunboylu et al., 2017). Similar movements have also occurred outside of Europe as countries like Australia, Singapore, Argentina and the United States established clear frameworks for introducing technology and programming within primary and secondary education (Australian Curriculum Assessment and Reporting Authority, 2015; Digital News Asia, 2015; Jara et al., 2018; Siu et al., 2003; Smith, 2016).

In line with these global developments, the Welsh Government have also made boosting digital skills, including computational thinking, a priority. The research projects presented in this thesis will focus on the Welsh education system. Since 2018, curriculum guidance in Wales has highlighted *"digital competence"* as a mandatory skill that must be embedded across the curriculum. The term *digital competence* was first introduced by the Welsh Government in the Digital Competence Framework (DCF; Hwb, 2018). This guidance is now offered to teachers in Wales as a Cross-Curricular Skills Framework. In this context, digital competence has been defined as *"The set of skills, knowledge and attitudes that enable learners to use technologies and systems confidently, creatively and critically"* (Hwb, 2018). Framework guidance clearly emphasises that digital competence should not be confused with ICT, further illustrating a step away from the old curriculum. The framework recognises digital competence as a distinct area of learning that should be cross-curricular (like literacy and numeracy) and should focus on developing digital skills that can be applied to a wide range of subjects and scenarios. Such skills are thought to benefit pupils in the long run, regardless of future career directions and technological advancements. Thus, guidance prompts teachers to begin introducing these digital skills from primary schooling.

In addition to the DCF, Wales introduced a *"New Curriculum for Wales"* in 2020 (Hwb, 2024a), that was first implemented in 2022 for those aged 3 to 16 years. This new curriculum provided guidance to help schools develop their own curriculum and highlighted *science and technology* as one of six specialist areas of learning and experience (see Figure 1.1). Within this specialised area, teachers are provided with six *"descriptions of learning"*, which provide guidance on how learners should progress within each area. One of the six descriptions within the specialised area of *science and technology* is *"Computation is the foundation for our digital world."* Teachers are then given lists of skills that children should achieve as they progress through five distinct progression steps (see Figure 1.2).

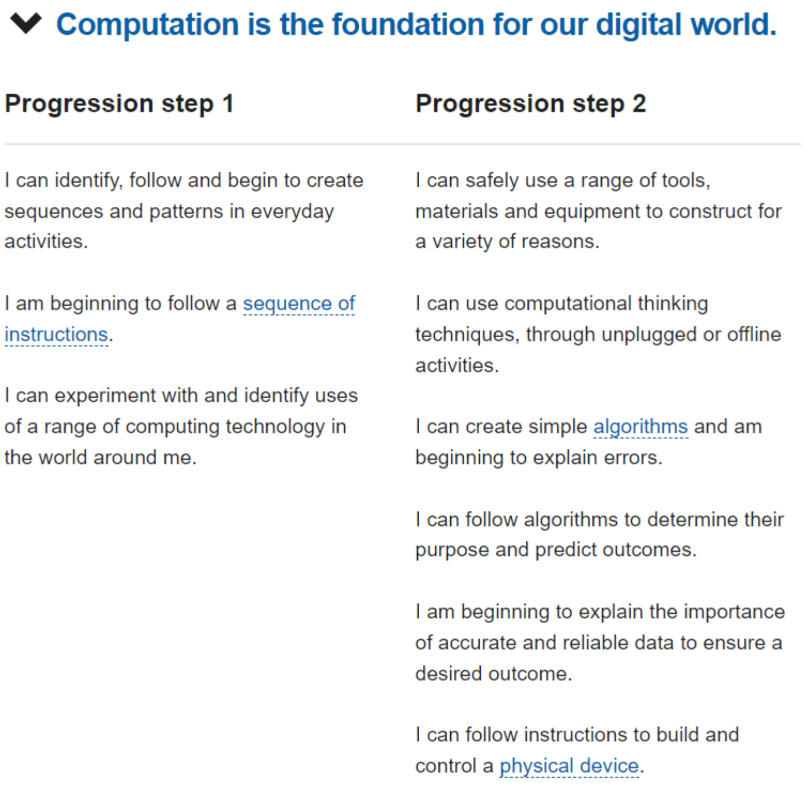
Figure 1.1

The Six Specialist Areas of Learning and Experience Highlighted in the New Curriculum for Wales.

Note: Taken from https://hwb.gov.wales/curriculum-for-wales/.

Figure 1.2

Progression Steps Provided to Teachers as Learning Goals for their Pupils.



Computation is the foundation for our digital world.

Progression step 1	Progression step 2
I can identify, follow and begin to create sequences and patterns in everyday activities.	I can safely use a range of tools, materials and equipment to construct for a variety of reasons.
I am beginning to follow a sequence of instructions.	I can use computational thinking techniques, through unplugged or offline activities.
I can experiment with and identify uses of a range of computing technology in the world around me.	I can create simple algorithms and am beginning to explain errors.
	I can follow algorithms to determine their purpose and predict outcomes.
	I am beginning to explain the importance of accurate and reliable data to ensure a desired outcome.
	I can follow instructions to build and control a physical device.

Note: These steps describe how learning should progress within each "description of learning". Screenshot taken from https://hwb.gov.wales/curriculum-for-wales/science-and-technology/descriptions-of-learning/, illustrating progression steps one and two which correspond to learning expectations in primary school years.

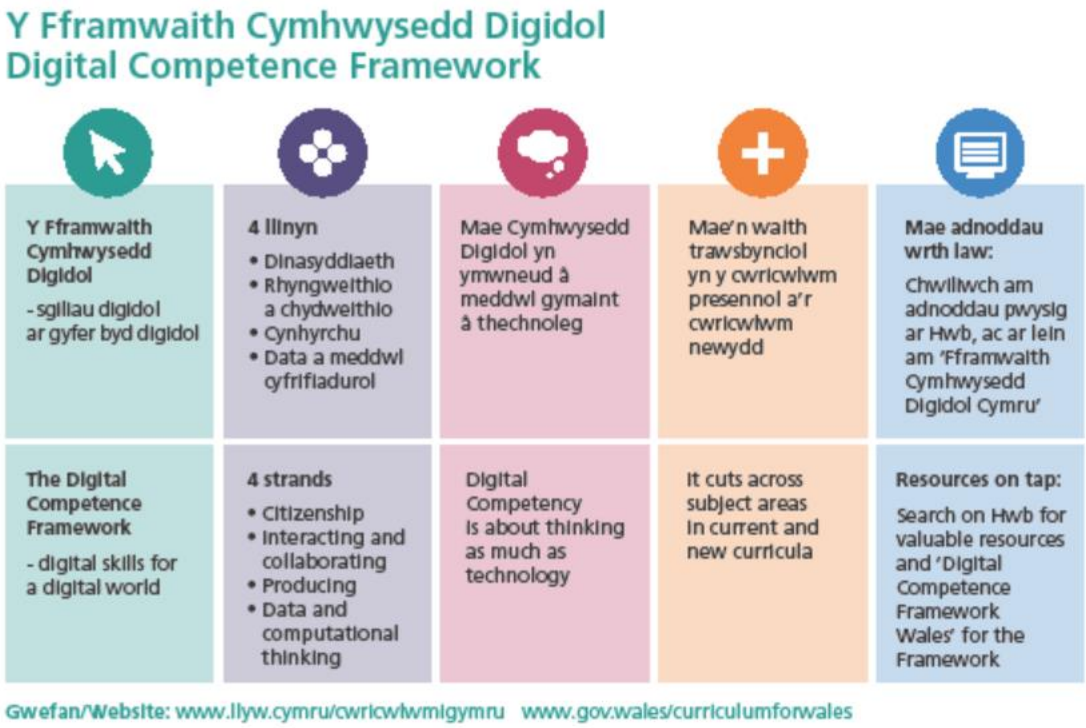
Computational Thinking (CT)

The DCF highlights *"data and computational thinking"* as a key learning area when it comes to developing digital competency (see Figure 1.3). Curriculum guidance for teachers defines computational thinking (CT) as "*a combination of scientific enquiry, problem solving and thinking skills."* (Hwb, 2018). Within academia, however, there is little agreement on a formal definition of CT (Bers, 2020; Tang et al., 2020). Generally, CT is thought to encompass a broad set of analytic and problem-solving skills most often used in computer science but that can serve everyone (Barr et al., 2011; Barr & Stephenson, 2011; Computer Science Teachers Association, 2020; Lee et al., 2011). The idea that CT skills can benefit everyone, not just those working in technical roles or children learning with computer technologies, is important.

Although CT was originally identified as a problem-solving skill grounded in computer science, in 2006, Jeanette Wing presented a paper highlighting that these skills are universally applicable to all individuals, not just computer scientists. Wing later defined CT as *"the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent"* (Wing, 2011). Thus, Wing suggests that CT should be considered an every-day life skill and should be a part of every child's analytic ability.

Figure 1.3

The Different aspects of the Digital Competence Framework (2018).



Note: A screenshot taken from https://educationwales.blog.gov.wales/2018/02/27/digital-competence-framework-wall-chart-and-postcard/.

But how are these seemingly complex ideas applicable within early childhood education? In her 2020 book, *Coding as a Playground,* Marina Bers highlighted powerful ideas from CT and illustrated how they align with traditional early childhood education concepts and skills (see Table 1.1). Here she simplifies CT ideas, identifying skills that teachers can focus on when instructing young children. For example, debugging skills can be broken down into several stages such as identifying problems (i.e., checking their work for mistakes), solving problems (correcting the mistake) and perseverance. Additionally, when it comes to developing children's understanding of hardware and software, teachers can start simply by explaining to children that smart objects (i.e., computers, cars etc.) do not work on their own or by magic, and instead need to be operated by a human. These examples illustrate how complex CT ideas can be taught in a way that is accessible and developmentally appropriate for children in early primary education. Furthermore, they illustrate how CT consists of generalisable skills that can be applied to areas of everyday life, regardless of children's future career prospects. After all, not all children will grow up to take on roles explicitly related to computing or programming.

Table 1.1

Powerful Ideas and Early Childhood Education (Bers, 2020).

Powerful idea	Related Early Childhood Concepts and Skills
Algorithms	• Sequencing/order (foundational math and literacy skills) • Logical organisation
Modularity	• Breaking up a large job into smaller steps • Writing instructions • Grouping a list of instructions into a given category or module to complete a larger project
Control Structures	• Recognising patterns and repetition • Cause and effect
Representation	• Symbolic representation (i.e., letters represent sounds) • Models
Hardware/Software	• Understanding that "smart" objects do not work by magic (i.e., cars, computers, tablets etc.) • Recognising objects that are human engineered
Design Process	• Problem solving • Perseverance • Editing/ revision (i.e., in writing)
Debugging	• Identifying problems (checking your work) • Problem-solving

- Perseverance

Computational Thinking and Programming

Many of these CT ideas are frequently targeted within programming education. European Schoolnet emphasised that programming is progressively emerging as a critical competence, stating that it *"is part of logical reasoning and represents one of the key skills which are part of what is now called 21st-century skills"* (European Schoolnet, 2015, p. 2). For example, the ability to *"create simple algorithms"* is emphasised as a programming and CT skill within the new curriculum for Wales. The term *"algorithms"* is defined for teachers as *"Processes or sets of instructions to be followed in calculations of other problem-solving operations, especially by a computer"* (Hwb, 2024b). Bers previously proposed that sequencing activities can help deepen children's understanding of algorithms (Bers, 2020). Sequencing has been defined as a series of individual steps or instructions ordered to achieve a desired outcome (Brennan & Resnick, 2012). These can be instructions that are unrelated to programming or computing. Activities that target sequencing skills align with New Curriculum learning goals for early years pupils such as, *"I can identify, follow and begin to create sequences and patterns in everyday activities"* and *"I am beginning to follow a sequence of instructions"* (Progression step 1, Hwb, 2024b).

Debugging is another ability classically recognised as a CT skill (Shute, et al. 2017). Within programming education, debugging has been defined as identifying and fixing errors in an algorithm (Bers et al. 2019). Bers et al., (2014) broke down the concept of debugging into four aspects: (1) recognition of the issue (i.e., the child must first recognise that the algorithm does not result in the desired outcome); (2) goal evaluation (the child must decide whether to stick with their original goal or come up with a new one); (3) hypothesis generation (creating a prediction for the problem's cause) and (4) problem-solving (attempting to resolve the problem). Debugging skills are highlighted as a primary school progression step within the New Curriculum for Wales, as teachers should encourage children to begin explaining errors in algorithms (Progression step 2; Hwb, 2024b).

Curriculum guidelines also suggest that children should learn to *"follow algorithms to determine their purpose and predict outcomes"* (Progression step 2; Hwb, 2024b). Children's ability to think logically (another fundamental aspect of CT; barefootcoding.org) supports the development of the ability to formulate predictions. For example, a child may use the knowledge they have about the function of different programming instructions to anticipate what an algorithm will do. In summary, programming education serves as a dynamic platform through which key CT skills, such as algorithm writing, sequencing, debugging and prediction skills can be developed in young learners. These skills can then be applied to various areas of learning (Hoppe & Werneburg, 2019).

Methods of Teaching Programming in Early Education

In the realm of early childhood education, hands-on learning theories have informed approaches to teaching programming to children. Teaching methods have evolved with three main methods: screen-based technologies, unplugged learning approaches, and tangible technologies. Each of these methods aims to promote learning using playful techniques, grounded in the belief that play fosters language development, social skills, creativity, and critical thinking in children (Strawhacker et al., 2018).

Theories of Hands-On Learning

Several educational and developmental theories have significantly shaped our understanding of how children learn, suggesting physical experience is an important part of learning. The Montessori education approach, initiated by Dr. Maria Montessori in the early 20th century, is a child-centred method grounded in the belief that children are inherently motivated to explore and learn (Montessori, 1967). Montessori classrooms are typically designed to be interactive environments, where children engage in hands-on learning using specially crafted materials (e.g., enamelled metal, wood and fabric). These materials are designed to develop various skills, from sensorial perception to mathematical understanding and language acquisition (see Marshall, 2017 for review). The emphasis on hands-on learning is a core principle of Montessori education, aligning with the idea that children learn best through direct, tactile experiences. Teachers in the Montessori system act as guides, carefully observing each child's unique strengths and challenges, and tailoring their approach to individual needs. Thus, the role of the teacher extends beyond traditional instruction; they create an environment that encourages curiosity, independence, and self-discipline (Kiran et al. 2021; Montessori, 1967).

Following the work of Montessori, Jean Piaget’s constructivist approach to children’s cognitive development posited that learning is shaped by a child’s physical experiences within their environment (Piaget & Cook, 1952). For example, descriptive studies of children revealed that skills of abstract reasoning and the ability to manipulate symbols arise from children’s exploratory actions within the immediate concrete environment (Kolb, 2014). As a result, children can often solve problems when given concrete materials to manipulate before they can solve them abstractly. Piaget and Inhelder's (1956) conservation task provided empirical evidence to support this notion. In their experiment, children poured water back and forth between wide and narrow containers. Piaget and Inhelder observed that children who engaged in hands-on pouring activities were more likely to demonstrate an understanding of volume conservation compared to those who just observed a demonstration. Thus, these findings emphasised the significance of hands-on experiences in shaping a child's perception of abstract concepts (i.e., volume conservation).

Seymore Papert (1928 – 2016; inspired by the works of Montessori and Piaget) explicitly linked hands-on learning theories to programming skill development. Papert developed his constructionist theory of learning which has its basis in the theory of constructivism (Piaget & Cook, 1952). Constructionism proposes that learning occurs best when using physical artefacts within social settings. These objects are used *"to think with"* and act as transitional mediums to facilitate understanding of complex symbolic concepts, thus supporting concrete ways of thinking (Papert & Harel, 1991).

Following Montessori's, Piaget's and Papert's ideas, a new movement surrounding curriculum development and teaching emerged. These changes utilised the principles of their theories for the advancement of experience-based educational programmes. Guidance from the New Curriculum for Wales explicitly states that when designing and planning the science and technology aspects of the curriculum schools should, where relevant, facilitate learning through active and practical experiences (Hwb, 2024b). Thus, inside the classroom, teachers have used various methods to introduce programming education within early schooling, which encourage hands-on, playful learning in line with theoretical recommendations. These include interactive screen-based technologies, unplugged learning methods and educational robotics devices.

Screen-Based Technologies

Screen-based programming applications like Scratch (Resnick et al., 2009) promote programming knowledge through playful learning by offering a user-friendly and visually engaging environment. Additionally, its drag-and-drop interface allows children to "snap" programming bricks together in different combinations without the obscure syntax of traditional programming languages (Resnick et al., 2009). In other words, Scratch uses a visual programming environment to eliminate syntax error problems thus allowing pupils to focus on algorithm creation and problem-solving. Scratch then enables first-time programmers to master constructs of computer programming through the creation of games and interactive art, which taps into children's natural inclination for imaginative play (Goldstein, 2020).

As a result of the interest in creating a program more accessible for younger children, Scratch Jr was developed (Strawhacker et al., 2015). Scratch Jr (www.scratchjr.org) is a programming language which enables children aged 5 to 7 years to create interactive stories and games by connecting graphical programming blocks (Bers, 2018). Compared to the original Scratch program, Scratch Jr consists of simplified programming commands that do not use written language, thus making it suitable for pre-literate children. The program is designed to support problem-solving by reducing unnecessary low-level cognitive burdens. For instance, the simplified block instructions keep the programming difficulty at a developmentally appropriate level, thus allowing children to allocate cognitive resources to the high-level thinking processes involved in imagination and creativity (Bers, 2018).

Initially, these programming languages were operated solely on computers, however, programs like Scratch Jr are now available as applications on touchscreen devices (i.e., tablets and smartphones). As this technology is no longer restricted to a desktop computer, it is more accessible to younger children. Traditional keyboard-based devices (i.e., desktop computers and laptops) require a certain level of cognitive development to understand the keyboard symbols along with sufficient fine motor development to use the keyboard and mousepad, making these devices developmentally inappropriate for very young children (Geist, 2014).

However, while applications like Scratch Jr may be advantageous in some ways, their screen-based nature may be off-putting for some parents and teachers. Concerns about the effects of too much screen time have increased following the Covid-19 pandemic as children's screen-viewing behaviours increased greatly during lockdown periods (Salway et al., 2023). This was primarily due to gaming, video calling, television, and online learning. Consequently, in September 2023, the UK government launched a new inquiry into screen time following reemerging concerns about its effects on children's education and wellbeing (UK Parliament, 2023). As concerns about the effects of increased screen time on children's education and well-being gain prominence, educators will be seeking alternative approaches to programming education. Both unplugged learning approaches and educational robotics can be used to deliver programming and CT content via active and practical experiences.

Unplugged Learning Approaches

Unplugged methods aim to expose children to CT and programming without using computers or other screen-based technologies (i.e., tablets and mobile devices). Thus, such methods aim to build a foundational understanding of programming principles through tactile and interactive learning experiences (Bell et al., 2009). Unplugged programming activities often include games, puzzles, and interactive exercises that simulate the CT skills required in computer programming. For example, games that involve step-by-step instructions to achieve a goal (i.e., describing how to make a sandwich) can help pupils understand the concept of algorithms.

A recent study from del Olmo-Muñoz et al., (2020) aimed to investigate whether it was more beneficial to introduce CT in early years classrooms through unplugged activities before plugged-in (screen-based) activities or whether teachers should teach CT exclusively through screen-based activities. Eighty-four children (aged 5 to 6 years) completed CT assessments before and after the CT program. They also completed five 45-minute sessions utilising activities from the website Code.org and these sessions either utilised unplugged teaching methods or plugged-in methods. Code.org activities were chosen by authors as they were seen to offer a good combination of both unplugged and plugged-in activities that were similar in content. The authors analysed the development of CT skills by comparing participants in the unplugged condition to those in the plugged-in condition. Children's CT skills were measured using a paper-based assessment which consisted of a collection of items from the 2016 and 2017 editions of the International Bebras Contest (Dagienė & Sentance, 2016). Average CT scores were reported on a scale of 0 to 10. Mann-Whitney U analyses showed that, on average, there were significantly larger improvements in CT scores in the unplugged learning group ($M = 3.76$, $SD = 2.13$) in comparison to the plugged-in group ($M = 2.44$, $SD = 1.65$). The authors concluded that it is more appropriate to work on CT in early years classrooms through an approach that introduces unplugged activities before screen-based (plugged-in) activities and that the approach was more beneficial than using screen-based methods alone.

These findings support the guidance provided by the New Curriculum for Wales. The guidance states that unplugged activities should be used throughout education to help pupils visualise computational concepts (Hwb, 2024b). It emphasises that hands-on, practical activities with a range of tools are especially relevant for teaching principles of programming and developing a deeper conceptual understanding of key syntax and constructs before implementation and application. However, Welsh curriculum guidance also instructs teachers to provide learning experiences that bridge the physical and digital worlds through devices that pupils can interact with and manipulate their environment (Hwb, 2024b). A limitation of unplugged learning activities is that they do not provide these technology integrated experiences. Educational robotics, however, can help teachers combine physical and digital learning.

Educational Robotics (ER)

Educational robotics (ER) are devices that combine hands-on learning with tangible technology. These technologies have developed greatly over the last few decades. In the 1960s, Papert and his colleagues at the Massachusetts Institute of Technology (MIT) created LOGO, an educational programming language. LOGO programming allowed children to write keyboard commands that produced line graphics in their concrete environment with a small robotic turtle. This unique approach provided an environment for playful learning about geometry and programming, marking a transformative moment in understanding complex concepts through physical interaction (Schneider, 2017).

The LOGO Turtle then paved the way for several tangible programming technologies for children. The TORTIS emerged in 1974 and was developed by Radia Perlman. Perlman worked at MIT for several years and is thought to have pioneered the introduction of computer programming with children as young as 4 years old. The development of TORTIS meant preliterate children could practice robotics for the first time (Morgado et al., 2006). With a resemblance to the LOGO Turtle, children were able to program the behaviour of a physical turtle robot using a series of mostly movement-related commands (forward, backward or rotate about its centre). Additional commands included “toot”, “pen up”, “pen down”, “light on” and “light off”. However, instead of programming these commands via a computer keyboard (which can be difficult for children still developing their motor skills), children could manipulate tangible command cards. Another important component of the TORTIS system was the Slot Machine. This provided a platform for children to place their command cards, thus allowing them to visualise their programming sequence (Perlman, 1974). Once all the command cards had been placed, children could then execute these commands at the touch of a button.

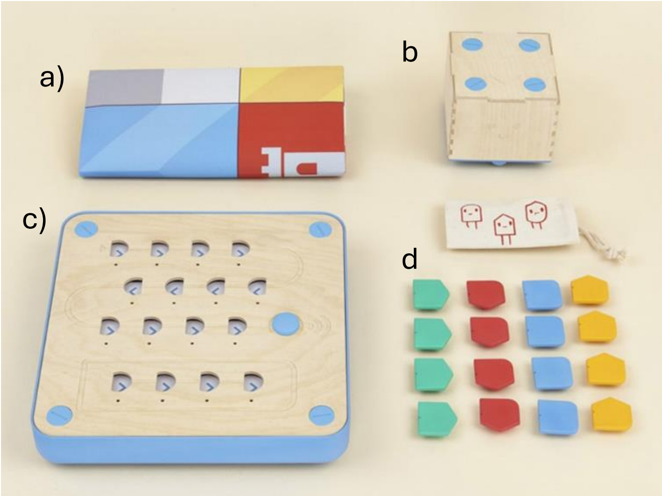
In 2011, KIBO (Sullivan et al., 2017) was developed with some of Perlman’s principles in mind. KIBO was introduced as a screen-free robotics kit specifically designed for 4 to 7-year-olds. To operate KIBO, children must link together a series of wooden blocks that each represent a different instruction (e.g., forward, turn right, spin, sing). As they scan the barcodes of each block, the sequence of instructions (the algorithm) then dictates KIBO's movements. Like TORTIS, KIBO makes programming accessible to younger children by minimising the level of reading proficiency required through simplified tangible instruction blocks and provides children with a clear display of their algorithms. However, KIBO's instruction blocks have been criticised for being difficult to scan. González-González and colleagues (2019) reported that children in their study often did not wait for the beep or LED to confirm that the code had been read before continuing with their algorithm, leading to mistakes in the sequence.

Bee-bot robots (www.tts-group.co.uk) are the most popular type of robots used with young children in developmental research across the globe (Garvis & Keane, 2023). The Bee-bot is shaped like a bee and has a yellow and black casing. Children can operate Bee-bot robots by pressing buttons on top of the device. These include move forward, move backwards, turn left, turn right, pause, and reset. However, Bee-bot robots store these commands internally, meaning the child does not have a visual representation of their algorithm. Bakala et al. (2021) explain that this kind of user interface makes programming cognitively demanding due to the high load on working memory, whereby the child needs to remember an entire sequence of code. They go on to suggest that these interfaces may therefore limit programming activities to sequencing-only tasks as additional tasks like debugging and prediction are more challenging without a visual algorithm.

The more recently developed Cubetto robot (www.PrimoToys.com) appears to overcome the limitations identified from research with KIBO and Bee-bot robots. Primo Toys, the makers of Cubetto, support the notion that programming is the new literacy and therefore believe it should be prioritised at an early age. In 2016 this led them to develop Cubetto, a cuboid robot designed for children aged 3 to 7 years old. Inspired by Montessori values, Primo Toys designed Cubetto to be hands-on, collaborative, and made from wood (Migliaresi, 2016). Like other educational robots, Cubetto's design completely avoids textual and numerical language thus making it suitable for pre-literate children. Cubetto comes equipped with an interface board, a range of tangible function tokens (forward, right, and left turn functions) and a colourful floor map (see Figure 1.4). Children can navigate the robot around the map by placing the desired tokens in the interface board and pressing the 'Go' button.

Figure 1.4

Cubetto Playset.



Note: Each set comes with a floor map (a), a Cubetto robot (b), an interface board (c) and a range of movement tokens (d).

The addition of an interface board for children to place their tokens into sets Cubetto apart from the KIBO and Bee-boot robots. For instance, children can easily place tokens into the board without having to scan them (like with KIBO). Additionally, Cubetto overcomes Bee-bot's problems of program visualisation by allowing children to physically manipulate their tokens in its large interface board. In doing so, children do not need to remember the commands they have given to the robot. Instead, they have a clear view of their algorithm, thus allowing them to better predict the outcome of their sequences and to detect errors in their algorithms. For these reasons, Cubetto robots will be used in the research projects described in this thesis.

The grant used to fund the research presented in this thesis was obtained with the co-operation of Primo Toys. Furthermore, Primo Toys provided the Cubetto equipment used in both laboratory research (presented in Chapter 4) and school-based research (presented in Chapters 5 and 6). Primo Toys was not involved in discussions regarding the design of the studies I conducted, nor did they play a role in the collection or analysis of the data; interpretation of the results; or writing of this thesis.

Past Educational Robotics Research. As ER have been identified as an appropriate learning tool for children in early primary education, past research has investigated how ER can be used to aid the development of CT skills. In this section I now review past research in this area, to provide an overview of how ER have been utilised to enhance CT skills like sequencing, debugging and prediction skills.

Sequencing. Sequencing is a skill examined extensively by researchers (Ching & Hsu, 2023). Kazakoff et al., (2013) demonstrated that ER could improve children's sequencing following a short ER intervention. Using CHERP (a hybrid tangible and graphical computer language; Bers, 2010; Horn et al. 2011) They assessed the effect of a 1-week robotics curriculum on children's picture-sequencing skills (n = 29, aged 4 to 6 years, mean age = 4.77). CHERP allowed children to write instructions for a physical robot using either tangible or graphical block instructions. Kazakoff and colleagues concluded that participation in an intensive ER and programming curriculum improved children's picture-sequencing skills relative to a control group (a 'learning as normal' group who continued with their regular curriculum). Furthermore, Strawhacker, Sullivan and Bers (2013) investigated whether programming interventions across 3 different interfaces impacted performance on a sequencing assessment. Kindergarten children (n = 36) were allocated to one of three CHERP programming conditions: tangible interface (i.e., robot learning), graphical interface (i.e., screen-based learning) or hybrid interface (i.e., combined robot/ screen-based learning). They found significant differences between the tangible and graphical interface conditions, whereby children in the tangible interface condition answered a higher percentage of sequencing assessment trials correctly post-intervention (73.3%) compared to children in the graphical interface condition (12.5%). The findings of these studies suggest that children as young as 4 years-old can improve their sequencing abilities through ER activities and that these tangible interfaces can provide better learning outcomes than screen-based interfaces.

Debugging. Researchers have also used ER interventions to develop children's debugging abilities. Across 6 robotics sessions, Bers et al., (2014) found that children as young as four years old could demonstrate debugging skills with a programmable robot (CHERP). Furthermore, research from Pugnali et al., (2017) evidenced the advantages of using ER to develop children's (aged 4 to 7 years, n = 28) debugging skills rather than screen-based programming systems. Their study assessed the impact of a ~15-hour programming curriculum and explored whether learning outcomes varied based on the method of instruction (i.e., tangible robot KIBO or screen-based programming language Scratch Jr). Results showed that children learning in the ER condition performed significantly better on paper-based debugging tasks at the end of the course than children using the screen-based interface. These debugging tasks utilised KIBO or Scratch Jr commands depending on condition allocation.

Most recently, large-scale research from Misirli and Komis (2023) employed a sample of 526 children (aged 4 to 6 years) to investigate how children engage in debugging practices with a tangible robot. Children used physical command cards to write their algorithms before inputting the instructions into a Bee-bot robot. Misirli and Komis referred to these cards as a 'pseudocode' to aid children's visualisation of their algorithm. Their findings suggested that teaching children programming skills using a Bee-bot robot encouraged children's development of CT skills, including debugging. Furthermore, the authors suggested that debugging skills can be developed through open play with a tangible robot, and not only with guided 'step-by-step' strategies. These studies illustrate the potential of ER in cultivating debugging skills among young children. However, it is worth noting that these studies did not employ control groups thus limiting the ability to draw definitive conclusions about the causal relationship between the robotics interventions and the observed outcomes.

Prediction. Research has investigated how ER can be used to help children practice prediction skills. Slangen, Keulen, and Gravemeijer (2011) investigated what children (aged 10 to 11 years) could learn from working with Lego Mindstorms NXT robots. During programming lessons, children were presented with algorithms of increasing difficulty and were tasked with predicting the robot's behaviour based on these sequences. Upon using qualitative methods to observe children's learning, the authors concluded that experience with robots challenged children to predict, hypothesise and then test their assumptions. These observations suggest that ER can be used as a tool to develop aspects of children's logical thinking skills through prediction tasks. However, to the best of my knowledge, research is yet to investigate the relation between ER learning and prediction abilities using quantitative methods, a control comparison group and samples of younger children.

Spatial Skills. In addition to CT skills, research has also investigated potential links between experiential learning with ER and the development of spatial skills. Spatial visualisation refers to the ability to mentally transform and manipulate objects (Uttal et al., 2013) and may be important for programming ER successfully due to the physicality of these robots and their presence within a child's physical environment. When programming a robot, if its orientation is incongruent to the programmer's (i.e., the child's), then spatial visualisation and perspective taking are required to pre-plan an algorithm. Although not regularly categorised as a CT-specific skill, spatial visualisation skills have been suggested to be highly relevant to CT abilities and programming performance (Jones & Burnett, 2008). For example, with a sample of 92 primary students (aged 6 to 10 years), Città et al. (2019) assessed children's programming abilities (using a pencil-paper algorithm writing task) and spatial skills (using a mental rotation task). They identified a positive correlation between programming and spatial skills.

Research exploring whether learning with ER can improve spatial skills using child samples has produced mixed results. Sisman et al., (2021) used ER to develop the spatial skills of 34 children (aged 8 to 12) over a 31-week period. Spatial visualisation and mental rotation tasks were used to determine children's spatial ability. Sisman and colleagues found that children's spatial visualisation and mental rotation abilities had significantly improved at the end of the robotics course in comparison to a control group. However, research from González-Calero et al., (2019) found gender differences in the impact of ER learning on children's mental rotation abilities (sample aged 7 to 8 years). Although they found that males in a 2-hour ER exposure condition demonstrated higher performance on a mental rotation task than a control group, they found no significant difference between these groups amongst a female sample. Furthermore, Diago et al., (2022) also investigated the effects of ER engagement amongst children (aged 8 to 9 years) and found no evidence of improved mental rotation abilities post-intervention. In summary, the evidence for whether ER may improve children's spatial abilities is mixed and thus questions remain about spatial visualisation and its relation to programming in these learning scenarios. For instance, whether spatial skills aid programming performance, whether programming activities can aid the development of spatial skills, or both. Thus, the studies presented within this thesis aimed to investigate these questions. Furthermore, these projects did so in samples of children under the age of seven as there is a lack of research investigating the relation between spatial abilities and programming in younger children.

Children's Beliefs. Past research has also assessed the effect of ER curriculums on children's beliefs about robotics. Beliefs towards these topics are thought to be multidimensional, involving a combination of self-efficacy beliefs (how confident a child is in their ability to use a robot or complete programming tasks) and beliefs in the value or importance of these topics (Eccles & Wigfield, 2002).

An intervention study presented within this thesis (Chapter 5) investigated pupil beliefs in addition to their CT abilities. This was important given that past evidence on whether ER exposure improves children’s value and self-efficacy beliefs is mixed. On one hand, some have found that using robotics in education can improve children’s attitudes. For example, one study (Zviel-Girshin et al., 2020) implemented a robotics curriculum (with lessons once or twice a week for seven months) and surveyed 84 children (aged 5 to 7 years). Researchers reported that children felt confident in their ability to use robots and held positive beliefs about continuing learning with robotics in the future. On the other hand, others have reported negative effects on children’s beliefs (Hussain, 2006; Leonard et al., 2016). For example, Leonard et al, (2016) found that 10 to 11-year-old children’s self-efficacy decreased following a robotics program that lasted 6 to 10 weeks.

It is important to investigate whether ER exposure in primary school may positively or negatively impact children's self-efficacy and value beliefs as past research has found children’s early STEM (Science, Technology, Engineering, Mathematics) beliefs can impact later learning outcomes and continued interest (Simpkins et al., 2006). One of the research studies presented in this thesis aimed to do just that. As part of a school-based intervention (see Chapter 5) children’s self-efficacy and value beliefs were assessed (using Likert-scale style responses) pre- and post-intervention to investigate potential changes following controlled robotics exposure during a six-week program.

Overall, research has shown how ER can be used to support children’s learning of programming and the development of CT skills. Additionally, research has investigated children’s beliefs about programming and robotics. However, it appears that some areas (i.e., sequencing and debugging) have a larger, or more conclusive evidence base than others (i.e., prediction skills, spatial skills, and child beliefs). Additionally, some of these areas are lacking empirical research with younger children (i.e., under the age of 7 years). Thus, the research projects in this thesis have contributed to these areas of the literature by exploring how ER can be used to aid the development of sequencing, debugging, prediction and spatial visualisation skills, in children aged 4 to 7 years. Additionally, my intervention study aimed to investigate the effects of robotics exposure on children’s beliefs. Importantly, the research presented here has employed qualitative and quantitative methods and control groups.

The new curriculum for Wales already encourages learning with tangible technologies by identifying goals for pupils such as “*I can follow instructions to build and control a physical device”* (Hwb, 2024b). Further investigations into the use of ER with children in early primary education may provide more evidence supporting the use of these devices in early childhood classrooms. Additional research may also highlight how ER can be used most effectively within the classroom to optimise children's learning and development.

The Role of the Teacher

To explore how robotics can be successfully integrated within early primary school classrooms, the research presented in this thesis not only investigated CT, programming, and robotics from a pupil perspective but also a teacher perspective. To my knowledge, past research evaluating the benefits of robotics for the development of children's CT skills had not simultaneously explored the role of the teacher. Instead, many intervention studies have been implemented by researchers thus minimising the role of classroom teachers (see, for example, Sullivan & Bers, 2015; Sullivan & Bers, 2016; Sullivan et al, 2013; Strawhacker & Bers, 2015). It is important to integrate teachers into intervention research as they are the ones responsible for implementing classroom and curriculum changes in the long term. Thus, the second half of this chapter explores the role of the teacher in CT, programming, and robotics education.

Clearly, there are benefits to using ER to support children's learning of programming and the development of CT skills. Additionally, these technologies are suitable for achieving many of the goals outlined in primary school curriculums globally and within the new curriculum for Wales. However, CT, programming and robotics are somewhat new additions to these curriculums. This may make integrating these concepts into teaching practices more difficult for certain teachers. For example, for some, these topics may not have been covered in their teaching degree programs if they qualified several years ago. Thus, teachers may be unaware of *how* to teach programming and robotics education; additionally, they may not know *why* teaching these topics may be beneficial for children's learning. Such beliefs may result in teachers avoiding these topics (Ertmer, et al., 2012; Larke, 2019) which may negatively impact children's learning as a result. Thus, it is important to investigate practising teachers' beliefs about CT, programming and robotics, and whether they are attempting to integrate these concepts within their classrooms. How confident do practising teachers feel about teaching these topics? Do teachers believe it is important to integrate these topics within early years classrooms?

Previously, research has characterised teacher beliefs in two ways: (1) value beliefs and (2) self-efficacy beliefs. Firstly, value beliefs encompass the perceived importance of specified goals and choices (Anderson & Maninger, 2007). Secondly, teachers' self-efficacy beliefs are conceptualised as their confidence in their ability to bring about desirable changes in pupils' behaviours, motivation, and achievement (Tschannen-Moran & Hoy, 2001). Both value and self-efficacy beliefs are often subject-specific. For instance, a teacher may feel confident teaching in one domain, but less confident teaching in another.

Previous research has found that low self-efficacy for programming and robotics is common in samples of primary school teachers (Khanlari, 2016; Ohashi et al., 2018; Ray et al., 2020). For example, in a survey study of 142 Japanese primary school teachers (Ohashi et al., 2018), only 4% (n = 5) of teachers reported they felt confident about teaching programming to their pupils. Similarly, survey findings from Khanlari (2016) illustrated that 82% (n = 11) of primary school teachers identified teacher confidence as a major obstacle to robotics use in Canadian classrooms. However, with regards to value beliefs, this study found that generally, teachers held positive beliefs about the value of using robotics and the potential benefits for their pupils' learning. For example, when asked whether early primary school children were too young to understand and work with robotics, 64% believed pupil age was not an obstacle. Most teachers in this study believed robotics were appropriate and valuable learning tools for young children, thus indicating positive value beliefs. However, the small sample size used in this study highlights the importance of exploring teacher beliefs in a larger sample while also using more in-depth qualitative methods. The research presented in this thesis aimed to do just that (see Chapters 2 and 3).

Understanding the Impact of Teacher Beliefs

Research that aids our understanding of teacher beliefs is important for the successful integration of new digital concepts (like programming) and technologies (like robotics) within the classroom. As suggested by previous research findings, teacher beliefs can impact teaching practices, specifically technology integration (Ertmer et al., 2012). This is because negative teacher beliefs are thought to be a large barrier to the integration of new technologies within classrooms (Ertmer, 1999; Ertmer et al., 2012). A meta-analysis by Hew and Brush (2007) analysed 48 studies to identify the most frequently cited barriers impacting technology integration. In their final conclusions, teacher beliefs ranked within the top three, along with teachers' knowledge and access to resources. These authors highlighted several studies whose findings suggested that negative value beliefs about the relevance of using technology within the classroom may prevent the integration of technology within lessons (Ertmer, 2005; Ertmer et al., 1999; Newhouse, 2001). Furthermore, when investigating the impact of teacher confidence on robotics use, one study (Elkin et al., 2014) found that when teachers were more comfortable with robotics kits and content, this increased the potential for integrating robotics into the classroom. These findings suggest that negative value beliefs about the importance of programming and robotics, or low self-efficacy regarding their ability to teach these topics may prevent teachers from teaching programming content to their pupils and from introducing new technologies (like robotics). This would be problematic for Welsh teachers as they would not achieve several aims highlighted in the new curriculum.

Not only can teacher beliefs impact their decisions about what they teach, but research has previously suggested that teacher beliefs may impact pupil outcomes. Studies have shown that teacher self-efficacy beliefs in STEM subjects like science and mathematics may be associated with pupil's self-efficacy beliefs in these subjects (Midgley et al.,1989; Opperman et al., 2019; Stipek et al., 2001). Furthermore, researchers have found evidence suggesting that teachers with higher self-efficacy may be more effective at increasing pupil achievement (Klassen et al., 2021). For example, there is evidence that teacher self-efficacy beliefs are associated with children's achievement in subjects like mathematics (Ashton, 1983; 1986; Goddard et al., 2000). Thus, it is likely that teacher self-efficacy beliefs could impact pupil outcomes in other STEM subjects including areas of technology (i.e., programming and robotics education).

Teacher self-efficacy may impact pupil achievement in two ways. On one hand, higher teacher self-efficacy may improve teaching behaviours and classroom practices (Lauermann & Butler, 2021). For example, teachers with higher self-efficacy may be more resilient than those with lower self-efficacy as they may not view teaching setbacks as personal failures. Thus, when faced with challenges in the classroom, teachers may feel more confident employing a wider range of teaching methods to overcome these challenges. For instance, education research has found that high teacher self-efficacy can show positive links with classroom practices such as planning lessons that advance children's abilities, making opportunities for meaningful learning and effectively managing classroom behaviour (Chacon, 2005; Woolfolk et al., 1990). It has also been proposed that increased teacher self-efficacy may be transferred to pupils via role-modelling processes, whereby teachers' confidence is reflected by pupils and thus this increased self-efficacy in pupils then increases pupil persistence. This is then thought to have subsequent benefits on pupil achievement (Lauermann & ten Hagen, 2021).

Overall, this evidence illustrates that teachers' personal beliefs can influence both teaching approaches and pupil outcomes. It is possible that negative teacher beliefs about programming and robotics education may engender negative self-efficacy beliefs in children and negatively impact student achievement in this subject area. Beyond this, negative beliefs about teaching programming may impact teaching effectiveness and prevent the integration of new programming practices within the classroom, thus limiting children's exposure to and knowledge of computing technologies. It is therefore important to investigate teacher beliefs when new curriculum changes are implemented, as personal beliefs may impact curriculum take up and execution.

The Importance of Effective Teacher Education

Teacher education is important for ensuring teachers feel ready to integrate ER into their teaching practices (Gavrilas et al., 2024). For example, education research has illustrated that effective teacher education is likely to have positive impacts on teacher knowledge, teacher confidence (Konen & Horton, 2000) and pupils' educational outcomes (Hattie, 2003). To improve the effectiveness of these teacher programs, those designing the programs should be aware of the advantages of *"teacher education"* over *"teacher training"* (Stephens et al., 2004).

Stephens and colleagues (2004) argued that the terms *training* and *education* signify different pedagogical approaches. Using content analysis methods, they investigated differences between Postgraduate Certificate of Education (PGCE) courses in England, where policymakers speak of teacher training, and in Norway, where they refer to teacher education. They note that teacher training typically emphasises the development of practical skills necessary for specific teaching tasks. Thus, training sessions may involve drills, simulations, and practice sessions to ensure that teachers can perform specific teaching tasks efficiently. Education, on the other hand, has a broader focus on intellectual and personal development. It includes a mix of theoretical and practical learning to encourage reflection, analysis, and a deeper understanding of subjects to help teachers apply knowledge across diverse contexts. This approach may have the potential to help teachers feel more comfortable and confident in adapting instructional strategies to suit the specific requirements of their classrooms, potentially promoting effective and student-centred teaching practices. Research projects presented in this thesis explored primary school teachers' beliefs and past experiences of teacher education programs relating to CT, programming, and robotics (see Chapters 2 and 3). These studies thus further investigated whether teacher programs in Wales have utilised pedagogies and methods associated with "teacher training" as opposed to "teacher education."

How to Investigate Teacher Beliefs

Given that teacher education can improve teacher beliefs (Castro et al., 2018; Chang & Peterson, 2018; Kim et al., 2015), thereby potentially preventing a negative impact on teaching practices and pupil outcomes, it is no surprise that previous studies have typically assessed teacher beliefs towards CT, programming and robotics as part of an intervention study design. Thus, beliefs have been measured before robotics exposure and afterwards to look for improvements between the two time points.

One study (Kim et al., 2015), delivered a 3-week programming and robotics course to 16 pre-service primary school teachers (i.e., teachers who had not yet completed their teaching qualification). Through pre and post-course surveys and interviews, the authors concluded that the 3-week program improved teachers' motivation, enjoyment and interest towards programming and robotics. Furthermore, research from Chang and Peterson (2018) suggested that even a single 2-hour workshop may improve teachers' beliefs. They delivered a 2-hour educational technology course to 59 pre-service primary school teachers. This course focused on CT, programming and robotics. Based on written reflections from teachers, the authors concluded that the session increased teachers' understanding of CT and its teaching applications. They also believed it improved teachers' relevance beliefs. Research findings from studies conducted by Kim et al. (2015) and Chang and Peterson (2018) indicate positive shifts in pre-service teachers' beliefs after participating in teacher education workshops. However, it is essential to interpret these findings cautiously, recognising the potential bias in self-report measures and the lack of control comparison groups. Moreover, the context of pre-service training may contribute to these positive outcomes. In these final sections, I clarify and justify my focus on practising, early primary school teachers and I explore how these teachers' beliefs and experiences may differ from pre-service primary school teachers and those teaching secondary education.

Differentiating Between Pre- and In-service Teachers

Research that solely relies on samples of pre-service teachers (individuals in the process of completing their teaching qualifications) presents a significant limitation in the broader context of educational studies. While these individuals are undoubtedly crucial participants in educational research, their status as pre-service teachers may introduce potential biases that should be carefully considered. For example, pre-service teachers typically have a larger support network during their training. Being in the early stages of their teaching careers, these individuals often benefit from guidance and mentoring from university tutors as well as other pre-service teachers and fully qualified, practising teachers (Laker et al., 2008). This easily accessible support may influence their perceptions and attitudes toward innovative teaching methods, such as the integration of robotics.

Furthermore, the support they receive during this phase might create a somewhat artificial environment, as they may be shielded from the full spectrum of challenges and responsibilities faced by experienced, full-time classroom teachers. Classroom dynamics and the evolving demands of day-to-day teaching may significantly shape educators' beliefs and practices, and pre-service teachers may not yet have had the opportunity to fully navigate these complexities. Thus, limited exposure to real-world teaching scenarios may bias their perspectives on incorporating robotics into their (future) teaching practices.

Moreover, the landscape of teacher qualification courses is constantly evolving to incorporate the latest educational trends and technological advancements. The increasing emphasis on digital skills and CT in the curriculum is a recent development that may not be fully reflected in the experiences of teachers who are part of older qualification programs. As a result, their perspectives and past experiences may not align with the current demands and expectations of current curriculum guidance. Their lack of experience with programming and robotics is likely to stop practising teachers from implementing these into their curriculums. This assumption is supported by the findings of Larke (2019) who, through classroom observations and interviews with teachers, reported that practising teachers found it difficult to include mandatory computing programs and therefore they would rather neglect it. Thus, while research involving pre-service teachers offers valuable insights, it is crucial to acknowledge the limitations associated with their status in the teaching profession. The projects in this thesis aimed to achieve a comprehensive understanding by investigating the beliefs of fully qualified, experienced teachers who navigate the complex and dynamic landscape of early education classrooms daily.

Differentiating Between Primary and Secondary Teachers

Research from Kay and Moss (2012) investigated beliefs about using ER in a sample of practising teachers. They invited 20 teachers to take part in a three-day robotics workshop. Based on self-reported, teacher questionnaire responses, they claimed that their robotics workshop increased teachers' confidence regarding robotics education and increased the likelihood of them using ER with their pupils. This study illustrated how a teacher education workshop could improve practising teachers' beliefs about using robotics. However, this study can be criticised for not exclusively assessing the beliefs of primary school teachers. Instead, the authors included both primary and secondary school teachers in their analyses without differentiating between the two groups. This limits the generalisability of their conclusions as primary education has unique characteristics that can shape teachers' beliefs and opinions differently from those of secondary school teachers. Factors such as teacher education, class size and teaching responsibilities play a crucial role in this distinction and can have a large impact on teaching experiences and internal beliefs (Bers, 2010; Bers & Portsmore, 2005; Gutsky, 1981).

For instance, primary school teachers are responsible for teaching the entire curriculum, everything from mathematics to literacy, music to art. Consequently, their pre-service training focuses on providing them with subject knowledge that is more generalised. Even after they have obtained an in-service teaching position, primary teachers are offered fewer opportunities for STEM learning than secondary and higher education teachers (Bers, 2010; Bers & Portsmore, 2005; Sullivan & Moriarty, 2009). Thus, primary school teachers face unique challenges compared to their secondary school counterparts when it comes to teaching programming and other computational concepts. Research that fails to distinguish between the beliefs of teachers within different levels of education may oversimplify the complexities of educational settings, limiting the practicality and impact of the findings.

This limitation emphasises the importance of research that specifically investigates the beliefs and experiences of primary school teachers. The findings of which would support the development of tailored strategies for overcoming challenges in these education settings. By precisely targeting teaching challenges, the adoption and spread of innovation in education may occur more quickly and reliably (Rogers et al., 2019). Consequently, the goals of new global curriculums (e.g., the new curriculum for Wales) can be achieved through a more effective implementation of practices.

Current Thesis

Within this chapter, I described recent changes made to primary school curriculums internationally, with a particular focus on the Welsh context. Central changes to the new curriculum for Wales include a new emphasis on '*digital competence.'* Consequently, curriculum guidance highlights computational thinking and programming skills as key areas of learning. I went on to explore theoretical approaches and methods previously used to teach these concepts to children in early years education (i.e., under the age of eight). These included screen-based programs, unplugged learning approaches, and educational robotics. I presented evidence that educational robotics can be used to develop children's computational thinking and programming abilities in early childhood education.

However, questions remain about *how* educational robotics can be integrated within early primary school classrooms to develop children's computational thinking and programming skills. Having an in-depth understanding of teachers' beliefs regarding programming, robotics and computational thinking is a vital step in figuring out how to get these devices into classroom practices more effectively. If teachers do not feel confident teaching these concepts, or feel these topics are not important for young children, they will likely neglect this area of learning. Research has yet to target robotics integration from the perspective of the pupil *and* the teacher. In Chapter 2, I describe a focus group study that employed qualitative methods to speak directly to practising primary school teachers. This study provided insight into teachers' current approaches to programming and robotics education. Here, open conversations also probed teachers' beliefs about teaching in this area to uncover details about the digital education landscape in Wales from the teachers' perspectives. In Chapter 3, I build upon the findings of the focus group with a mixed-methods survey. This study continued to further investigate the methods used to teach programming and CT in early primary school classrooms across Wales and explored teacher beliefs in a much larger sample than the one used in the focus group. This larger sample provided insights from teachers from more diverse backgrounds and with a wider range of teaching experiences.

Robotics research has yet to use insights from practising primary school teachers to inform intervention design, even though co-produced research is likely to be more impactful (Oliver et al., 2019). For instance, innovative research designs and ideas for feasible ways to collect data can be identified through collaborations with stakeholders (Bowen & Martens, 2005). Thus, the findings from Chapters 2 and 3 were used to inform intervention research described in Chapters 5 and 6. But first, in Chapter 4, I present a laboratory study that investigated how best to introduce children to Cubetto (the educational robot designed by Primo Toys, used, and referenced throughout the projects in this thesis). This study explored how children (aged 4 to 7 years) could learn with Cubetto, testing whether children's spatial skills could impact programming performance. Finally, this laboratory study piloted programming tasks and computational thinking measures to be used in subsequent intervention research (Chapter 5).

In Chapters 5 and 6, I present a 6-week school-based robotics intervention that utilised the findings from Chapters 2, 3 and 4. This study employed quantitative methods, a control group, and a longitudinal design, to assess the impact of a robotics curriculum on teachers and pupils across South Wales. Data was collected from 430 children (aged 4 to 7 years) and 15 classroom teachers at multiple time points. Schools were assigned to one of three conditions: (1) Intervention+ (pupils completed the robotics curriculum, teachers attended a teacher education workshop before intervention); (2) Intervention (pupils completed robotics curriculum, teachers did *not* receive additional training); (3) Control group (no robotics curriculum, no teacher training). Teaching approaches and computational thinking assessments for pupils were guided by the findings of my previous laboratory study (Chapter 4). Furthermore, the design of the teacher education workshop was informed by my previous focus group and survey findings (Chapters 2 and 3).

Chapter 5 introduces the intervention study, focusing on pupil outcomes and explores whether using educational robotics in the classroom can benefit children's computational thinking skills and attitudes towards programming and robotics. In Chapter 6, I investigate whether teaching with robotics in the classroom positively impacts teacher beliefs (e.g., value and self-efficacy). Additionally, I explore the effects of teacher education on teacher beliefs. Finally, Chapter 6 combines the data collected from pupils and their teachers to explore whether teacher beliefs impacted pupil achievement throughout the intervention.

Overall, the studies in this thesis aimed to approach the integration of educational robotics (specifically Cubetto) in primary school classrooms, from both pupil and teacher perspectives. By combining these perspectives, I identify and make recommendations for ensuring the optimal delivery of programming and robotics education as teachers move forward with the new curriculum for Wales. As initiatives to integrate programming and robotics within primary education are present across the world, the research projects presented in this thesis also provide insights that are beneficial outside of a Welsh context.

Chapter 2. Exploring Teachers' Beliefs about Programming Education with a Focus Group.

Introduction

I start this Chapter by summarising recent changes to the Welsh primary school curriculum, including its focus on computational thinking, programming, and robotics. I then continue to describe the role of the teacher when it comes to children meeting these education goals, exploring how teacher beliefs may hinder teaching and learning in this area. Thus, I argue the importance of research that investigates teachers' beliefs regarding programming education. The study presented in this Chapter did just that and investigated beliefs held by Welsh primary school teachers. It also explored the methods and approaches used by teachers to deliver programming content in early education classrooms. This was done using a focus group and qualitative analysis. This study was the first of two studies investigating teachers' beliefs and experiences of programming education. Thus, this study was designed so that the results and insights gathered here were used to inform the design of the survey study presented in Chapter 3.

Changes to Primary School Curriculums

As explored in Chapter 1, changes within industry (i.e., the increased need for digitally skilled workers) have influenced the addition of programming to primary school curriculums across the world (Balanskat & Engelhardt, 2015; Bers 2020; European Schoolnet, 2015; Uzunboylu et al., 2017). These changes are mirrored in the recent changes to the Welsh curriculum as the New Curriculum for Wales (introduced in 2022) emphasises *science and technology* as a key area of learning. Consequently, curriculum guidance highlights computational thinking (CT) and programming as important skills even in early primary education (Hwb, 2018; Hwb, 2024a). CT has been defined for teachers as *"a combination of scientific enquiry, problem solving and thinking skills"* (Hwb, 2018). In line with this definition, scientific literature and Welsh curriculum guidance frequently highlight sequencing, debugging and prediction skills as CT skills to target within early education (Bers, 2020; Hwb, 2024b; see Chapter 1 for more detail). Sequencing has been defined as a series of individual steps or instructions ordered to achieve a desired outcome (Brennan & Resnick, 2012). Debugging refers to identifying and fixing errors in an algorithm (Bers et al., 2019). Regarding prediction skills, curriculum guidelines suggest that children should learn to *"follow algorithms to determine their purpose and predict outcomes"* (Progression step 2; Hwb, 2024b). For example, a child may use the knowledge they have about the function of different programming instructions to anticipate what an algorithm will do.

Educational robotics (ER) encompass physical programmable robots that can be used to engage children in hands-on learning activities. In the previous chapter, I explored research that highlighted the benefits of using ER to engage children under the age of 8 years in programming and CT activities (Bennie et al., 2015; Sullivan et al., 2017; Veenman & Spaans, 2005). For example, Kazakoff et al., (2013) demonstrated how ER could improve children's (aged 4 to 6) sequencing skills following a 1-week intervention. Additionally, Pugnali et al., (2017) found larger improvements in children's (aged 4 to 7) debugging abilities once they had completed a ~15-hour programming curriculum with a robot, in comparison to children who completed the curriculum on a screen-based programming language. These studies illustrate how these devices can be used to develop children's CT abilities and can be used to target learning objectives highlighted by Welsh curriculum guidance (i.e., the Digital Competence Framework and the New Curriculum for Wales).

Exploring the Role of Teacher Beliefs

Although past research has evidenced the benefits of using ER with children and suggests that ER can be used to meet the needs of school curriculums (see Chapter 1), it cannot be assumed that teachers' beliefs about programming and robotics education are aligned with those held by researchers and policy makers, who are familiar with the emerging research evidence (Owen et al., 2022; Willingham & Daniel, 2021). Thus, researchers must work with primary school teachers to understand their beliefs in this area.

Research has shown that negative teacher beliefs can hinder the integration of new digital concepts (i.e., programming) and their corresponding technologies (i.e., robotics) within the classroom. Through classroom observations and interviews with teachers, Larke (2019) found that teachers would neglect teaching mandatory computing education if they believed it difficult to teach. Furthermore, teachers' internal beliefs have previously been evidenced as a barrier to technology integration (Ertmer et al., 2012). Hew and Brush (2007) analysed 48 empirical studies and highlighted the three most frequently cited barriers impacting technology integration in classrooms. These included (1) teachers' beliefs, (2) teachers' knowledge and skills and (3) resources. These authors highlighted several studies whose findings suggested that negative value beliefs about the relevance of using technology within the classroom may prevent the integration of technology within lessons (Ertmer, 2005; Ertmer et al., 1999;

Newhouse, 2001). Findings like these illustrate how teacher beliefs could be a barrier to programming and robotics education, thus illustrating the importance of research that investigates teacher beliefs.

Typically, researchers characterise teachers' beliefs in two ways: (1) value beliefs and (2) teacher self-efficacy beliefs. Firstly, value beliefs encompass the perceived importance of specified goals and choices (Anderson & Maninger, 2007). For example, in the context of this research, teachers' value beliefs may include how important or relevant they believe programming and robotics education is for children in early primary education. On the other hand, teacher self-efficacy beliefs refer to teachers' confidence in their ability to bring about desirable changes in students' behaviours, motivation, and achievement (Tschannen-Moran & Hoy, 2001). Self-efficacy beliefs here may also include how confident teachers feel using ER and teaching programming concepts to young children.

As school curriculums evolve and new concepts are introduced, it is possible that new aims for children's learning may not align with teachers' past experiences and beliefs, as research from Liao et al., (2017) suggests that teacher education opportunities are not sufficiently supporting teachers' knowledge of emerging technologies and their use within teaching practices. As a result, many struggle to integrate and teach with new technologies and approaches. Such experiences may result in low self-efficacy as teachers do not know how to use new technologies properly. Additionally, teachers may not see the value in new technologies like ER if no one has explained how they can benefit children's learning and they have no experience using robotics themselves. Given that teacher beliefs may negatively impact chosen pedagogies and teaching practices, research must investigate teacher beliefs, and this is likely to be especially important in newer, more unfamiliar topic areas like programming and robotics.

Research findings have also suggested that teacher beliefs can impact pupil outcomes. For instance, meta-analyses have evidenced significant effects of teacher self-efficacy on pupil achievement (Kim & Seo, 2018; Klassen & Tze, 2014). Researchers have theorised that teacher self-efficacy may positively impact pupil achievement via improved teaching behaviours and classroom practices (Lauermann & Butler, 2021; Mok & Moore, 2019), or via role-modelling processes, whereby increased teacher self-efficacy may be transferred to pupils which results in increased pupil persistence which then benefits pupil achievement (Lauermann & ten Hagen, 2021). Such research findings further highlight the importance of research that investigates teacher beliefs given that they could impact pupil achievement.

Research investigating teacher beliefs in Wales would be particularly valuable given the changing educational landscape and the increasing focus on digital skills. The Welsh Government believe that CT and programming are valuable skills for children to start developing in early primary education, but do teachers believe the same? Do they feel confident teaching children these skills? Or are teachers in Wales avoiding programming and robotics education within their teaching practices due to negative value and self-efficacy beliefs? It is important that researchers and policy makers understand the current situation, so that if teacher beliefs are preventing learning in these areas, then it can be addressed. Thus, the study presented in this Chapter was the first in a series of studies that ultimately aimed to promote programming and robotics education in early primary school classrooms. The insights gathered from this study (e.g., the identification of teaching barriers) were later used to aid the design of a school-based robotics intervention that was delivered to primary school children by their classroom teachers (to be presented in Chapters 5 and 6).

Teacher Beliefs about Barriers to Programming Education

It is important to acknowledge that teacher beliefs are unlikely to be the only factor limiting the integration of ER and programming education. Studies have investigated the barriers teachers may face as they attempt to teach programming with ER. Research from Ertmer (1999) explores the relations between first- and second- order barriers to technology integration. First order barriers are referred to as external factors such as limited equipment, training, and teaching support. Meanwhile, second order barriers relate to teachers' own beliefs about things like curriculum priorities and assessment practices. Ertmer highlighted that though teachers are unlikely to experience the same barriers in the same ways, all teachers are likely to experience both types of barriers (first and second order) as they attempt to integrate technology within their classrooms.

This conclusion is reflected in the findings of more recent research. A review paper from Mason and Rich (2019) found that barriers, including a lack of technology resources or reliable internet access, institutional obstacles in the form of unsupportive headteachers, and teachers' beliefs can inhibit the use of technology within the classroom. Furthermore, Dralle-Moreano (2021) used qualitative methods (including interviews and focus groups) to investigate how those teaching children in the US (aged 4 to 14) described the barriers they experienced when teaching programming to their pupils. They summarised that teachers most commonly mentioned a lack of teacher education and their pupil's abilities (i.e., reading abilities) as barriers. When asked to think about how these barriers could be overcome, the teachers highlighted themes including the need for teacher education, the idea that collaboration amongst teachers is important, and a broader need for teachers to understand the value of teaching programming. These findings from Mason and Rich (2019) and Dralle-Moreano (2021) supported Ertmer's suggestion that teachers experience both first- and second-order barriers within technology related teaching. It is evident that by working with teachers to examine their beliefs, researchers can also gain insight into the additional barriers they face whilst teaching, that are external to the teacher themselves.

However, one limitation of these studies is that neither study assessed the beliefs of primary school teachers exclusively. Instead, Mason and Rich (2019) reviewed some studies that employed samples of both primary and secondary school teachers. Similarly, Dralle-Moreano (2021) also interviewed those working in secondary education teaching children up to the age of 14 years. Consequently, neither study was able to make any distinctions between primary and secondary teaching experiences when drawing their conclusions. This is problematic as several characteristics of primary education make it probable that primary school teachers' beliefs and opinions will differ from those held by secondary school teachers. In fact, findings from Sentance and Csizmadia (2017) illustrate belief differences between primary and secondary school teachers when it comes to computing education. They surveyed 339 teachers (77% secondary school educators, 16% primary educators) and found differences between secondary and primary teachers in their responses about difficulties surrounding teaching computing. For example, more primary teachers (40%) mentioned lack of knowledge being a difficulty than secondary teachers (26%). This may be because primary school teachers are responsible for teaching several subjects, while secondary school teachers focus on fewer subjects (Greifenstein et al., 2021). Consequently, primary school teachers' pre-service training focuses on providing them with subject knowledge that is more generalised. Furthermore, once primary school teachers are qualified, they may be offered fewer teacher education opportunities for STEM learning than secondary and higher education teachers (Bers, 2010; Bers & Portsmore, 2005; Sullivan & Moriarty, 2009). Additionally, the literature suggests that additional variables such as class size, and teaching responsibilities can have a large impact on teaching experiences and internal beliefs (Bers, 2010; Bers & Portsmore, 2005; Gusky, 1981). Thus, it can be argued that primary school teachers face unique challenges compared to their secondary school counterparts when it comes to teaching programming and other computational concepts.

Current Study

To further understand teachers' teaching practices and to explore what might limit their engagement with programming and educational robotics education, I aimed to (1) explore primary school teachers' methods of teaching these topics, (2) further explore teachers' beliefs about programming and robotics, and (3) to identify routes to overcoming any barriers to programming education identified by teachers. These insights will likely prove particularly helpful as changes are made to the educational landscape in Wales. The New Curriculum guidelines were published in January 2020 (Hwb, 2024c); however, implementation was delayed until 2022 due to the COVID-19 pandemic. Thus, this focus group study was conducted in May 2021, at a time when teachers were considering how to reconcile curriculum changes and new learning targets with their personal values, beliefs, and past experiences.

Investigating teachers' beliefs in this area may provide important contributions for those who develop the curriculum and other researchers. For me, speaking directly to teachers provided valuable insights that I later used to aid the design of a school-based intervention that aimed to investigate the benefits of using robotics in early primary school classrooms. This study provided an opportunity to explore what teachers were already doing to teach programming, what they believed worked well and what improvements were needed. However, this study went beyond just identifying a list of beliefs but also aimed to explore how teachers and their schools attempted to overcome barriers to teaching programming and robotics content effectively. As a result, this study made recommendations for how schools may overcome identified barriers when implementing educational robotics in the classroom. These recommendations emerged following conversations with practising primary school teachers and explorations of the literature and these insights were integrated into my later research studies (see Chapters 5 and 6).

Focus group discussions with primary school teachers and qualitative analysis methods were used to answer the following research questions:

1) What beliefs do teachers hold about programming and robotics education?
2) How do teachers teach programming education?
3) What do teachers identify as barriers to programming and robotics education?
4) What strategies can help overcome barriers to programming education?

Methods

Justification of the Method

Qualitative focus groups were chosen as this study is phase one in understanding teachers' beliefs and experiences of programming education. In the next chapter (Chapter 3), I will describe phase two, which involved the distribution of a mixed-methods online survey to teachers across Wales. The themes explored and the questions presented in this survey were informed by the results from the current focus group.

The focus group was conducted virtually using Zoom video calling software. This video-conference platform was chosen as it allowed discussions to take place in real-time (synchronously), enabling conversation to flow back and forth between researchers and participants (Fox et al., 2007). Another benefit of conducting the focus group session online was that neither the researcher nor the participants were required to travel during a time when COVID-19 pandemic guidelines restricted group gatherings outside of household groups. This was also advantageous for saving time, costs (Kenney, 2005) and improved the geographic reach of the research, allowing teachers from both urban and rural areas of Wales to participate (Namey et al., 2020).

Participants and Recruitment

Thirteen teachers took part in the focus group interview (falling just outside of the typical focus group range of 6 to 12 individuals, Namey et al., 2020). These teachers were located across Wales and taught in Foundation Phase classrooms (ages 4 to 7 years). Participant demographic information is shown in Table 2.1.

Table 2.1

Demographic Information for Participating Teachers.

Variable	Total (n = 13)	
	n	%
Gender		
Female	12	92.31
Male	1	7.69
Age		
25 - 34	7	53.85
35 - 44	2	15.39
45 - 54		
55 - 64	1	7.69
Not provided	3	23.08
Experience		
Pre-service teacher	1	7.69
1 – 5 years	2	15.39
6 – 10 years	4	30.77
11 – 15 years	2	15.39
16 – 20 years		
21 – 25 years	1	7.69
25 years +	1	7.69
Not provided	2	15.38
School type		
Government funded	13	100
Year group taught *		
Reception	4	30.77
Year 1	13	100
Year 2	12	92.31
Key Stage 2	5	38.46

*Note: * Most teachers had experience teaching more than one year group in early primary school years, so percentages add up to more than 100%. Key Stage 2 = children aged 8 to 11.*

Teachers were recruited via Techniquest (a third-party collaborator). Techniquest is a children's science and education centre based in Cardiff, Wales. This study was advertised directly to teachers by members of the Techniquest team. Using their internal database, they were able to contact primary schools across Wales and provide them with details about the study. The published advert (see Appendix A) explained that this study was aiming to explore teachers' thoughts on using robotics within the classroom. Teachers were informed that experience with programming or robotics was not required for them to attend the session. Additionally, it was explicitly stated that input from inexperienced teachers would be just as valuable as input from teachers who had experience teaching with robotics.

Prior to their participation, teachers received information regarding the nature of the study and provided written consent. They were gifted a copy of the book "Ada Twist, Scientist" (written by Andrea Beaty) as compensation for their participation. This research was approved by the Cardiff University Psychology Ethics Committee (approval number EC.21.03.09.6327G).

Procedure

The focus group session ran for 2.5 hours (duration typical of focus group research, Namey et al., 2020). These informal discussions were video and voice recorded for transcription after the fact. Both breakout rooms and an interactive Mentimeter activity were used on occasion to encourage conversation. The use of breakout rooms, despite the potential risk of missing important qualitative insights, was intentionally chosen to enhance participant engagement by creating a more comfortable and focused environment for discussion. This approach also aimed to reduce the risks associated with potential withdrawal from the conversation, as smaller group settings can encourage more active participation and allow individuals to express their thoughts more freely without the pressure of a larger audience. The structure of the meeting was designed in line with the research questions proposed for this study. The final focus group schedule was broken down into five short sessions which are described in detail below. Each of these sessions were delivered to teachers by me (as the Primary Investigator), with support from a secondary researcher.

Session One

I opened the session by covering initial introductions, both teacher and researcher, and by delivering a short PowerPoint presentation. This presentation described the rationale for this study and clearly placed the study within the context of Welsh primary education. For example, teachers heard about the recent push for programming education in early childhood classrooms (in line with the Digital Competence Framework, 2018). Using Welsh government resources as guidance (Hwb, 2018; Welsh Government, 2017), I described computer programming and listed possible tools or methods that could be used to teach programming to young children (i.e., screen-based programs and tangible technologies like educational robotics). I closed the first session by highlighting the importance of teachers' opinions and the role of the teacher with regard to technology integration within the classroom.

Session Two

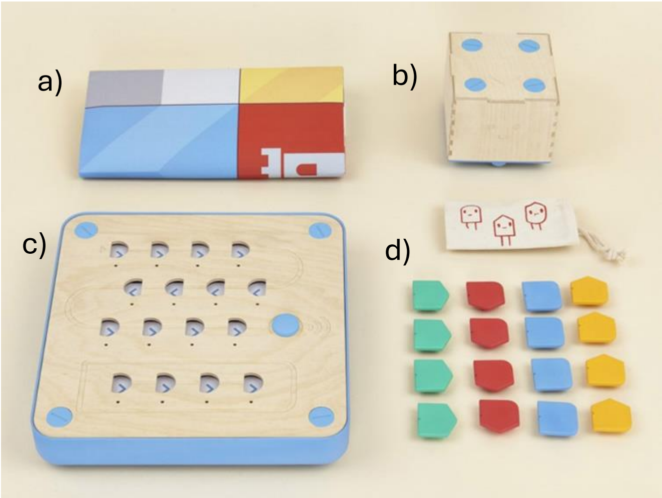
This session involved discussions that centred around the current computing curriculum and prompted teachers to consider their thoughts about computer programming specifically. Teachers were placed into breakout rooms (approximately 4 per group) to discuss what they currently did for computing and how they felt about teaching the subject. Additionally, they were asked to share their experiences with computer programming (if they had any). Breakout rooms were not recorded nor were they attended by either of the researchers. Following these 10-minute breakout discussions, the group reconvened in the main "room" and each group shared their experiences and opinions.

Session Three

This session began with another presentation delivered to the whole group. This presentation introduced Cubetto (www.PrimoToys.com), a wooden robot developed by Primo Toys which my later research focused on (see Chapters 4, 5 and 6). Cubetto playsets come equipped with an interface board, a range of function tokens (forward, right, and left turn functions) and a colourful floor map (see Figure 2.1). Children can navigate the robot around a map by placing desired tokens in the interface board and pressing the 'Go' button. Video demonstrations were included to illustrate how Cubetto could be operated by children. Following this presentation, teachers were once again placed into breakout rooms to discuss their thoughts about using robotics generally within their classrooms and any previous experiences they may have had with programmable robotics. These breakout sessions again lasted approximately 10 minutes, were not recorded, and were not attended by either of the researchers. Instead, teachers shared their thoughts and discussions with the wider group once the breakout discussions had ended.

Figure 2.1

Cubetto Playset.



Note: Each set comes with a floor map (a), a Cubetto robot (b), an interface board (c) and a range of movement tokens (d).

Session Four

Teachers were prompted to discuss their experiences of teacher education programs and possible classroom barriers to robotics integration. Teachers were questioned about their previous education opportunities and their willingness to attend programs that focus on learning about programming and using robotics in the classroom. In the second half of this session, teachers were encouraged to think about possible classroom barriers that may prevent them from introducing robotics to their pupils. To facilitate discussion, an interactive Mentimeter word cloud portal was used. Teachers were able to view the interactive poll on their devices and were presented with the question *"What barriers or challenges are there when it comes to using robotics in the classroom?"* Teachers were then able to anonymously provide as many answers as they liked, and their answers appeared on the screen for the rest of the group to see. This exercise was used to stimulate discussions rather than to collect analysable data. As a group, participants were then able to discuss the results, identifying similarities, differences, and most common answers.

Session Five

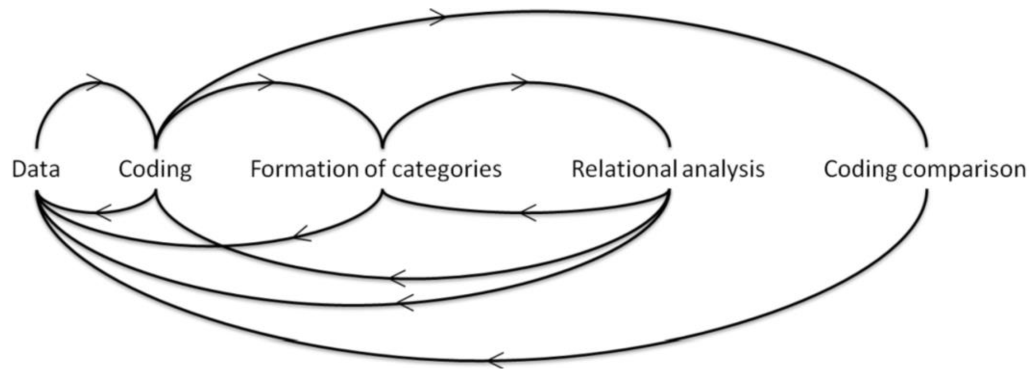
This final session centred around the participants' general teaching practices and prompted discussions about what would make a good classroom curriculum. Teachers discussed lesson structure and topic timelines within the classroom. They were also asked to think about how often they are likely to use new technologies with their pupils moving forward. Additionally, teachers were prompted to discuss how a robotics curriculum could be designed so that it best catered to their needs and made integration as straightforward as possible. Finally, the session closed with a final discussion about the new Welsh curriculum due to enter practice in 2022.

Data Analysis

The focus group interview was transcribed and open-coded using the software package *'NVivo'*. Due to the exploratory nature of this study, the constant comparative method was used to code these quotations (Glaser 1965, Merriam, 1998). This is an interactive, sometimes termed 'messy' method of data analysis in which data is constantly compared to identify similarities, differences, and patterns. During the first readings of the transcript, all quotations of interest were highlighted. To begin, quotes were selected if they appeared to offer insight into teachers' beliefs about programming and robotics education, or the methods they had previously used to teach these topics. During subsequent readings, each highlighted quote was analysed and given initial codes. These codes were compared and revised between multiple readings of the transcript (see Figure 2.2, Girvan et al., 2013). This helped to minimise the occurrence of redundant codes. The next step in the constant comparative process involved tentatively integrating codes into sub-categories and then broader higher-level categories. These initial categories were presented to three other researchers for discussion. In this discussion, we pulled apart each category and sub-category to explore each code to explore how codes could be organised in different ways. These discussions also raised the need for inter-rater coding.

Figure 2.2

The Constant Comparative Method (Girvan et al., 2013).



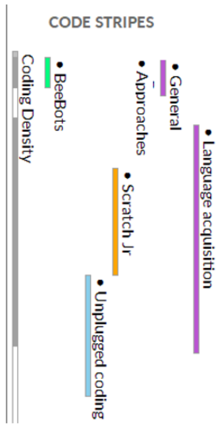
Inter-coder reliability has previously been applied in other forms of qualitative analysis (e.g., content analysis), during which multiple researchers code, clarify and recode data until a specific level of agreement is reached (Neuendorf, 2002). Cohen's Kappa is commonly used to calculate this level of agreement (Foster et al., 2008). Due to the nature of constant comparative analysis, inter-coder reliability presents itself differently. This is because codes are allowed to emerge from the data rather than being selected from the literature prior to the study (Strauss & Corbin, 1998). Thus, reliability checks were conducted by inviting a second coder to openly code a portion of the focus group transcript. Having done an initial round of coding myself (see Figure 2.3a), a second coder openly coded approximately 25% of the transcript (see Figure 2.3b). This was followed by a meeting of coders to discuss areas of divergence in coding. After these discussions, I then completed another round of coding myself. This allowed me to further revise my codes following discussions with the second researcher (see Figure 2.3c). The screenshots copied below show how quotations of interest became more focused and generated codes became more refined following discussions with a second researcher. For example, it quickly became clear that codes identifying mentions of "Bee-bot" or "Scratch Jr" were too broad to provide any detailed insights. Thus, through multiple rounds of coding, more detailed codes were identified which provided specific insights into teachers' experiences with these teaching methods. Table 2.2 contains the final codes identified and their corresponding sub-categories and broader categories.

Figure 2.3

Examples of Transcript Coding.

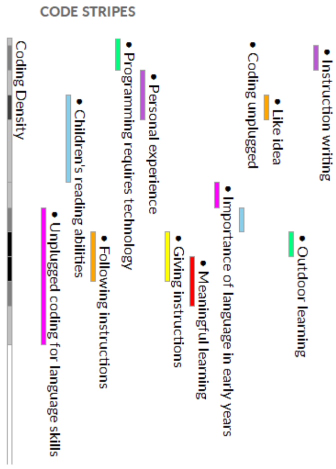
a) **SC:** We talked about being coding unplugged mainly. Because it does seem everyone is used to thinking it's based more into technology with an iPad with Scratch. We talked about the use of Bee-Bot for instruction writing. I've done bits with Micro:Bit but that was a course I managed to get onto with the BBC with my son. But I quite like the idea of scratch Jr because it's, like you said before, children from reception, nursery age, they can't always read. I mean, even in year 2 we've got children who wouldn't be able to read it. Erm, so we were talking about that.

PJ: I think we were saying about how important the language is, in the early years, in year one and year two really to do the unplugged activities because they need to go outdoors and follow instructions, um, play in construction and follow instructions building a model and telling their partner to do the same. Making meaningful, um, situations for them to do those activities. Um, and just basically get the language. It's really important.



b) **SC:** We talked about being coding unplugged mainly. Because it does seem everyone is used to thinking it's based more into technology with an iPad with Scratch. We talked about the use of Bee-Bot for instruction writing. I've done bits with Micro:Bit but that was a course I managed to get onto with the BBC with my son. But I quite like the idea of scratch Jr because it's, like you said before, children from reception, nursery age, they can't always read. I mean, even in year 2 we've got children who wouldn't be able to read it. Erm, so we were talking about that.

PJ: I think we were saying about how important the language is, in the early years, in year one and year two really to do the unplugged activities because they need to go outdoors and follow instructions, um, play in construction and follow instructions building a model and telling their partner to do the same. Making meaningful, um, situations for them to do those activities. Um, and just basically get the language. It's really important.



c) **SC:** We talked about being coding unplugged mainly. Because it does seem everyone is used to thinking it's based more into technology with an iPad with Scratch. We talked about the use of Bee-Bot for instruction writing. I've done bits with Micro:Bit but that was a course I managed to get onto with the BBC with my son. But I quite like the idea of scratch Jr because it's, like you said before, children from reception, nursery age, they can't always read. I mean, even in year 2 we've got children who wouldn't be able to read it. Erm, so we were talking about that.

PJ: I think we were saying about how important the language is, in the early years, in year one and year two really to do the unplugged activities because they need to go outdoors and follow instructions, um, play in construction and follow instructions building a model and telling their partner to do the same. Making meaningful, um, situations for them to do those activities. Um, and just basically get the language. It's really important.

Note: Screenshots from NVivo illustrating the development of codes and how codes were refined following multiple rounds of coding. Image a) is a snapshot from the first round of coding, image b) shows coding completed by a second researcher and image c) shows another round of coding completed by researcher one following agreement discussions between the two researchers.

Table 2.2

The Final Categories, Sub-Categories and Codes Identified During Analysis of the Focus Group Transcript.

Category	Subcategory	Codes
Perceptions of programming education		"vitally important" "based more into technology"
Barriers to programming education.	Technology factors	Lack of program visualisation Lack of versatility Screen time concerns Longevity Durability Cost
	Pupil factors	Reading abilities Current technology skills Improving tech skills "upskilled themselves"
	External factors	Lack of time High workload Timetable pressures Changes to teaching priorities Infrequent access to technology Space Limitations of teacher education Generic teaching technologies KS2 or higher content
	Teacher factors	Limited knowledge Confidence
Pedagogic approaches	Pupil-led instruction Unplugged learning	Instruction writing Outdoor learning

		Online resources Barefoot coding
Forming support networks	Just-in-time supports Limitations	Develop own knowledge Teachers supporting teachers "expert groups" Drop in sessions "COVID bubbles"
Tailored teacher education programs	Course content Structure and approach	Interdisciplinary teaching Avoid "shoehorning" Meaningful learning Level of support

Results

In this section, I explore the categories, sub-categories and codes identified during the analysis of this focus group session. Below, the five separate categories identified during the analysis are presented as headings. The sub-categories within each of those top categories are presented as subheadings. Individual codes are highlighted within the main body of text in bold. Quotations appear in quotation marks ("") and are italicised to be easily distinguished from the text. Any text that appears in quotation marks and bold lettering is a direct quote that has been recorded as a code.

Following the discussions that took place in this focus group study, this results section begins by describing teachers' general perceptions of programming education and their experiences of barriers that make teaching the subject difficult. This is then followed by an examination of coping strategies teachers and schools have in place to overcome some of these barriers (i.e., adaptations to pedagogy and the formation of internal support networks). Finally, the data is explored to gather recommendations for improving teacher education courses.

Perceptions of Programming Education

Overall, the group held positive beliefs about the role of programming education within primary school, with several teachers sharing the opinion that introducing children to this subject was ***"vitally important"*** and that *"the earlier they start, the better."* Interestingly, when it comes to delivering programming lessons, *"it does seem everyone is used to thinking it's*

based more into technology. *"* Approaches to teaching programming will be covered in more detail in the sections below.

Barriers to Programming Education

Guided by researcher questions, teachers discussed what they believed to be barriers to robotics integration and programming instruction. The barriers identified were sub-categorised as (1) technology factors, (2) pupil factors, (3) external factors or (4) teacher factors.

Technology Factors

Teacher discussions highlighted two main technologies used in the classroom to teach programming to young children: Bee-bot (a tangible robot) and Scratch Jr (a screen-based program). Children can operate Bee-bot robots (www.tts-group.co.uk) by pressing buttons on top of the device. These include move forward, move backwards, turn left, turn right, pause, and reset. On the other hand, Scratch Jr is a programming language which enables children aged 5 to 7 years to create interactive stories and games by connecting graphical programming blocks (Bers, 2018).

Multiple teachers appeared to have hands-on experience with Bee-bot robots and highlighted several limitations of this technology. For example, one teacher voiced that *"with the more complex ones, like Bee-bot I guess, some teachers do feel it's not suitable for the younger ones. The 3's and 4-year-olds."* Several teachers later emphasised that the robot's **lack of program visualisation** is something young children find challenging. One teacher explained, "*a lot of children struggle with Bee-bot with remembering to clear it, not being able to see the actual directions and route you've taken down."* Another teacher shared similar experiences with their pupils, *"when they've popped things into the Bee-bot they're not really sure what it's going to do or where it's going to go."* When prompted to consider the advantages of using a robot or programming language with a visual for algorithms, teachers liked the idea. One teacher mused that this would mean children could *"see where they've gone wrong"* which could *"help with the debugging and predicting."*

Teachers who had not used educational robotics in their teaching practices also contributed to the discussion. One teacher was concerned about the **versatility** of robotics. In their comments, they compared robotics to screen-based technologies,

> "*It's not really visual like, with a computer and the internet you can always keep up with the times. So, if you had an iPad you could change, it's very changeable and you*

could download an app, when they get bored of that app you could move onto the next. It's the next best thing".

Those who had not used robotics in their classroom voiced their concerns about the **longevity** of educational robotics in relation to the **cost** of the devices. One group were concerned about the **durability** of robot toys, and the practicalities of keeping track of all the components that come with the playsets. One teacher shared,

> *"We worry that nursery children might pick it up and think it was a toy of some sort and put it in a different box. For an expensive piece of kit, the different pieces could end up in different storage containers and things."*

Further discussion also implied that some teachers struggled to see the value of educational robotics when thinking about costings and budgets. One teacher questioned,

> *"Would it be value for money? The use of it. Then you'd have to think about the add-ons and things. If we did change year groups, would it just be stuck in the cupboard somewhere? Then we'll look back and think we could have spent our money elsewhere."*

In contrast to the Bee-bot robot, fewer teachers had personal experience teaching with Scratch Jr, but one teacher that had used it highlighted the app's versatility both inside and outside of the classroom, *"in one of the lockdowns we did set Scratch Jr as a task because, what I didn't realise, is that it wasn't just an app, you can actually log in on desktop computers as well."*

Several teachers shared that since returning to the classroom after the COVID-19 lockdowns, they have not utilised Scratch Jr as frequently. This appears to be due to some teachers' concerns about their pupil's **screen time** following the pandemic. Changes in learning during COVID-19 lockdowns led some schools to focus on teaching without the use of screens, which is likely to limit teaching with Scratch Jr. One teacher explained,

> *"I know from our point of view as a school, we've had quite a big push on NOT having children in front of screens. (…) I know we had a lot of worries when we came back that our children have just spent the last year, in one way or another, staring at screens. So, we think we've gone a bit backwards in that respect just because we literally haven't wanted to put computers in front of them."*

Pupil Factors

Teachers acknowledged that children's **reading abilities** may restrict their learning of programming concepts. However, most were aware of different teaching methods that could minimise this as a barrier to learning. For example, one teacher liked *"the idea of Scratch Jr because children from reception, nursery age, they can't always read."* This same teacher went on to highlight that reading proficiency can still be a challenge with older children, "*I mean, even in year 2 we've got children who wouldn't be able to read it."*

Some teachers felt that their pupil's **technology skills** have prevented them from teaching programming. With only having access to screen-based programming applications like Scratch Jr, one teacher described the difficulties of teaching pupils who *"haven't got the skills to log in and they don't even know how to turn the computer on and things."* They emphasised that the COVID-19 lockdowns *"haven't really helped"* this problem. As a result, this teacher has been focused on **improving pupil's tech skills** and found themselves *"working on basic skills,"* rather than introducing more complex digital skills.

However, it is important to note that pupil technology skills are not a concern for all teachers. In contrast, other teachers had witnessed their pupils **upskill themselves** during the pandemic. Whilst learning from home pupils were often *"responding to work, (...) responding to videos, responding to images, blogging."* As a result, their pupils had gained experience with different types of technology and their teacher felt that introducing programming in the near future was *"more than manageable."*

External Factors

A **lack of time** and **high workload** means that teachers cannot individually research and familiarise themselves with new knowledge and technologies. As a result, both teachers and pupils are missing out on new skills and exciting pieces of educational technology, *"I haven't personally had the, I hate to say it, the time to sit down and work out how to use Scratch Jr."*

Additionally, teachers discussed how COVID-19 disruption has resulted in additional **timetable pressures** and has caused **changes in teaching priorities**. While some teachers may *"have done a little bit"* of programming, to catch up on learning missed during pandemic lockdowns, their *"priority has been literacy and numeracy."* Other teachers shared that *"ICT is often pushed back"* as there is *"just so much pressure to fit everything in."* These quotes evidence that teachers struggle to find the time to teach technological concepts like programming in the classroom and that this has been exacerbated by the pandemic. Since returning to the classroom, teachers have prioritised other classroom subjects, and as a result, have not been using new technologies or delivering programming education within the classroom frequently. These accounts also hint that teachers perceive programming education as a topic that cannot be easily integrated with other classroom subjects. Instead, programming and robotics lessons are *"pushed aside"* to ensure other content is covered.

Classroom access to technology varied greatly within the group. One teacher shared that their classroom has **infrequent access to devices**, stating that they *"just have a trolly of 15 laptops for the Welsh section, a trolly of 15 laptops for an English section"* which they *"have a slot for one day, once a week."* This appears to have a negative impact on teaching and learning as it *"is not enough for everybody to get things done."* In contrast, some teachers were very open about the fact that they taught at an *"ICT rich school"* with each class having access to *"15 Chromebooks… iPads… Apple Suites."* One teacher also explained how their school was able to provide laptops *"for the children who didn't have devices"* during the pandemic when learning was happening at home.

There may be a tentative link between the likelihood of a teacher having difficulties with their pupil's technology skills and classroom access to technology. Within the group, those that found teaching with technology difficult due to their pupil's skill set also reported infrequent access to devices in the classroom. On the other hand, those who had access to devices frequently shared that they were happy with their pupils' technology skills, even after the pandemic. After being able to provide devices during lockdown, one teacher explained the positive impact of this, *"We kind of picked up where we left off then, um, because everything we were teaching them was obviously remotely and the lessons that we were sending to them, they needed to respond to us using their device."* These accounts suggest that the level of access a teacher has to technologies within their classroom is linked to how technically skilled their pupils are.

Finally, when considering external barriers, not only do tangible robotics take up a lot of **space** in practice, but teachers also need to consider where they can store the additional equipment. For small schools, this may be a problem. One teacher who did not have experience with using robotics in the classroom expressed their concerns about storage, explaining that they were unsure where they would keep the devices. *"EVERY corner of our school is being* *used and it's a very old building, very small classrooms. You know, we've got teachers in cupboards pretty much – no joke there."*

During their discussions, teachers considered the **limitations of teacher education** courses that specifically focused on programming or robotics content. Instead, teachers had typically experienced teacher education programs that focused on more **generic teaching technologies**. One teacher explained,

> *"We've done some through our school, it's a lot of Google Educator, Apple Teacher, we've done a lot of that, but it focuses on the programs that are actually available like your Google slides or things like that, it doesn't focus on robotics or programming. There's not a lot of courses out there."*

Consequently, most participants believed they would benefit greatly from professional development courses that specialised in programming and robotics, *"Definitely in terms of robotics and training there is a great need out there."*

Of those who had attended courses in the past, they made it very clear that these programs fell short in terms of scope as the content was tailored to those teaching children above the age of 8 years old. One teacher clearly stated, *"I've always found that everything is aimed either **KS2 or higher**. It's not low enough and I think that's where it's falling down."* They indicated that, like those who had not received any training, they were still without the knowledge that would allow them to successfully deliver programming and robotics lessons to their younger pupils. Another teacher further highlighted the need for tailored education programs, "*There isn't programming courses for lower down in the school and I think there needs to be."*

Teacher Factors

Most teachers in this sample felt they lacked sufficient programming and robotics knowledge. Following small group discussion, one teacher shared *"I think all of us, if I can speak for all of us, have all said that we've got **limited knowledge** of computing."* Teachers' **confidence** in their knowledge of programming and their ability to use robotics was another barrier highlighted by the group. One teacher shared that they viewed low teacher confidence in Early Years as *"the biggest thing"* preventing robotics education. They went on to explain, *"I think everyone in Key Stage 2, and myself, tend to do all the ICT courses but the teachers in Foundation Phase aren't as confident."* It appears that teacher confidence is negatively impacted by a lack of teacher education opportunities for those teaching in early primary school classrooms.

Pedagogic Approaches

Teachers and their schools appeared to have adopted certain pedagogies based on their experiences of external, pupil, teacher, and technology related barriers. For instance, due to limitations in teacher education, which consequently impacted teacher knowledge and confidence, some schools utilised **pupil-led instruction** methods. Those teachers appeared to use programming and robotics education as an opportunity to encourage peer-supported learning, whereby "*the older children, who are much more confident in using it"* work with younger pupils directly to ensure that "*knowledge is emanated downwards."* Schools utilising this pupil-led approach reported that older children in Key Stage 2 (aged 8 to 11 years) are given more learning opportunities for programming education. One teacher shared, *"In Key Stage 2 we do things like Hour of Code, coding um at lunchtime."* These older pupils have then been encouraged to share their knowledge with younger pupils.

When choosing pedagogies that minimise pupil related barriers (i.e., children's reading abilities and technology skills) and technology related barriers (i.e., Bee-Bot's lack of program visualisation and aims of avoiding screen time), some teachers have utilised **unplugged learning** approaches when teaching programming. These unplugged methods appear to focus on **instruction writing skills**, *"We do a lot of things such as making a sandwich but having your little jigsaw pieces, so they've got to put them… it's just that kind of instructional writing."* One teacher also mentioned that unplugged methods can provide opportunities for **outdoor learning**. They appear to encourage their pupils *"to go outdoors and follow instructions, um, play in construction and follow instructions building a model and telling their partner to do the same."*

Teachers have found that certain **online resources** are particularly useful for teachers wanting to engage in unplugged programming activities. One teacher explained,

> *"We use things like **Barefoot coding** as well to teach things like algorithms before doing it on the computer. The plans are so detailed, I often give them to my TA to use and she'll do it unplugged first and then I'll teach the ICT side of it then. But it gives them the knowledge before touching the computer."*

Barefoot coding (www.barefootcomputing.org) resources appeared to be popular across the group, with this organisation providing comprehensive materials that are easy to use and are enjoyed by teachers, *"I have used a few of their resources and they're very good. Very good physical, practical resources."*

Forming Support Networks

Discussions with teachers indicated that there appears to be an expectation for them to **develop their own knowledge** of programming education and emerging technologies or teaching methods. It appears that the pandemic may have exacerbated this, *"We've been self-taught as well through Covid when we weren't, you know, with everything being online and having to go online."* Furthermore, one teacher shared the programming knowledge they had gained outside of work, through personal experiences with their own child, *"I've done bits with Micro:Bit but that was a course I managed to get onto with the BBC with my son."* However, if teachers do not have the time to pursue these courses outside of work time (as suggested above) or they lack the knowledge of where to find such opportunities, they are instead likely to seek out support from other members of staff.

Just-in-time Support

As a result, primary educators meet with colleagues to create a selection of 'just-in-time' supports. Participants gave several accounts of **teachers supporting teachers**. For instance, they shared how they have attempted to build internal support systems with other colleagues to form their own knowledge base. One teacher explained how they and their colleagues had developed their programming and technology knowledge by forming ***"expert groups"*** where each member of staff had *"areas to go and look at"* before sharing what they had learned with others.

Several participants described how they were responsible for supporting other teachers when it came to teaching programming and robotics lessons. One teacher shared how they would offer **drop-in sessions** for other members of staff if they needed support with technology and computing.

> "*We had a 15-minute slot where staff can just come into my classroom, I can show them something really quick which they can then take away with them for the rest of the week."*

This teacher further explained how such strategies can be used to support those who are less confident with technological concepts like programming and robotics. They explained, *"That's been quite successful, because again, even though some staff are really nervous, it's a short burst so they come in, they can hear what I have to say or show them something and then they can go then. So that's worked quite well as well."*

Limitations

Although these 'teachers supporting teachers' strategies are successful in some cases, it appears that they are only effective to a certain extent. Several teachers touched upon the limitations of these support strategies. One teacher shared,

"I think we've managed to develop ourselves in a way in certain aspects of ICT. Sometimes, the further you go with it, the harder it is to find the expertise to be able to support."

In agreement, another teacher said, *"it comes to a point where you don't have… there is nowhere else to go internally so you do need to look externally for expert advice."*

Moreover, in recent years there have been instances where these supports were not accessible at all. COVID-19 restrictions meant that drop-in sessions and classroom support from other staff members became a thing of the past. Multiple participants mentioned the impact of *"**COVID bubbles.**"* Classroom "bubbles" were introduced to control mixing between different classrooms and year groups to curb transmission of the virus. As a result, teachers were no longer able to provide additional support to colleagues whilst restrictions were in place. One teacher shared,

"*Because I'm Foundation lead for ICT - usually I'd go into different classrooms and introduce topics and show the teachers how to do coding and things but with bubbles we can't do that.*"

Tailored Teacher Education Programs

Discussions surrounding the limitations of teacher education courses subsequently highlighted several ways these programs could be improved.

Course Content

Teachers collectively emphasised the importance of **interdisciplinary teaching** when attempting to integrate programming and robotics lessons. One teacher illustrated the prominence of interdisciplinary learning in other areas of teaching and argued for the same approach to programming and robotics education, *"I think it's making it cross curricular as well, whether that be how you can bring it into different subjects (…) having robotics shouldn't* *be a standalone subject."* This teacher went on to suggest that teacher education programs should focus on interdisciplinary teaching as an approach*, "It should be, you know, can we use this in our maths lesson? Can we use this in our literacy? Can we use this in art and design? Things like that."*

It is evident that teachers see the value in integrating programming and robotics content with other classroom subjects, however, they would like clear guidance on how to do so in a way that **avoids *"shoehorning"*** programming *"into other areas just to tick a box."* This would help ensure that **meaningful learning** occurs within the classroom, something else that teachers highlighted as important, *"we want to be able to integrate things in a meaningful way."*

Structure and Approach

Alongside the relevance of the training content, teachers were asked to consider what they would consider to be the ideal **level of support** from teacher education programs. One teacher explained that "*to start off, support would be beneficial."* They went on to imply that staff do not necessarily require a large, permanent support system when learning new content and new techniques. Instead, teachers would *"like the support there but may not access it as much once you've gotten the hang of everything."*

Discussion

This study aimed to explore teachers' beliefs about programming and robotics education in primary school and what methods teachers use to teach these topics. Furthermore, it aimed to identify what barriers teachers believe prevent them from teaching these lessons effectively, and what could be done to overcome said barriers. This section discusses the findings in relation to the three research questions outlined in the introduction and additional literature. It also highlights what has been learned from this study and explains how these findings impact the future research directions to be explored in later chapters of this thesis.

What Beliefs Do Teachers Hold About Programming and Robotics Education?

Overall, teachers held positive value beliefs regarding the importance and relevance of teaching programming within early primary school classrooms. Several times during this focus group participants highlighted programming as a vitally important skill that should be introduced to children early on in their schooling. When it came to teaching programming to young children, some teachers believed that programming education automatically required the use of computers or other screen-based technologies. Such beliefs could be problematic for teachers who want to reduce screen-time, do not have frequent access to devices or those who are limited by their pupils' technology skills. Consequently, this belief that programming skills can only be taught with technology may deter teachers from teaching and introducing this subject. However, this belief appeared to be more prominent amongst teachers with less experience in delivering programming lessons. Accounts from those more familiar with the topic suggested that they believed technology was not essential as programming content could be taught using unplugged programming methods. Unplugged methods aim to expose children to programming without using screen-based technologies (i.e., tablets and mobile devices), thus introducing programming principles through tactile and interactive learning experiences (Bell et al., 2009). These links suggest that teacher beliefs about how to teach programming may be linked to teachers' individual experiences. For example, the teachers who were more familiar with programming as a topic area may have been more aware of alternative teaching approaches and thus were less likely to hold this belief about the need for technology.

How Do Teachers Teach Programming Education?

Generally, all participants were familiar with at least one kind of programming technology. When it came to teaching programming to young children, teachers in this study highlighted three teaching methods: educational robotics, screen-based programming applications and unplugged programming. In terms of educational robotics, Bee-bot was mentioned by most teachers in this sample. Teachers reported that their pupils have enjoyed learning with Bee-bot in the past and that they have used these devices for interdisciplinary learning. Particularly, Bee-bots were a useful tool for encouraging children's creativity whilst learning, through activities like obstacle course making. These accounts support the notion that programming activities and educational robotics may encourage children's creativity and expression (Bers, 2020).

According to teachers in this study, an additional advantage of Bee-bot robots is their physicality. Teachers shared that their pupils enjoyed playing and learning in a hands-on manner with these devices. Hands-on learning has been shown to benefit children's learning experiences within programming education. For example, robotics have been used to improve children's computational thinking skills (Ching & Hsu, 2023). Additionally, robotics have been used to demonstrate that learning programming concepts through play with tangible objects can help young learners transition to and understand virtual environments (i.e., Scratch Jr) better in the future (Futschek & Moschitz, 2011).

Regarding screen-based methods of teaching programming, teachers in this study were most familiar with Scratch Jr. Scratch Jr is a downloadable application that can be used on computers or touchscreen devices (i.e., iPads). This program enables children aged 5 to 7 years to create interactive stories and games by connecting graphical programming blocks (Bers, 2018). Scratch Jr can easily be installed on different devices and can be used across different learning environments (i.e., at school and at home). Such digital programs do not require teachers to purchase new pieces of equipment; instead, they can be used on the devices they already own. Teachers in this study highlighted Scratch Jr's versatility as an advantage over robotics toys like Bee-bot.

Teachers in this study believed that both Bee-bot robots and Scratch Jr make programming accessible to pre-literate children, allowing them to introduce these activities to younger age groups. Literature suggests that when programming in these symbolic languages, children learn that different symbols and colours have a corresponding action (i.e., forwards, left turn, right turn; Relkin et al., 2021). Thus, children are not required to read written instructions and instead learn the symbolic functions of different buttons.

Finally, some teachers who participated in this focus group shared how they used unplugged learning activities to teach programming to their pupils. Unplugged learning methods aim to teach children about programming principles through tactile and interactive learning experiences (Bell et al., 2009). Teachers in this study shared how they have used this teaching approach to develop children's instruction writing and sequencing skills (i.e., through tasks that included listing the steps of making a sandwich), and how this approach can be combined with outdoor learning.

Overall, this sample of Welsh primary school teachers have highlighted a range of methods they have used to teach programming to young children. These have included educational robotics toys, computers, and touchscreen devices, as well as unplugged learning activities. However, it appears that some of these pedagogies may be used more than others. For instance, most teachers were familiar with robotics and screen-based methods, however, few teachers touched on unplugged learning approaches. It is possible that teachers' favoured pedagogies within programming education may have been related to their past experiences of teaching the subject. For example, in this study, those that identified themselves as having more experience with programming education were the teachers who used unplugged learning methods in the classroom. On the other hand, those less familiar with teaching programming appeared to believe that programming must be taught using technology. These findings seem to suggest that teacher beliefs are impacted by past teaching experiences which in turn may impact chosen teaching pedagogies. These assumptions are supported by previous conclusions from Fuchs et al., (1994) who argued that teachers' classroom practices and lesson planning behaviours can be influenced by their personal beliefs.

What Do Teachers Identify as Barriers to Programming and Robotics Education?

This focus group study identified a range of first and second order barriers (Ertmer, 1999) that teachers believed had hindered their programming instruction in the past. In describing first order barriers, Ertmer (1999) identified external factors such as limited equipment and teaching support. Meanwhile, second order barriers relate to teachers' own beliefs about things like curriculum priorities and assessment practices.

First-order Barriers

First order barriers identified by teachers in this study included pupil factors (e.g., reading abilities, technology skills), technology factors (e.g., program visualisation, screen time), and external factors (e.g., time, cost of resources and teacher education).

Pupil Factors. Firstly, teachers identified children's reading abilities as a potential barrier to programming education. However, some teachers did also acknowledge the availability of resources like Bee-bot and Scratch Jr that get children programming using symbolic programming languages and do not require proficient reading skills. Additionally, children's technology skills were highlighted as a barrier to teaching programming. Some teachers had approached programming education using computers and laptops which they found young children struggled with. These keyboard-based devices require a certain level of cognitive development to understand the keyboard symbols along with sufficient fine motor development to use the keyboard and mouse (Geist, 2014). The necessity of these skills can make these devices developmentally inappropriate for very young children and consequently, they may find these devices difficult to use. These challenges may prevent teachers from delivering programming lessons if these are the only devices teachers have access to.

Technology Factors. Teachers identified several technological factors that may be first-order barriers to programming education in early childhood classrooms. When considering the limitations of Bee-bot devices, teachers made it very clear that the lack of program visualisation makes learning more difficult for younger pupils. To use Bee-bot robots, children use small buttons on the device to input their sequence which the robot then stores internally. Bakala et al., (2021) explain that this kind of user interface makes programming cognitively demanding due to the high load on working memory, whereby the child is expected to hold their programming sequence in their memory without any visual support. It is evident that this is difficult for early learners as teachers here commented that children would quickly forget what they had instructed the robot to do.

The findings of this study have also highlighted that the screen-based nature of some technologies may be a barrier for some teachers and their schools. Upon returning to the classroom following pandemic restrictions, some schools here were concerned that children had spent a lot of time in front of screens (i.e., TV, phone, iPads etc.) while schools were closed. These assumptions are supported by empirical findings. Salway et al. (2023) investigated the screen-viewing behaviours of children before, during and after COVID-19 lockdowns in the UK and found large increases in children's screentime. This was primarily due to gaming, video calling and television. Consequently, since returning to school, some teachers in this study had made a conscious effort to employ methods of teaching that did not include screens. Thus, if these teachers only had access to screen-based programming games like Scratch Jr, they may have been less inclined to deliver programming sessions to their pupils as they focused their attention on reducing screen time.

Although it could be argued that teachers' motivations to reduce screen time may have been short lived if purely motivated by pandemic restrictions, recent reports suggest that this is still a concern. For example, in September 2023, the UK government launched a new inquiry into screen time following re-emerging concerns about its effects on children's education and well-being (UK Parliament, 2023). The findings of this focus group study suggest that these concerns about the effects of increased screen time on children's education and well-being will likely lead to educators seeking alternative approaches to programming education.

External Factors. Some of the additional barriers highlighted in this study reflect those found in previous studies. For example, teachers in this study identified barriers including the cost of resources (Chang et al., 2010), and a lack of technical support and content support (Greifenstein et al., 2021).

Time. Moreover, like other studies (Khanlari, 2014), teachers in this focus group identified time as a barrier to teaching. They spoke of timetable restrictions and pressure to fit in too many subjects and explained how these pressures had been exacerbated by the pandemic. For instance, upon returning to the classroom, teachers had to catch up on content missed while school closures were in place. The findings of this study provide additional insights into how time barriers have impacted teachers in a multitude of ways. For example, teachers found it difficult to find the time needed to personally seek out teacher education opportunities in subjects like programming and robotics. Moreover, those who had attended teacher education programs previously shared their difficulties in finding time to revisit the content taught to improve their understanding and to adapt the content to suit their classroom. These findings highlight how time barriers are not only related to the time spent teaching in the classroom. The open-ended discussions that took place during this focus group study have resulted in a more detailed understanding of how time may be a barrier for teachers and their teaching practices.

Teacher education. The remaining first-order barriers identified by teachers in this study related to teacher education programs. Teachers in this study shared that even when teacher education opportunities were available within primary schools, these were not accessible for all classroom teachers. Instead, several teachers shared that typically, someone in a more senior role would attend these courses and would disseminate the information back down to teachers at the classroom level. However, classroom teachers are likely to be the ones responsible for delivering this content to pupils, thus not providing them with the required teacher education may hinder their teaching in this area as it is likely that knowledge will not be passed on with the same level of detail.

Although some teachers did share their experiences of teacher education programs, they attended these courses whilst previously teaching children in the latter half of primary schooling (i.e., when teaching children over the age of 8). These teachers were very vocal about past programs lacking content that was appropriate for early primary school classrooms. Furthermore, those teaching younger children could not recall attending teacher education programs for programming and robotics. These discussions provide support for the notion that early primary school teachers are offered fewer opportunities for learning about programming and robotics compared to those teaching older children (Bers, 2010; Bers & Portsmore, 2005).

As programming and robotics is a relatively new focus within the Welsh curriculum, not providing all teachers with the education and support they need is likely to negatively impact teaching and pupil outcomes. One study has suggested that teachers may neglect teaching computing concepts if they believe it to be difficult to teach and do not know how to approach teaching (Larke, 2019). If a lack of knowledge negatively impacts teachers' beliefs about new topics and technologies, this is likely to limit integration within classrooms (Ertmer, 2005; Newhouse 2001). Such findings illustrate the importance of equal teacher education opportunities.

Following the discussions of this focus group, it is evident that providing early primary school teachers with materials and teaching content designed for older children does not aid early primary educators' development or knowledge. Instead, this leaves teachers unable to transfer this knowledge into the classroom and into practice. This is supported by research that has evidenced the correlation between the relevance of training content and transfer outcomes (Axtell et al., 1997).

Second-order Barriers

Second order barriers are typically more complex obstacles as they refer to teachers' personal beliefs (Ertmer, 1999). In the current study, teachers identified their lack of confidence and knowledge as barriers to technology integration. Consequently, teachers who felt inadequately prepared or lacked confidence appeared to be hesitant to integrate programming and robotics education within their classrooms. These discussions reflect the findings of other research investigating teachers' beliefs in this area. For instance, low self-efficacy for programming and robotics is commonly found in samples of primary school teachers (Khanlari, 2016; Ohashi et al., 2018; Ray et al., 2020). Additionally, one study (Sentance & Csizmadia, 2017) surveyed a sample of UK primary school teachers (n = 54) and found that 40% of the sample mentioned a lack of knowledge being a difficulty for them when it comes to delivering programming education. Negative beliefs about programming and new technologies like robotics may lead to avoidance within these areas, thus negatively impacting teaching practices and children's learning experiences. Additionally, discussions in this focus group appeared to suggest that second-order teacher barriers (i.e., low self-efficacy and/or knowledge) may result in a reliance on colleagues to step in and fill gaps in knowledge or confidence on behalf of the classroom teacher (i.e., by stepping in to deliver programming or technology lessons).

What Strategies Can Help Overcome Barriers to Programming Education?

Discussions with teachers in this focus group study have highlighted a range of possible solutions to the barriers they face when trying to teach programming and robotics lessons.

Overcoming Pupil Barriers

Using educational robotics to approach programming education may help overcome pupil related barriers. Some teachers in this study shared that their pupil's technology skills can limit the delivery of programming lessons. For example, some teachers had approached programming education using computers and laptops which they found young children struggled with. For children who struggle with traditional classroom technologies (i.e., laptops and desktop computers), robotics may be a great alternative learning tool. These hands-on tools can be used by children without them having to "log in" and children can operate the device by manipulating tangible blocks.

Another positive of using educational robotics toys with young children is that their programming languages avoid textual and numerical language. For example, the Bee-bot robot has tactile buttons on top of the robot that children can use to input their program (i.e., forward, backward, left, right). Thus, by minimising the level of reading proficiency required, the Bee-bot robot is aimed at pre-literate children which means programming education is not limited by children's reading abilities. However, although hands on and symbolic programming language can help minimise pupil-related barriers, choice of robot is important as teachers in this study considered limitations associated with the Bee-bot robot.

Overcoming Technology Barriers

Teachers highlighted that Bee-bot robots can be difficult devices for young children to use as they do not provide a clear display of the algorithm children program into the bot. These discussions imply that children would benefit from some form of display which clearly shows their programmed route or instructions. Radia Perlman's "TORTIS" was the first robot designed to include a platform for children to place their command cards, thus allowing them to clearly see their programming sequence and make cause and effect connections (i.e., program input to movement output). It was thought that having a clear visual of their sequence would also allow children to reorder their programming sequence more easily if required (Perlman, 1974). More recently, KIBO (Sullivan et al., 2017) was developed with these same principles in mind. To use KIBO, children must scan the barcodes on tangible cube instruction blocks. Placement of these blocks on a flat surface (i.e., table or floor) can then provide children with a clear display of their programming sequence. However, KIBOs instruction blocks have been criticised for being difficult to scan. González-González and colleagues (2019) reported that children in their study often did not wait for the beep or LED to confirm that the code had been read before continuing with their algorithm, leading to mistakes in the sequence. Moreover, children may mix up their blocks, placing them down incorrectly or knocking them out of place once they have been scanned.

To address this, the Cubetto playset (www.PrimoToys.com) combines design features from Bee-bot, KIBO and Radia Perlman's "TORTIS" robot. Like Bee-bot and KIBO, Cubetto playsets are screen free and fully tangible. This will appeal to teachers concerned about screen time within the classroom. However, differentiating itself from Bee-bot, Cubetto comes with a selection of movement tokens that children can physically manipulate to write their algorithms. While this is also a feature seen in the design of KIBO playsets, rather than organising and scanning a selection of blocks, Cubetto requires children to place tokens into an interface board where they are then positioned securely. Thus, there is less risk of children accidentally knocking their tokens out of sequence. Furthermore, the interface board features small LED lights alongside each movement token. These lights flash to indicate which movement token is being executed by the Cubetto robot. This may help children make cause and effect connections between their instructions and Cubetto's actions. This interface board may also overcome the Bee-bot robot's problems of program visualisation by allowing children to physically manipulate their tokens in its large interface board. In doing so, children no longer need to remember the commands they have given to the robot. Instead, they have a clear view of their algorithm, thus allowing them to better predict the outcome of their sequences and to detect errors in their program.

Overcoming Teacher Barriers

Some teachers in this study shared strategies their schools had put in place to help minimise teacher related barriers like low self-efficacy and poor knowledge. These strategies have focused on knowledge sharing and ensuring more confident teachers were supporting less confident teachers. For example, in some schools, more senior members of staff have previously provided drop-in sessions for teachers seeking more support for technology and computing. Additionally, some educators would float around multiple classrooms to help deliver technology lessons to the children. These sessions appeared to be helpful for those less confident in these areas, however, pandemic restrictions quickly highlighted the danger of relying on other members of staff for support. Discussions from this study suggest restrictions like "COVID bubbles" prevented teachers from mixing with multiple classrooms, and thus this negatively impacted the knowledge sharing support strategies that were in place. If teachers need to rely on other teachers to supplement their knowledge and teaching of topics like programming and robotics, children's learning may be impacted when these support systems are not available. This highlights the importance of ensuring that classroom teachers themselves have the knowledge and the confidence to fulfil all aspects of the curriculum, including areas of technology. Effective teacher education programs will help with this.

Overcoming External Barriers

By analysing the experiences and opinions of teachers, this study has identified a collection of teacher education needs that future programs should address. Discussions from this sample of educators suggested that teacher education in this area should (1) provide robotics and programming knowledge that is easily applicable within early childhood classroom contexts; (2) teach content that focuses on interdisciplinary learning and (3) provide structured support while adopting experiential learning approaches.

Developmentally Appropriate Content. Firstly, findings from this study suggest that teacher education programs need to help teachers understand how to deliver lessons that are age-appropriate for their pupils. Discussions here suggest that past teaching programs have not achieved this and have instead provided teachers with materials and teaching content designed for older primary school children. Consequently, early years teachers in this study have been unable to transfer this knowledge into their classrooms and into practice. Thus, there is a need for teacher education courses that clearly explain how programming and robotics content can best be delivered to children under the age of 8. Improving the relevance of teacher education content in this way is likely to lead to improved transfer into the classroom (Axtell et al., 1997).

It is likely that teacher education programs will need to focus on helping teachers think of and create age-appropriate materials and activities for their pupils. Moreover, the importance of appropriate robotics vocabulary was highlighted by the group. The language used in the education sessions should be jargon free to allow teachers to easily access the content. Additionally, educators should be aware of how to later pass on this knowledge and explain these concepts in child-friendly ways. Finally, tailored teacher education programs should highlight a range of teaching methods that are appropriate for delivering programming lessons to young children. This will help increase teachers' awareness of alternative teaching approaches and may help overcome additional barriers. For example, a teacher education workshop that covers educational robotics and unplugged learning methods may be beneficial for teachers who feel programming lessons are difficult to teach as they require access to technologies like computers or iPads.

Interdisciplinary Learning. Secondly, current findings suggest teacher education programs should demonstrate how to teach programming, not as an individual skill, but instead as an approach to developing knowledge of other subjects and general computational thinking skills. To do so, courses should take an interdisciplinary approach, combining programming and robotics technologies with other classroom subjects (e.g., mathematics, literacy, art). If training courses can provide formal guidance on how to integrate robotics and programming within other classroom subjects, it is likely that the frequency of these sessions within the classroom would increase.

Additional benefits of interdisciplinary learning in teacher education programs include saving time as teachers could integrate programmable technologies (like robotics) into their existing curriculums instead of finding the time to teach programming sessions separately to other subjects. Furthermore, by integrating programming education with other classroom subjects, teachers may not need to prioritise other subjects over these technology skills. Discussions in this focus group suggested that some teachers have previously removed technology lessons from timetables to cover other core subjects. Thus, interdisciplinary approaches may aid teachers' understanding of the versatility of programming education and robotics technologies (Greifenstein et al., 2021).

Structure and Approach. Alongside the relevance of the content of teacher education programs, it is important to consider the structure of the support being offered to teachers. Discussions in this study highlighted the expectations placed upon teachers to develop their knowledge of programming and robotics in their own time. Some teachers noted that even after attending teacher education sessions, they were still expected to revisit the technologies used. To help reduce these additional demands on individual teachers, teacher education programs should be structured in a way that allows teachers to gain hands-on experience with the programming technologies featured.

Research has illustrated the benefits of providing opportunities for active learning (i.e., getting participants involved in the course material through hands-on activities) instead of passive instructional methods (see Burke & Hutchins, 2007 for review). Experiential learning approaches have been found to not only increase the likelihood of training transfer but to also improve teachers' confidence. Konen and Horton (2000) found that hands-on training activities reduced teachers' anxiety about the subject they were teaching and increased their confidence. These changes in attitude were found to last over time and remained once teachers began delivering lessons. These conclusions reinforce the value of hands-on training for teachers as well as their pupils.

Time for hands-on learning with the robotics equipment would also give teachers space to think about what would work best for them in practice, and how they could tailor sessions to meet the needs of their pupils specifically. After all, teaching is not a "one size fits all" exercise. Teachers will deliver lessons in different ways depending on the abilities of their pupils and the types of resources and technologies available to them. This personalised approach aligns well with teachers' thoughts about the New Curriculum for Wales which was introduced by the Welsh government around the time this study was conducted. This curriculum is designed to take a flexible, learner-centred approach to learning, allowing schools and teachers to design a scheme of work that reflects the needs and interests of their pupils (Hwb, 2024c). Government guidance for the Curriculum for Wales continues to emphasise the importance of digital skills and ensuring that pupils achieve "digital competence" by the time they leave school (Hwb, 2018). As teachers will not receive formal guidance on how to achieve this from government bodies, teacher education programs which provide opportunities for experiential learning and lesson planning will be immensely valuable.

Overall, these findings have highlighted the importance of prioritising *teacher education* over *teacher training*. Teacher training portrays effective teaching as the mere replication of predetermined mechanical tasks (Stephens et al., 2004). As a result, it is likely that teachers will not feel comfortable replicating learning in their own classrooms if they feel the activities are not suited to their pupils specifically. The discussions above evidence this as teachers who had previously attended training programs could not tailor the content for their younger pupils.

On the other hand, teacher education implies that educators can apply disciplinary knowledge across diverse contexts (Stephens et al., 2004). To be effective, external education programs should lay the foundations, formally introducing teachers to programming concepts and technologies. This will ensure they develop an understanding of programming and robotics education that they can then take back to school and may potentially enhance collaboration amongst colleagues (thus avoiding the one-way administration of support that teachers discussed in this study). Furthermore, providing all teachers with structured support at first will help ensure individual knowledge and confidence are sufficient should the internal support systems not be available (i.e., like they were during the COVID-19 pandemic).

Conclusion

In this study, I aimed to investigate the beliefs primary school teachers held about programming and robotics education and the teaching methods they used to approach these topics. I used a semi-structured focus group to talk directly to primary school teachers about their beliefs and teaching experiences, and I analysed their responses using constant comparative analysis methods. Regarding teachers' personal beliefs, the findings of this study suggested that this sample of primary school teachers held positive beliefs about the importance and relevance of teaching programming and robotics education in early primary school. Regarding their teaching methods in this area, teachers reported using a range of educational robotics (e.g., Bee-bot), screen-based technologies (e.g., Scratch Jr) and unplugged programming methods. However, teachers' chosen methods appeared to vary based on their personal experiences of potential teaching barriers. For example, teachers who did not feel restricted by their own knowledge of programming appeared to employ a wider range of pedagogies to cater to their pupil's needs. Additionally, some teachers' choice of pedagogies was limited by their pupils' technology skills. For example, for teachers with children struggling to operate computers and laptops, screen-based computer programs like Scratch Jr were less accessible. It is important to note that teachers' accounts differed on what they considered to be barriers to programming instruction. It stands to reason that different schools will face different challenges and consequently, different pedagogies may be best suited to certain schools. Furthermore, it appears that opportunities for attending teacher education programs vary greatly not only between primary schools, but also with schools across different year levels (with those teaching older year groups having more opportunities).

Given the variation in teachers' beliefs and experiences within this focus group sample, it would be beneficial to further investigate these topics within a larger sample. Thus, the next steps for my research included developing a mixed-methods survey and distributing it to primary school teachers across Wales. The survey continued to build a bigger picture of the methods currently used to teach computational thinking, programming, and robotics in early primary school classrooms across Wales. This helped me further understand what programming education within primary schools looked like across Wales more broadly before the formal introduction of the New Curriculum. The survey also further investigated teachers' perceptions of barriers and the challenges they faced as they prepared to integrate digital education more frequently within their curriculums as the implementation of the New Curriculum loomed.

Finally, the survey continued to explore teachers' experiences of previous teacher education programs, or lack of experiences, as the focus group findings seem to suggest.

Chapter 3. Exploring Teachers' Beliefs about Programming Education through a Survey.

Introduction

The study described in this chapter continues to explore primary school teachers' beliefs about programming and robotics education. The variation in teachers' beliefs and experiences within my previous focus group study (see Chapter 2) suggested it would be beneficial to further investigate these topics within a larger sample. The insights gained in the previous chapter were used to aid the development of this mixed-methods online survey. For example, my focus group study identified *three* areas of interest that this survey explores further: (1) methods used by teachers to deliver programming education, (2) teachers' beliefs about programming and robotics education generally and (3) beliefs about previous teacher education opportunities in this area.

As I explored in previous chapters of this thesis, changing demands within the modern job market have resulted in international initiatives to feature technology skills more predominantly within education. Consequently, this has seen the addition of programming to primary school curriculums across the world (Balanskat & Engelhardt, 2015; Bers 2020; European Schoolnet, 2015; Uzunboylu et al., 2017), and Wales is no different. For example, both the Digital Competence Framework (Hwb, 2018) and the recently introduced New Curriculum for Wales (Hwb, 2024b) have highlighted *'science and technology'* as a focal point for learning. As part of this learning goal, guidance within the curriculum has emphasised the importance of incorporating computational thinking (CT) and programming skills, even at the early stages of primary education. For educators, CT is described as *"a combination of scientific enquiry, problem solving and thinking skills"* (Hwb, 2018). Guidance then highlights a range of CT skills that teachers should target within their curriculum. These include skills such as sequencing (creating a series of individual steps or instructions ordered to achieve a desired outcome; Brennan & Resnick, 2012) and debugging (identifying and fixing errors in an algorithm or sequence of instructions; Bers et al., 2019).

To teach CT and programming education to young children, teachers in my previous focus group (see Chapter 2) identified methods including screen-based programming languages, unplugged learning and educational robotics. That sample of teachers appeared to be more familiar with screen-based and robot-based teaching methods for CT and programming content, however, discussions highlighted limitations of screen-based learning methods (mainly linked to screen time concerns and pupils' limited technology skills). Further analysis of focus group discussions identified how educational robotics may help overcome these barriers. Educational robotics (ER) encompass physical programmable robots that can be used to engage children in hands-on learning activities. Focus group teachers who had used ER previously believed that robotics had helped them expose their pupils to technological concepts in a way that was fun and engaging. Other research has also shown that ER can be used to support children's CT skills (Bennie et al., 2015; Sullivan et al., 2017; Veenman & Spaans, 2005). Thus, the survey used in this study continued to explore primary school teachers' past experiences with ER and programming education more broadly.

The survey in this study was distributed across the whole of Wales thus increasing the geographical reach of my research, more so than my focus group study. This was important given how accounts and beliefs differed between teachers and schools that participated in my focus group research. For example, teachers' accounts differed on what they considered to be barriers to programming education. It was evident that different schools faced different challenges and consequently adopted different pedagogies to best suit their needs. Moreover, the distribution of this survey was timely in a Welsh context. This survey research was conducted in September 2021. Although the New Curriculum guidelines were published in January 2020 (Hwb, 2024a), they had not yet been fully implemented due to delays following the COVID-19 pandemic. Thus, this survey data was collected at a time when teachers were considering how to reconcile curriculum changes and new learning targets with their own personal beliefs. As a result, the timing of this survey provided a unique insight into teachers' beliefs as these changes were coming to fruition.

Investigating Teacher Knowledge

This survey continued to explore these themes of CT, programming, and robotics. Firstly, to expand upon my previous focus group, this survey investigated teachers' knowledge of CT and how CT skills can be taught to children in early education. Teachers' knowledge of CT was not something that was explicitly explored in my previous focus group study, however, given the clear emphasis on these skills within the Welsh digital curriculums, it is important to investigate teachers' knowledge in this area. For instance, do their definitions and approaches reflect those favoured by policymakers, or do teachers need additional support and information to align their understanding with the government's definition and expectations (i.e., through education programs)? This survey aimed to investigate this.

Investigating Teaching Practices

Secondly, this study continued to build upon my previous focus group research as this survey investigated teachers' pedagogical approaches to teaching programming within the classroom. Pedagogy refers to the methodology and process of how teachers approach teaching and learning (Grossman, 2009) or more simply, their teaching practices. The previous focus group study (see Chapter 2) provided insight into the methods some Welsh primary school teachers had previously used to teach programming and CT. The results of that study suggested that most teachers had previous experience using Bee-bot robots (a tangible robot toy; www.tts-group.co.uk) and Scratch Jr (a screen-based program, www.scratchjr.org). Children can operate Bee-bot robots by pressing buttons on top of the device. These include move forward, move backwards, turn left, turn right, pause, and reset. On the other hand, Scratch Jr is a programming language which enables children to create interactive stories and games by connecting graphical programming blocks (Bers, 2018). A small sample of teachers who participated in my focus group described using unplugged learning methods with their pupils. Unplugged methods aim to expose children to CT and programming without using screen-based technologies (i.e., tablets and mobile devices), thus introducing programming principles through tactile and interactive learning experiences (Bell et al., 2009). This survey aimed to continue building a bigger, collective picture of the methods used to teach programming and CT in early primary school classrooms, and whether current teaching practices aligned with the recommendations within the Welsh curriculum guidance.

Investigating Teacher Beliefs

This study also aimed to investigate teachers' beliefs about CT, programming, and robotics education. As explained in Chapter 1, researchers typically characterise teacher beliefs in two ways: (1) value beliefs (e.g., how important it is to teach programming in early primary years) and (2) self-efficacy beliefs (e.g., teachers' confidence in teaching programming and robotics education). Low self-efficacy for programming and robotics is commonly found in samples of primary school teachers (Khanlari, 2016; Ohashi et al., 2018; Ray et al., 2020) and the findings of my previous focus group study somewhat mirrored those from previous research. However, despite their admissions of low confidence, teachers who participated in the focus group still appeared optimistic about programming and robotics education. To further explore teachers' self-efficacy beliefs, as part of this survey, teachers were asked to think about how confident they were in their future abilities to *learn* how to teach these topics and skills to young children.

This survey study also aimed to explore teachers' beliefs about the long-term benefits of early programming introduction within primary school classrooms. Readings of the current literature suggest that previously, the extent to which value beliefs have been explored is mostly limited to whether teachers believe the integration of these concepts is beneficial for current teaching and student learning goals (Jaipal-Jamani & Angeli, 2017; Kaya et al., 2017; Khanlari & Mansourkiaie, 2015). For example, teachers are often questioned on whether they think robotics can aid learning in maths and science, or whether these devices can encourage engagement and motivation. Few papers explore whether teachers see CT, programming, and robotics education as something that can serve children in the long term. Understanding teachers' beliefs about the long-term benefits of CT, programming, and robotics education is vital for ensuring that educational initiatives align with broader goals of preparing students for future challenges and opportunities in a digital world. Thus, this survey prompted teachers to think about whether these skills are important for future attainment and skill development. For example, teachers were asked to consider whether they believe that programming with robotics could help children pick up other programming languages in the future and whether introducing these skills early may help pupils when they eventually enter the job market.

Investigating and understanding teachers' beliefs is important for several reasons, as illustrated by the findings of previous research. For example, research has shown that negative teacher beliefs can hinder the integration of new digital concepts (like programming) and technologies (like ER) within the classroom. One study (Larke, 2019) interviewed and observed primary school teachers and found that they would neglect teaching computing education if they believed it difficult to teach, thus illustrating how negative beliefs can impact teaching practices. Furthermore, teachers' personal beliefs have been found to prevent technology integration within the classroom (Ertmer et al., 2012). A meta-analysis study from Hew and Brush (2007) analysed 48 studies to identify the three most frequently cited barriers impacting technology integration. Teacher beliefs were ranked within the top three, along with teachers' knowledge and access to resources. Findings like these illustrate how teacher beliefs could be a barrier to programming and robotics education, thus illustrating the importance of research that investigates teacher beliefs.

Beliefs About Barriers

My previous focus group study also investigated teachers' beliefs about potential barriers to programming and robotics education. In line with Ertmer's (1999) previous research, the focus group barriers were categorised as first or second-order barriers. First order barriers are referred to as external factors such as limited equipment, training, and teaching support. Meanwhile, second order barriers relate to teachers' own beliefs about things like curriculum priorities and teaching abilities (Ertmer, 1999).

First-order Barriers. My focus group research (see Chapter 2) identified a range of first-order barriers that were categorised as (1) pupil related factors, (2) technology related factors, (3) external factors and (4) teacher education factors. Regarding pupil factors, I found that teachers highlighted children's limited reading abilities and technology skills as potential barriers to programming education. For example, some teachers in the previous study approached programming using computers and laptops and they had found these technologies to be challenging for young children. These devices, with their keyboard-based input, require a certain level of cognitive development to understand the keyboard symbols, along with sufficient fine motor skills to use the keyboard and mouse or trackpad (Geist, 2014). These insights are valuable for identifying future research directions. For example, educational robotics may help minimise these pupil factors as they are screen and language free. To further my understanding of such barriers, this survey continued to explore how children have previously engaged with programming education from a teacher's perspective and whether teachers believe their pupil's abilities limit programming learning in the classroom.

Regarding technology barriers, teachers in my focus group study highlighted the limitations of both tangible robotics (like Bee-bot) and screen-based programs (like Scratch Jr). For example, teachers believed that the lack of program visualisation with Bee-bot robots made learning more difficult for younger pupils. This is likely because the lack of visualisation can increase working memory demands as children are expected to hold their programming sequence in their memory (Bakala et al., 2021). Moreover, teachers in the focus group held concerns about children's screen time and thus they avoided teaching programming using screen-based methods (i.e., computers, laptops, and iPads). By continuing to investigate teacher beliefs, this survey explored whether these concerns were a short-term consequence of pandemic restrictions, or if these concerns persisted in the following school year.

Some of the external factors identified as barriers within the focus group study reflected barriers found in previous studies. For example, my focus group sample highlighted a lack of resources, the high cost of resources and time constraints as external barriers to programming and robotics integration. This list of barriers supported previous findings from Khanlari (2014) who interviewed 11 Canadian primary school teachers and found that they too identified access to resources, costs, and time constraints as barriers to robotics use. However, both my focus group research and Khanlari's research employed small sample sizes, thus limiting the generalisability of the findings. By targeting a much larger sample size in this study, the survey results and barriers identified should be more representative of Welsh primary school teachers' beliefs more broadly.

Finally, my focus group study highlighted several barriers relating to teacher education programs. Mainly, I found that education opportunities had not previously been available to all early years teachers. Instead, these opportunities were aimed at the more senior members of staff or those teaching older children who are then expected to disseminate the knowledge to other members of staff. Those findings supported the notion that early primary school teachers are offered fewer opportunities for STEM learning compared to those teaching older children (Bers, 2010; Bers & Portsmore, 2005). My focus group research also found that for the small portion of teachers who did get to attend education programs, the content was not always accessible or applicable to early years settings. Instead, they explained that these programs often featured content and activities that were designed for children in later primary school years. Thus, teachers lacked the knowledge and the self-efficacy to transfer this knowledge into the classroom and practice. This survey aimed to explore whether this view was maintained across a larger and geographically broader sample.

Second-order Barriers. A lack of teacher self-efficacy and teacher knowledge were also highlighted by focus group teachers as barriers to programming and robotics education. Those discussions reflected the findings of other studies that found low self-efficacy for programming and robotics to be common in samples of primary school teachers (Khanlari, 2016; Ohashi et al., 2018; Ray et al., 2020). Moreover, one study (Sentance & Csizmadia, 2017) surveyed a sample of primary school teachers (n = 54) and found that 40% of the sample mentioned a lack of knowledge being a difficulty for them when it came to delivering programming education. However, it appears that when past research has measured teachers' beliefs about their knowledge and confidence, these beliefs have been specific to that present moment in time. Although this current study also explored teachers' 'current' beliefs in this way, it also prompted teachers to consider how confident they were in their future abilities to *learn* how to teach these topics and skills to young children. Exploring whether teachers believe they are capable of learning to teach programming and education content successfully will have implications for the approaches adopted by future intervention studies. For example, do teachers need convincing that they are capable of learning these things? Or instead, can interventions jump straight into teaching teachers programming and robotics content as they are already optimistic about the idea of delivering these lessons? Thus, these insights were valuable for guiding the design of the intervention study presented in Chapters 5 and 6 of this thesis.

Overall, my focus group research highlighted a range of interesting barriers that may limit programming and robotics integration in primary school classrooms. Additionally, it also illustrated that teachers' experiences of barriers differed between schools, meaning they either experienced different barriers in different ways, or they experienced different barriers altogether. Thus, the current study aimed to further investigate these beliefs about barriers in a larger sample of primary school teachers to expand upon those previous findings. By doing so, I then explored whether those past focus group findings were supported by data that was collected from a larger sample across the whole of Wales. Moreover, future chapters of this thesis describe school-based intervention research with teachers and their pupils. Gaining a more in-depth understanding of teachers' beliefs about barriers benefited the design and implementation of my future intervention research. For example, researchers have previously made recommendations for future teacher education programs and research studies based on what has worked well in the past (Schina et al., 2021). I strengthened my intervention research by making design decisions for my own teacher education program that have been guided by insights from practising Welsh primary school teachers themselves, utilising their accounts about how they think education programs should be improved.

Current Study

In summary, this survey research aimed to build upon the findings of my previous focus group study (see Chapter 2). That study identified *three* areas of interest that this survey then explored further including: (1) methods used by teachers to deliver programming education, (2) teachers' beliefs about programming and robotics education generally and (3) beliefs about previous teacher education opportunities in this area. It is important to note that teachers in the current study did not complete any intervention or teacher education program prior to completing this survey. Thus, they were asked to share their beliefs based on their individual experiences as practising teachers. This is something lacking in the existing literature as studies have typically explored teacher beliefs following interventions or additional training (Chang et al., 2010; Ensign, 2017, Kim et al., 2015). The survey in this study was distributed across the whole of Wales, thus increasing the geographical reach of my research, more so than my focus group study. This was important given how accounts and beliefs differed between teachers and schools that participated in my focus group research. This consequently aided my understanding of teachers' beliefs as they prepared to integrate digital education more frequently within their curriculums as the implementation of the New Curriculum loomed.

The data gathered in this survey study was used to answer the following research questions:

1) What knowledge do teachers have of computational thinking?
2) What teaching practices are used across Wales to deliver programming and robotics education?
3) What are teachers' beliefs regarding programming and robotics education?
4) What do teachers believe are barriers to programming and robotics education?

Methods

Participants

A self-report survey was distributed to early primary school teachers across Wales. A short advert (see Appendix B) and an online survey link was shared amongst teachers via a school database managed by Techniquest (a children's science centre in Cardiff, Wales). Teachers who taught children aged 4 to 7 years were invited to complete the survey. Seventy-nine teachers started the survey; however, four teachers did not progress past the consent screen and seven teachers who started the survey only answered questions pertaining to their demographic information (less than 40% of the survey questions). They did not return to complete the remaining sections and so were excluded from the following analyses.

The final sample was composed of 68 primary school teachers. Ninety-two percent of respondents were female (n = 62), which was slightly higher than the national average for primary school teachers in Wales (83% as of November 2021, Welsh Government, 2022). Respondents were from 19 of the 22 counties in Wales and all were in-service, practising teachers. Almost all respondents taught in mainstream, Government funded schools, with only one teacher working at a school for children with Special Educational Needs. The majority (84%) were full-time classroom teachers, 10% were part time teachers and 6% held other job roles including Headteacher, Deputy Headteacher, Foundation Phase Leader, or Supply Teacher.

A breakdown of the number of teachers from each year group can be found in Table 3.1. There was nearly an equal number of teachers that taught Year 1 and Year 2 pupils, with a smaller percentage of the sample teaching Reception classes. Additionally, just over a third of the sample taught mixed year groups at the time of completing the survey. For this reason, percentage of teachers for individual year groups adds up to more than 100%.

Table 3.1

Year Groups Taught by Survey Respondents.

Year Group	N	%
Reception	26	38
Year 1	35	52
Year 2	36	53
Multiple/ mixed classes	25	37
Not specified	1	1

Exploring the teaching background of those who completed the survey, years of teaching experience ranged from 1 year to 27 years, with the average number of years in service being 12. All but two of the respondents had obtained their teaching qualification via a bachelor's degree or a Postgraduate Certificate in Education qualification. Only two teachers had a postgraduate certificate or degree that was specific to computing.

Ethical Approval

Prior to completing the survey, teachers provided written informed consent. Information provided at the start of the survey clearly explained that they were not obliged to complete all questions if they did not wish to. As compensation for their participation, teachers could opt into entering a prize draw to win a Cubetto playset for their classroom. This study was approved by the Cardiff University School of Psychology Ethics Committee and is associated with ethics application number EC.21.03.09.6327G.

Materials

This survey was designed using Qualtrics XM survey software (Qualtrics, 2021). Teachers had the opportunity to answer questions regarding demographic characteristics, their knowledge of computational thinking, their beliefs about programming and robotics education including potential barriers to implementation, and finally their beliefs about teacher education programs in these areas. Each of these themes are described in more detail below. Overall, this survey was comprised of 61 items and took, on average, 14 minutes to complete (see Appendix C for full survey).

Background Information

Teachers completing the questionnaire were asked about their gender, job role, years of teaching experience, qualifications, and the age of the pupils they teach. They were also asked to provide information about their school, including school type (i.e., private, state, SEN) and in which county their school was located. Finally, they were asked to consider their pupils' access to technology in the classroom, before estimating their pupils' weekly technology usage.

Computational Thinking

First, without being provided with a formal definition, teachers were asked to state whether they were *"very"*, *"slightly"* or *"not at all"* familiar with the term *'computational thinking'*. Next, they were presented with a collection of statements and were asked to consider whether each statement fit their personal definition of computational thinking (CT). These items were previously used in teacher research by Caeli & Bundsgaard (2020) and included statements like *"CT involves understanding people"*, *"CT involves observing patterns and trends in data"* and *"CT can be done without computers."* In this survey, participants could drag and drop each statement into one of three boxes: (1) *"Does fit my definition of computational thinking"*, (2) *"Does not fit my definition of computational thinking"* or (3) *"I'm unsure about this statement."*

Computer Programming

Teachers were given the following definition of programming (as defined by Gerson et al., 2022, p.5):

> *"Programming (also called coding) can be defined in a variety of ways, but it is often thought of as creating directions or instructions for a computer or robot that direct behaviour (i.e., events and sequences of events)."*

Teachers were then asked about their own experiences of teaching programming and to report whether they taught programming at the time. Those who did were asked how often they taught programming, how much they enjoyed doing so, whether they would like to teach it more often and whether they encountered any challenges when teaching the subject. They were also asked to describe any online resources they used and to list any teaching tools used to help them teach programming whilst considering the advantages and disadvantages of the different methods. Those who did not have experience teaching programming were asked whether they were interested in doing so and could explain why they were/were not interested in an open-ended response.

Educational Robotics

Educational robotics were briefly introduced to teachers as devices that combine accessible and age-appropriate materials to provide children with knowledge of programming through hands-on, practical experiences. Examples of robotics tools were provided based on the findings of my previous focus group study (see Chapter 2). For example, Bee-bot robots (www.tts-group.co.uk) were given as an example as my previous focus group findings suggested teachers were most familiar with this robot device. Teachers were then asked to report to what extent had they used robotics previously. Teachers who had experience teaching with the devices were asked to describe (in a free text response) how they integrated them within their teaching and whether they would like to use them more often. Teachers who reported wanting to use robotics but had no experience doing so were asked to consider why they had not used them in the classroom. They provided their answer in an open-ended text box.

Regardless of their previous exposure to robotics, all teachers were asked to consider how confident they felt, or would feel, using these devices with their pupils. They were then given a list of statements to consider, for example, "*I feel confident I can incorporate robotics with other subjects*." They could then state whether they agreed, disagreed or were unsure about each statement.

Teachers were also presented with a collection of 15 items originally used by Khanlari (2016) to explore teachers' beliefs about robotics. These statements got teachers to consider the potential benefits and disadvantages of using robotics in the classroom. They were asked to drag each item into one of three boxes, categorising whether they agreed, disagreed or were unsure about each statement. Items included, "*Using programmable robotics can promote CT skills*", "*Using programmable robotics can promote pupil collaboration*" and "*Using programmable robotics will increase the amount of stress and anxiety my students experience.*"

The next activity in this section prompted teachers to consider obstacles that may prevent them from using robotics with their pupils. They were presented with a list of 11 potential barriers. Khanlari (2016) previously used this list of potential barriers in a teacher questionnaire and participants could respond by identifying each item as either a *"major obstacle"*, a *"minor obstacle"* or *"not an obstacle."* As these categorisations would not allow teachers to accurately rank one barrier higher than another, the current survey utilised a different response method to allow for a more in-depth exploration of teachers' beliefs about potential barriers. Teachers were asked to report how big of an obstacle they perceived each item to be (on a scale of 0 – 100). They were instructed to give the most points to the item they considered to be the biggest obstacle and fewer points to the statements they considered to be smaller obstacles. They were able to clearly highlight items they did not believe to be an obstacle by checking a box separate from the scales.

Future Development

Teachers were presented with 5 statements regarding programming, robotics, and future development. Statements included, "*Using robotics will encourage my pupils to pursue a STEM career*" and "*Using robotics with my pupils will discourage gender stereotypes in STEM subjects*" (Khanlari, 2016). Teachers could state whether they agreed or disagreed with each statement, or whether they were unsure.

Training and Other Support

The final section of this survey focused on teachers' education opportunities. Teachers were asked to describe any computational thinking, programming, or robotics education they had received, either as part of their pre-service teacher qualification program or as part of additional education provided by their employer. They did so in an open-ended text box. Questions in this section also asked teachers to describe any internal support systems their school had in place for programming and computing. This addition was prompted by the previous focus group findings as teachers in that study discussed the internal support systems their schools had created to assist colleagues struggling with teaching in these areas.

Data Analysis

Demographic data were analysed using descriptive frequencies, as were closed-ended questions. For example, teachers' level of experience with programming was identified using pre-determined answers. Responses for each answer were quantified and percentages were calculated. This survey also explored teachers' beliefs about the benefits of programming education through an open-ended question. These qualitative responses were analysed using summative content analysis which involved identifying keywords and quantifying them. This technique is frequently used for open-ended survey questions (Hsieh & Shannon, 2005).

Results

This section presents the results of the survey completed by Welsh primary school teachers. The presentation of these results reflects the structure of the survey.

Access to Technology

Before exploring teachers' knowledge and beliefs about computational thinking, programming, and robotics, I felt it was important to get a sense of teachers' access to technology resources. Table 3.2 lists the technologies presented to teachers, and illustrates the percentage of teachers using each device within the classroom and the average number of hours their pupils use each device per week. iPads were the most popular device used for teaching with 95% of participants using them regularly. On the other hand, robotics toys were used less frequently, with only 21% of teachers reporting using robotics with their pupils.

Table 3.2

Pupil's Technology Use in the Classroom, as Reported by their Classroom Teacher.

Device	**Total**		**Mean hours per week**
	n	%	
iPads	64	94	2.5
Interactive whiteboard	49	72	3.38
Chromebooks	40	59	2.06
Laptops	27	40	1.63
Computers	15	22	1.06
Robotics	14	21	1.13
Digital camera	4	6	2

Teachers were also prompted to consider their pupils' access to technology during lesson time. Most teachers (62%) indicated that their pupils have frequent access to devices in the classroom. In contrast, 19% of teachers admitted they wished they had access to devices more often.

Regarding the number of devices available per child, 29% of teachers reported that pupils were rarely required to share one device. On the other hand, around 15% of teachers indicated that they often had several pupils sharing a single device. Overall, 35% of teachers wished they had more devices available to them during teaching.

Teachers' Knowledge about Computational Thinking

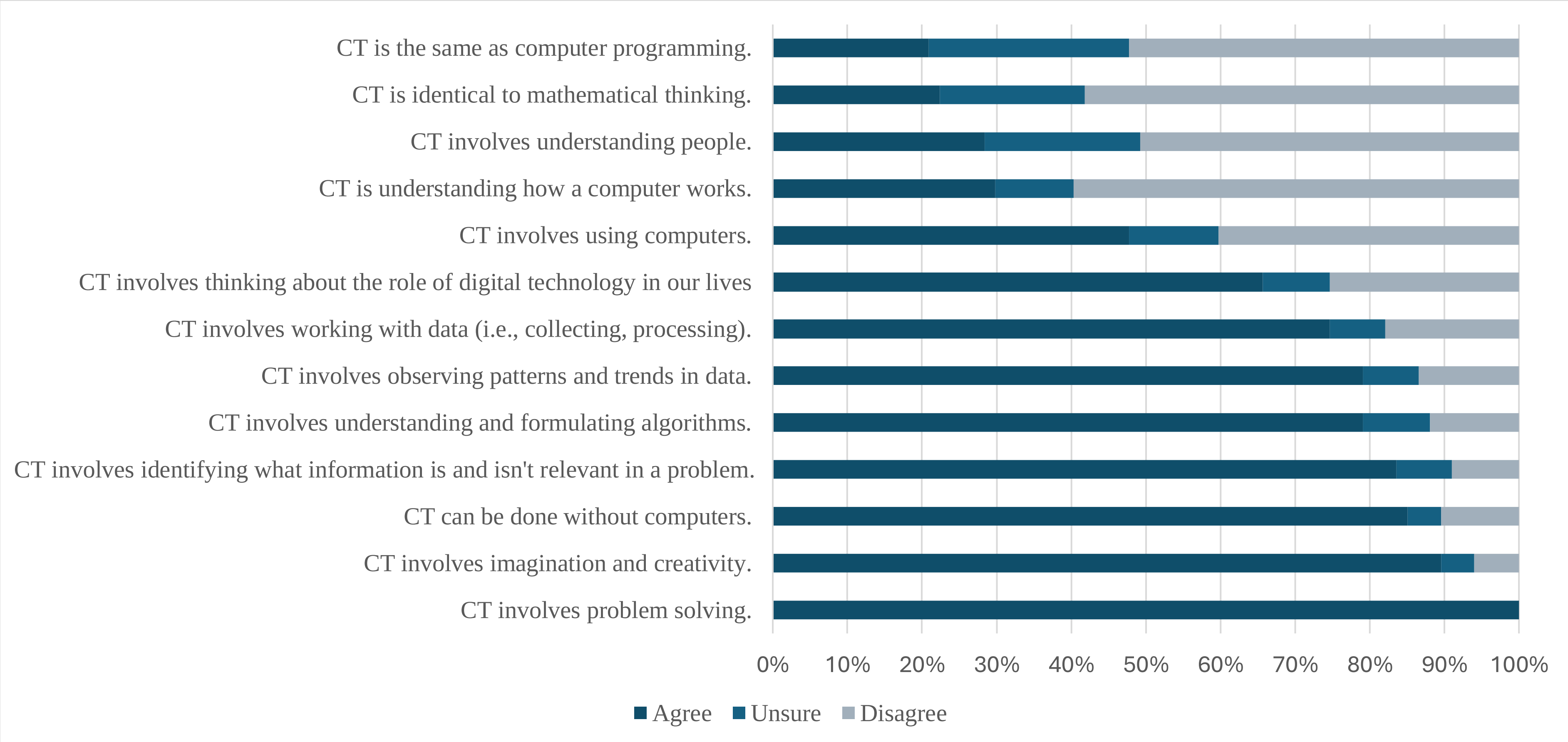
A minority of teachers (15%) felt they were *"very familiar"* with the concept of computational thinking (CT). The majority (59%) identified themselves as being *"slightly familiar"*, whereas 26% stated they were *"not at all familiar"* with the topic.

When exploring teachers' definitions of CT, in general, there was a high degree of agreement among respondents (see Figure 3.1). Most agreed that general cognitive skills such as problem solving, and creativity fit their definition of CT. As shown in Figure 3.1, teachers unanimously agreed that CT involves problem solving. Definitions least agreed on by teachers included CT being equal to or the same as other skills (e.g., *"the same as computer programming"* or *"identical to mathematical thinking"*), as well as subject specific definitions (i.e., statements specific to programming like *"working with and observing patterns in data"* and *"knowledge of algorithms"*).

When considering whether CT involves using computers, teachers' opinions were divided. Around 48% of teachers agreed that CT does involve working with computers, whereas 40% disagreed and 12% remained unsure. However, when teachers were asked whether it is possible to develop CT without computers, most teachers (85%) agreed that it could, with only 10% disagreeing and 4% remaining unsure.

Figure 3.1

Teachers' Definitions of Computational Thinking.



Note: Bars illustrate the percentage of responses for each statement. CT = Computational Thinking.

Computer Programming

Previous Teaching Experiences

Seventy-four percent of the sample were currently delivering programming lessons to their pupils at the time they completed the survey. However, 64% of this sub-group taught programming less than once a month, with 77% of this group stating that they would like to teach this subject more often. On the other hand, 19% of total respondents reported that they did not teach programming to their pupils, however, 62% percent of this sub-group were interested in teaching programming in their classroom.

Through open-ended questioning, teachers were prompted to discuss why they wanted to start teaching programming or wanted to increase the amount they teach. As displayed in Table 3.3, the themes identified focused on benefits for the children. Almost half of respondents perceived learning programming as something that is likely to benefit children's skill development (including problem solving, critical thinking, planning, and creativity). Other perceived benefits for children included their enjoyment in lessons, developing interpersonal skills (e.g., collaboration and resilience) and preparedness for later schooling and the modern workplace.

Table 3.3

The Benefits of Teaching Programming, as Perceived by Teachers.

Theme	Count	Detail
Pupil skill development	15	Problem solving, critical thinking, planning, creativity, logical thinking, mathematical skills
Future careers/development	8	Requirements of the modern workplace
Pupil enjoyment	8	Fun, child interest
Interpersonal skills	5	Collaboration, resilience, motivation, confidence
Lifelong learning	4	Continued development, lifelong skills
Cross curricular learning	4	Mathematics, writing skills
Understanding the digital world	3	How things work
Teacher skill development	1	Improve skills and confidence

Note: Count refers to the number times a theme was mentioned by individual teachers. Themes identified from teachers' open-ended responses.

Methods of Teaching Programming

A summary of the programming tools used by teachers can be seen in Table 3.4. Using Bee-bot robots proved to be the most popular method of teaching programming to children with 84% of teachers using them. Teachers reported that Bee-bot robots were easy to use, age appropriate and kept children engaged through hands-on learning. The second most popular tool amongst teachers was Hwb resources (Welsh Government, 2024a). Through the Hwb website, the Welsh Government provides bilingual, digital services to all schools to support teaching. Teachers liked that Hwb's programming resources were available to everyone, easily accessible, provided a wide range of activities for lessons and were fun for their pupils.

Teachers were also asked to consider the disadvantages of the programming tools they had used. One common disadvantage was that screen-based programs (e.g., Scratch, Hwb resources, purple mash, JiT5, J2Code/ J2ECode and Code.org) often require children to have mastered basic computing skills (i.e., logging onto a computer, sufficient motor skills). Furthermore, with some programming languages and devices (e.g., Bee-bot and Scratch), teachers found it difficult to monitor and measure pupil progress. Teachers also highlighted the difficulty of using one tool to cater to a range of abilities. They reported that some tools did not cater for high ability pupils (e.g., Bee-bot, JiT5), and others were not suitable for low ability pupils (e.g., Hour of Code, J2Code/ J2ECode, JiT5)

Teachers also reported a range of situational variables that can make using certain programming tools more difficult. Alongside a lack of physical devices, a lack of supporting resources for different tools also impacted their teaching. Additionally, some tools (e.g., Lego, Code.org) required teachers to support pupils directly during learning, and teaching in smaller groups meant that more staff members were needed to deliver a lesson successfully.

Table 3.4

Summary of Tools Used by Teachers to Teach Programming and their Perception of Benefits and Disadvantages (According to Free Text Responses).

Tool	Benefits	Disadvantages	N	%
Bee-bot	Easy to use. Age appropriate. Adaptable for different learning levels. Visual learning. Pupil enjoyment. Pupil engagement. Cross-curricular learning. Hands-on learning.	At times not suitable for high ability children. Battery powered. Easily damaged. Requires understanding of left and right. Limited number of devices. Lack of additional resources. Out of date. Difficulty monitoring pupil progress.	37	86
Hwb resources	Accessible for all. Easy to use. Can be used outside the classroom. Variety of activities and resources. Pupil enjoyment.	Difficult for pupils to log in. Limited to laptops/ computers. WIFI required.	12	27.91
Scratch	Good for KS2. Accessible for teachers. Pupil enjoyment.	Difficulty monitoring pupil progress. Difficult for pupils to log in. Lack of teacher knowledge.	7	16.28
Purple mash	Age appropriate. Pupils can practice different skills. Requires little planning or preparation. Pupil engagement. Accessible to whole school (consistency).	Difficult for pupils to log in. Requires computer skills. Tailored for English curriculum.	5	11.63

JiT5	Good for younger children. Simple to use. Pupil enjoyment. Accessible on iPads.	Requires understanding of left and right. At times not suitable for high ability children. Requires strong motor skills with computer mouse.	5	11.63
Lego	Pupil enjoyment.	Few devices available. Requires adult supervision (pupil:adult).	4	9.3
LOGO			3	6.98
J2Code/ J2ECode	Challenges children. Easy to theme. Pupil enjoyment.	Not suitable for early years. Difficult for pupils to log in. Lack of motivating rewards.	3	6.98
Unplugged coding	Easy to apply to everyday contexts. Fewer devices/ resources needed. Easy to link with other topics.	Difficult to link to specific skills.	2	2
Codeapillar	Simple. Age appropriate.	Battery powered. Few devices available.	2	4.65
Lego (wedo) kits	Pupil enjoyment. Encourages teamwork.	Expensive. Time consuming to set up.	2	4.65
Twinkle app	Simple and clear instructions.	No ownership over work.	1	2.32
Scratch Jr	Easy to use. Can be used outside the classroom.	Few devices available.	1	2.32
Robo Mouse	Easy to use. Encourages independent learning. Pupil engagement.		1	2.32
Minecraft	Pupil enjoyment. Encourages teamwork.	Program limited to laptops. Limited access to laptops across the school.	1	2.32
Microbots			1	2.32
Micro:Bit	Easy to use. Accompanying resources available via website.	Fragile and delicate.	1	2.32

Kodable	Easy to use. Visual learning.		1	2.32
Hour of code	Good graphics and sounds.	Not suitable for early years.	1	2.32
Harry Potter codable wands	Engaging. Good for younger children.	Technical problems.	1	2.32
Code.org	Pupil enjoyment.	Requires adult supervision. Requires basic computer knowledge.	1	2.32
Bee-Bot app	Encourages independent learning.	Difficulty tracking pupil progress.	1	2.32

Robotics

Past Teaching Experiences

Just over half of the sample (59%) had used robotics with their current class before, but stated they did not use them often. In contrast, 26% of teachers had never used robotics before (with or without their pupils). Of this sub-group, 95% of teachers would like to use robotics with their pupils with one teacher (5%) remaining unsure.

When describing classroom activities previously delivered using robotics, it appeared that teachers mainly used the devices to target cross-curricular learning. For example, with a focus on learning in mathematics, teachers described using Bee-bot robots to support children's learning of topics such as positioning, angles, directions and learning numbers. These skills were taught by encouraging children to move the robot to the correct locations on a floor map. Teachers reported that such activities can be easily altered to fit a range of different themes. For instance, one teacher explained that "*Printable floor maps are really good to keep children focused and with the changeable jackets you can adapt them to suit a range of topics rather than a discrete learning activity*." Several teachers also described using robotics to combine mathematics and creativity by making symmetrical art. Interestingly, one teacher described an unplugged programming task inspired by the robots they used. They looked to utilise the same skills "*on an outdoor number square / word grid with a human Bee-Bot, giving and receiving directions."*

Benefits and Disadvantages of Robotics

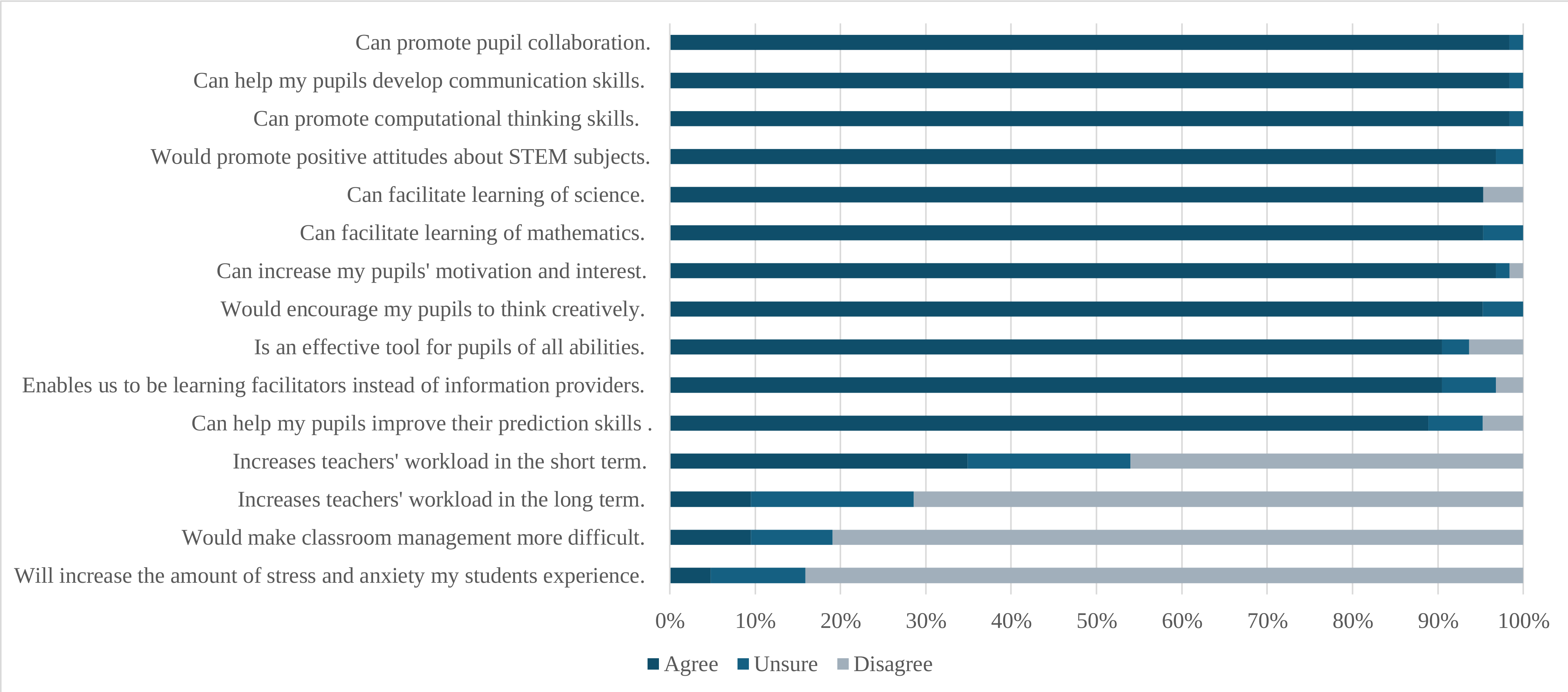
As illustrated by Figure 3.2, there was generally a high degree of agreement between teachers when they considered the potential benefits of learning with robotics, and opinions were generally positive. Teachers almost unanimously agreed that learning with robotics can promote a range of skills including computational thinking, communication, collaboration, creativity, and prediction. Similarly, teachers were mostly in agreement that robotics can support the acquisition of knowledge in other subject areas including mathematics and science. Furthermore, most teachers felt that robotics would encourage positive beliefs in pupils, would not cause their pupils stress and anxiety, and importantly, would be an effective tool for pupils of all abilities.

When prompted to think about the potential negative impact of using robotics in the classroom, most teachers disagreed with statements such as *"using robotics would make classroom management difficult"* and *"would increase teachers' workload in the long term."*

There was less agreement amongst the group when asked whether they believed robotics integration *"increases teachers' workload in the short-term"*, with 35% believing it would increase short term workload, 46% believing it would not, and 19% remaining unsure.

Figure 3.2

Teachers' Beliefs About the Potential Pros and Cons of Teaching with Robotics.



Note: STEM (Science, Technology, Engineering and Mathematics).

Barriers to Using Robotics

Teachers ranked potential barriers to using educational robotics in the classroom by allocating more points to larger barriers, and fewer points to minor barriers. The points allocated to each statement by each teacher were averaged to form the ranking displayed in Table 3.5. Responses determined that access to technical resources was thought to be the largest barrier to using robotics in the classroom. The second largest barrier was the lack of technical support provided to teachers, with teachers' confidence closely following as the third largest barrier preventing robotics integration. On the other hand, teachers allocated fewer points to potential barriers including pupils' age and motivation.

Table 3.5

Teachers' Rankings of Potential Barriers to Robotics Integration.

Rank	Statement
1	There are not enough technical resources available in school
2	Teachers do not have adequate technical support
3	Teachers do not feel confident enough to use robotics in their classes
4	Teachers do not have adequate instructional support
5	Teachers are unsure how to make robotics relevant across multiple subjects
6	Teachers do not have the time to learn how to integrate robotics into their lessons
7	There is too much course content to teach to find time for robotics
8	Class sizes are too large to plan lessons using robotics
9	Schools do not have the space to store and use multiple robot devices in the classroom
10	My pupils are too young to be able to understand and work with robotics
11	My pupils would not be motivated to learn with robotics

Note: Statements with higher rankings (i.e., 1st) were identified as larger barriers.

Teacher Self-Efficacy

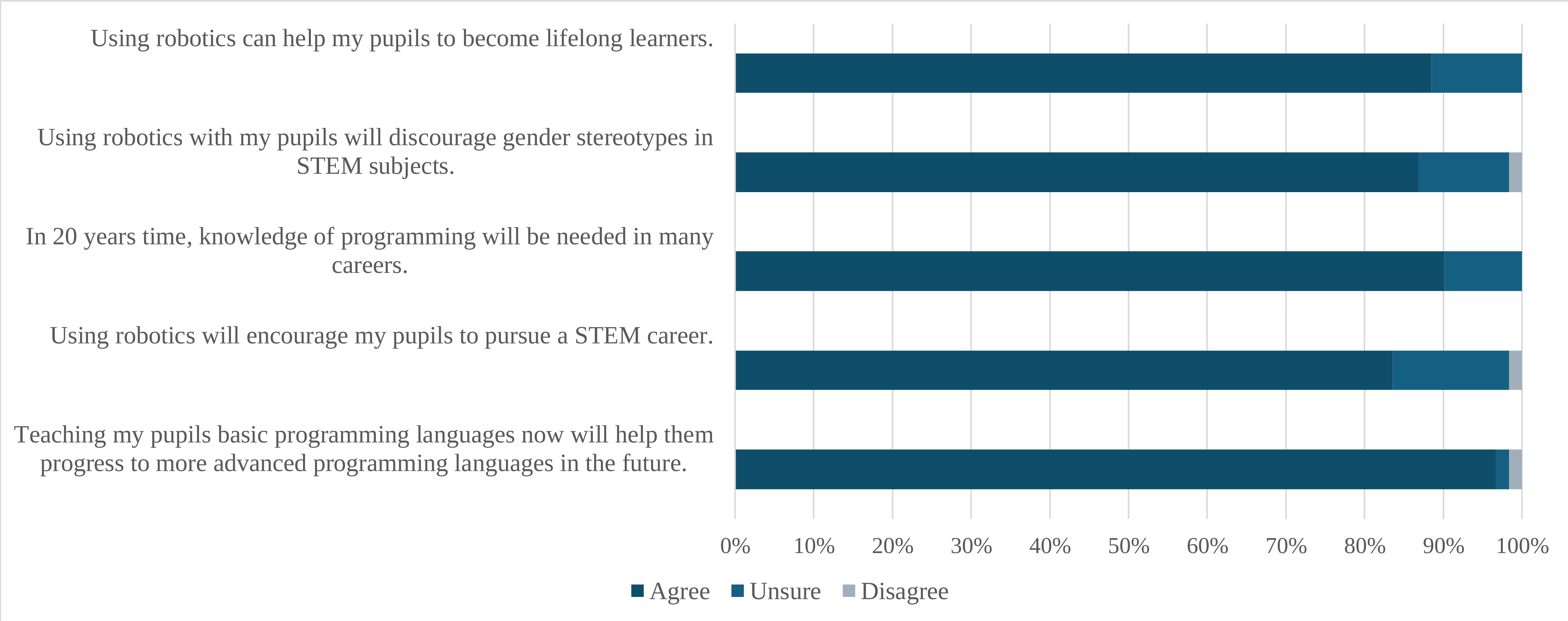
Despite teacher confidence being identified as a top 3 barrier, most teachers stated that they would feel somewhat comfortable using robotics devices in their classroom. Similarly, almost half of teachers said that they would feel comfortable incorporating robotics with other subjects. Although a quarter of the sample felt they would not be comfortable using robotics in their teaching practices, a large portion of teachers (80%) stated that they were confident they could learn how to use these technologies with their pupils. Very few teachers (8%) completely disagreed with this statement.

Future Pupil Development

When responding to statements about robotics use and future development for pupils, teachers held very positive beliefs. As shown in Figure 3.3, almost all (97%) teachers believed that teaching pupils basic programming skills at a young age will help them progress to more advanced programming languages in the future. Additionally, 90% of them felt that programming knowledge will be needed in most future careers and 84% believed that using robotics will encourage their pupils to pursue these STEM (Science, Technology, Engineering and Mathematics) careers. Regarding pupil attitudes, 87% believed that robotics can discourage gender stereotypes in STEM subjects, and 89% believed these devices could encourage pupils to be lifelong learners. Overall, teachers appeared to be in agreement regarding the benefits of robotics for future development. Although small groups of the sample were at times unsure, few disagreed with the statements.

Figure 3.3

Teachers' Perceptions of the Benefits of Robotics for Future Development.



Note: STEM (Science, Technology, Engineering, Mathematics)

Teacher Education and Other Supports

These results explore teachers' reflections on past teacher education opportunities. These include the pros and cons of past courses and identified needs that teachers would like future education programs to meet. Finally, this section presents findings about support systems created within schools to support teachers' knowledge and delivery of programming and robotics lessons.

Previous Education Opportunities

Seventy-five percent of teachers reported that their pre-service training (i.e., BEd, PGCE etc.) did not include guidance on teaching computational thinking, programming or robotics to their pupils. In fact, only 18% of respondents recalled being given guidance on these topics during their teacher education program.

It appeared that some schools instead have offered inhouse training to aid teachers' knowledge of computational thinking, programming and robotics. Forty-three percent of the sample reported that their school had, at some point, provided teacher education in these areas, with the majority clarifying that this included advice on how to use robotics in the classroom. An open-ended question prompted teachers to consider the programs they attended. They were asked to think about what they did and did not enjoy, as well as what they felt was missing from the programs. A summary of these responses is shown in Table 3.6.

Many teachers highlighted the benefits of hands-on exploration with robot devices in these sessions. One teacher shared that they "*Enjoyed being able to have time to 'play' with the equipment before implementing it*." Moreover, in another teacher's reflections, they shared that "*It was so hands on, everyone had to use the resources themselves, so we came away feeling confident in an aspect of it."* These comments illustrate the importance of hands-on learning for teachers as well as their pupils when getting to grips with educational robotics.

Although some schools may have offered additional teacher education opportunities, these sessions have not been accessible to all members of staff. One teacher explained that "*Staff are able to attend any courses on coding they wish…. However, I feel as if some teachers struggle to fit the activities into an already jam-packed timetable*." Additionally, another teacher described,

> "*After school training. Trialled in personal time no additional funding for trainer or trainees. Whereas in other professions especially health service these would be paid*

sessions. There is a limit to the goodwill staff can give I know this sounds negative, I don't want to be, I love the children, but teachers need to have their time valued too."

It is evident that in a lot of cases, the onus is on the individual teacher when it comes to additional education in this area. While schools encourage staff to pursue these opportunities it is "*down to teachers to explore as part of personal choice/ continuing professional development.*"

Table 3.6

Summary of Teachers Open-Ended Descriptions of Previous Teacher Education Opportunities.

Pros	Cons	Needs
Having time to play with the equipment before implementing in class. Hands-on and practical.	Unable to utilise training because equipment is expensive.	How to teach basic concepts to young children.
Understanding how to use robotics across the curriculum.	Ideas taught in training not suitable for younger age groups.	Variety of activity ideas and resources.
Example activities shared.	Training focused on numeracy and literacy.	How to use robotics simply and effectively with young pupils (i.e., under 8 years old).
Improved understanding of how computational thinking can be implemented across the curriculum.	Training now outdated – reliance on staff to keep up to date.	Using robotics in the training session.
Improved understanding of the importance of teaching computational thinking.	Physical robotics were not used in the training session.	

Teacher Education Needs

The table above also highlights the education needs of teachers as identified by those who have attended education programs at some point in their professional career. Through open-ended questioning, teachers were able to provide additional detail regarding their education needs based on where their previous programs fell short. Teachers would like support on how to teach basic concepts to young children, with one teacher commenting "*I loved the training I attended however, as I am in a reception class, the ideas and activities were not relevant."* Other teachers had similar reflections on their past training experiences, for example, another teacher said, "*The training I received was aimed at upper KS2, some advice and information on how to teach basic skills to younger pupils would be great."* Although teachers in early years classrooms may have robotics training provided to them, it appears that the sessions are not always tailored to suit the needs of children in younger year groups (i.e., under the age of 8 years old). As a result, teachers have struggled to deliver sessions that fulfil the digital skills requirements of the curriculum, "*Children in unsupported classes with unconfident teachers can quite often end up in a 'token' ICT/programming lesson 'button bashing' on a Bee-Bot."*

Those who said they had not received guidance on how to deliver robotics training to their pupils were then asked whether they felt they needed teacher education programs in this subject area. Just over 91% of teachers stated *"yes"*, that they do need training, while around 8% remained unsure. No teachers responded *"no"* to this question.

Internal Support Systems for Technology Use

When asked whether their school provided internal support for the teaching of programming and other computing concepts, 56% of the sample said yes, with 30% saying no and 14% remaining unsure. Through open-ended questioning, those who responded *"yes"* were asked to describe the support they had access to. Naturally, if teachers required support when teaching programming and computing lessons, they turned to the ICT lead or coordinator. Additionally, teachers described the importance of peer support in this subject area, "*Teachers with good programming skills will support teachers who find it difficult."* None of the teachers surveyed identified themselves as someone who provided this kind of additional support.

Nineteen teachers answered a question that asked whether COVID-19 restrictions had impacted their access to additional teaching support. Of these teachers, nine said that restrictions had impacted support, seven answered *"no"* and three were unsure. Through an open-ended question, teachers could provide details about how they had been impacted. Most teachers did not respond to this question, however, one teacher identified that COVID bubbles meant that they *"haven't been able to welcome outside agencies into school to teach pupils about computer coding."* Additionally, two teachers suggested that support had been limited due to the lack of face-to-face contact.

Discussion

This survey was developed using the findings from the focus group study described in Chapter 2 and further investigated what methods were used by primary school teachers to deliver programming education in Wales. It also further explored what teachers' beliefs were about programming and robotics education generally and what their beliefs were about previous teacher education opportunities in this area. This section will discuss the findings in relation to the research questions outlined in the introduction and additional literature. It will also draw comparisons with the results of the previous focus group study and will highlight what has been learned from this study before outlining how these findings impact the future research directions to be explored in later chapters of this thesis.

What Knowledge Do Teachers Have of Computational Thinking?

Most teachers surveyed in this study were at least somewhat familiar with the concept of computational thinking (CT). Respondents had the greatest confidence that CT involved problem solving, which aligned with how CT has been defined by both policymakers and researchers. For instance, curriculum guidance for Welsh primary school teachers defines CT as "*a combination of scientific enquiry, problem solving and thinking skills."* (Hwb, 2024a). Moreover, Wing (2006) also highlighted problem solving as a key aspect within their definition of CT, defining it as *"solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science"* (p.33).

Not only did the current sample of teachers define CT in ways that reflected formal definitions, but their ideas were also aligned with how CT has been previously conceptualised by other practising teachers across the globe (Corradini et al., 2017; Denning 2009; Fessakis & Prantsoudi, 2019; Garvin et al., 2019; Sands et al., 2018). For example, Rich et al's (2019) sample of American primary school teachers identified specific aspects of CT they believed to be directly related to problem solving. These included skills like debugging (i.e., identifying and fixing errors) and abstraction (i.e., identifying relevant details). Similarly, Garvin et al., (2019) also investigated American primary school teachers' conceptions of CT and found that most responses described CT as problem solving. In addition to problem solving, teachers in this survey study also conceptualised CT as something that involves imagination and creativity. These beliefs are supported by several studies (Brennan & Resnick, 2012; Shute et al., 2017) which have highlighted that the development of CT competencies involves creative problem solving to a certain degree.

It appears that teachers in this survey study had a good understanding of what CT is and what it entails. This is evident by the fact that their chosen definitions were aligned with curriculum guidelines, research definitions and other teachers' definitions of CT as described by past research. However, it seems that there was less consensus on *how* CT can be taught. Garvin et al., (2019) had previously found that their sample of primary school teachers often associated CT with computers or technology in general. Interestingly, the current study found that teachers held mixed views on the role of computers in teaching CT skills. Although most teachers acknowledged that CT *can* be done without them, the group was divided on whether CT does or does typically not involve using computers. It may be that although teachers are aware that CT *can* be done without a computer, they tend to adopt computer-based approaches to delivering these sessions.

Several things may have influenced whether teachers' approaches to delivering CT lessons typically involved computers. Firstly, my previous focus group study (see Chapter 2) suggested that teachers' access to technology and other resources is likely to greatly impact their choice of teaching methods. For instance, some teachers may only have access to computers or other screen-based technologies as opposed to alternatives like tangible robotics. Data from this survey further supported this notion as iPads, Chromebooks and laptops were used weekly by almost half of the sample. On average the time spent using these screen-based devices was twice or three times longer than the use of educational robotics. Furthermore, teachers' knowledge of alternative approaches like unplugged learning may be limited by their past teaching experiences. In this survey, all teachers that agreed CT could be taught without computers were a part of the minority sub-group that stated they were very familiar with CT. Those who disagreed or were unsure whether computers were an essential resource were those not at all or only slightly familiar with the concept of CT.

This finding reflects the focus group discussions in the previous chapter, whereby experience level appeared to influence teaching practices for programming content. Teachers with less programming experience tended to assume that the subject could only be taught using technology, whereas those more familiar with the topic had knowledge of alternative teaching methods (i.e., robotics or unplugged learning approaches). In the case of CT, while teachers may have been aware that there were alternatives to computer-based learning, they may have had little knowledge of *how* these alternative methods could be implemented. Ensuring teachers have knowledge of how to teach CT without screen-based technologies will likely benefit pupils' learning as teaching will not be limited by their teacher's knowledge of or the availability of technological resources.

What Teaching Practices Are Used Across Wales to Deliver Programming and Robotics Education?

Most teachers in this study taught programming to their current cohort of pupils at the time of completing the survey. When delivering these lessons, teachers reported using Hwb online resources and activities, the Scratch Jr application, and educational robotics. Hwb (Welsh Government, 2024a) is an online platform made to aid and support teachers. Through this platform, teachers in Wales can access free apps and software, virtual classrooms, and a large bank of materials that they can then use in the classroom. On the other hand, Scratch Jr is a programming language which enables children aged 5 to 7 years to create interactive stories and games by connecting graphical programming blocks (Bers, 2018). When it came to tangible robot toys, Bee-bots appeared to be used most by teachers within this sample and this finding reflects the discussions held in the previous focus group study (see Chapter 2). Additionally, Bee-bots popularity is reflected in published research with most studies investigating how robot-medicated can promote the development of CT skills in young children favouring this device (see Bakala et al., 2021 for review). Children can operate Bee-bot robots by pressing buttons on top of the device. These include move forward, move backwards, turn left, turn right, pause, and reset.

Across both my focus group and the current survey, teachers highlighted the importance of using robotics for interdisciplinary learning. Here, teachers have already attempted to integrate robotics with subjects including mathematics and art. For example, teachers described using Bee-bot robots to support children's learning of topics such as positioning, angles, directions and learning numbers. These accounts reflected the discussions held in my previous focus group study whereby teachers highlighted the benefits and importance of interdisciplinary learning.

The findings of this survey have also illustrated that when teaching programming and robotics education, teachers frequently feel the need to seek out additional support. Reflecting my past focus group findings (see Chapter 2), teachers here also shared accounts of *teachers supporting teachers*. Respondents described how they have often turned to more senior or more knowledgeable colleagues for guidance. Interestingly, survey respondents had mixed opinions on whether COVID-19 restrictions had negatively impacted their access to these additional support systems. Such responses did not align with discussions from the previous focus group study, as most teachers who answered survey questions on this topic felt support had not been impacted. In contrast, a smaller group of teachers did share examples of how teaching support had been affected (i.e., mainly due to COVID bubbles) and these accounts reflected those given in the previous study. These differences in responses may be due to differences in COVID-19 restrictions at the time each study was conducted. Teachers completed this survey in September 2021, whereas the previous focus group study was conducted in May 2021. It may be that by September, there was more variability in pandemic restrictions between schools (i.e., some schools enforcing COVID bubbles longer than others).

What Are Teachers' Beliefs Regarding Programming and Robotics Education?

It is evident that teachers see value in utilising hands-on learning approaches with young children, as demonstrated by their use of tangible robotics (i.e., Bee-bot) and unplugged programming methods. Unplugged methods aim to expose children to CT and programming without using screen-based technologies (i.e., tablets and mobile devices), thus introducing programming principles through tactile and interactive learning experiences (Bell et al., 2009). Supporting previous research (Khanlari, 2016), teachers here believed that programming and robotics education could positively impact pupils' interpersonal skills (i.e., collaboration, teamwork, communication, sharing ideas with others) and scientific enquiry skills (initiating, planning, performing, recording, and interpreting).

The survey distributed in this study not only explored teachers' beliefs about the short-term benefits of programming and robotics education, but survey questions also explored whether teachers perceived this learning as something that will benefit their pupils in the long term. This is something that few papers have measured previously, instead focusing on short term benefits for children's cognitive and social skills (see Tzagaraki et al., 2022 for review). Overall, teachers here believed that programming and robotics education could positively impact children's lifelong learning skills. Furthermore, the results indicated that teachers believed that their pupils would likely require skills like programming as they progress through school and enter the job market. This awareness is likely due to recent changes in curricula across Wales, which now explicitly highlight the importance of advanced digital skills like programming (see Welsh Digital Competence Framework; Hwb, 2018).

Moreover, teachers believed that exposure to robotics whilst young may help discourage future gender stereotypes. Findings from past research suggest that such beliefs about the impact of robotics exposure on gender stereotypes are accurate (Master et al., 2017). Sullivan and Bers (2019) employed a sample of 105 children (aged 4 to 8 years) and found that engaging in a collaborative robotics curriculum increased girls' interest in being an engineer when they grew up. Moreover, while boys were more interested in engineering at the start of the study, boys and girls were equally interested in the subject following robotics exposure. Thus, not only do these findings support the positive belief held by teachers in this study, but these findings also highlight the power of early exposure to robotics and programming in defying stereotypes toward technology and engineering fields. Combining teacher awareness about the benefits of educational robotics for discouraging stereotypes with research evidence of implementing robotics may bring about positive changes in pupil beliefs within the classroom.

It is encouraging to find that teachers in this study are knowledgeable about the potential short- term and long-term benefits of introducing programming and robotics education in early years education. This has resulted in a clear desire to teach this in the classroom. For instance, most teachers surveyed wanted to increase the frequency with which they teach programming and use robotics, and those without these teaching experiences already were keen to start teaching the subject. This then poses questions about why teachers are not teaching this content more frequently. This survey's exploration of potential barriers to programming and robotics integration may provide some insight into why this is the case.

What Do Teachers Believe Are Barriers to Programming and Robotics Education?

This survey study highlighted a list of barriers that teachers identified based on their experiences of trying to integrate CT, programming, and robotics into their teaching activities. These barriers support and, in some cases, expand upon those identified in my previous focus group study (see Chapter 2). As in the previous focus group chapter, teaching barriers could be categorised as first or second-order barriers. Ertmer (1999) previously identified first-order barriers as factors external to the teacher such as limited equipment and teaching support.

Alternatively, second-order barriers relate to teachers' own beliefs about things like curriculum priorities and teaching ability.

First-Order Barriers

First-order barriers identified by teachers in this study included pupil factors (i.e., their technology skills), external factors (i.e., lack of resources, time limitations) and teacher education factors (i.e., the content of education programs).

Pupil Factors. In this survey study, teachers were asked to consider the advantages and disadvantages of the teaching methods they had used in the past to teach programming content to their pupils. Teachers here highlighted several pupil factors that can make using different approaches difficult. Firstly, supporting the findings of the previous focus group study (see Chapter 2), teachers shared that pupils' technology skills can pose a challenge, particularly when using screen-based methods of teaching (i.e., Scratch Jr and some virtual Hwb activities). This is because keyboard-based devices require certain cognitive skills to understand the keyboard symbols as well as strong motor skills to use the keyboard and mouse (Geist, 2014). Additionally, children must be able to log into these computers and the chosen programming software. The necessity of these skills can make these devices developmentally inappropriate for very young children and consequently, they may find these devices difficult to use. These challenges may prevent teachers from delivering programming lessons if these are the only devices teachers have access to. Teachers in this study also shared that children's lack of knowledge of left and right can hinder their ability to use and understand programming languages. Their responses suggested that these difficulties occur when teachers approach programming education using both tangible robotics and screen-based programming methods.

External Factors. The external barriers identified in this study appeared to relate to teachers' teaching environments. In accordance with existing literature (e.g., Khanlari, 2016) and my previous focus group study, teachers perceived inadequate access to technical resources to be the largest obstacle preventing robotics integration. Furthermore, teachers highlighted additional barriers including inadequate technical and instructional support. In addition to these similarities, there are also differences between the findings of this study and those of previous research. In Khanlari's (2016) study, teachers believed robotics to be a time-consuming subject, that could prevent them from covering essential parts of the curriculum. Teachers in the current study were also asked to consider whether they believed there was too much course content to teach to find time for robotics in the classroom. The results indicated that teachers were less concerned about finding the time in the school day to teach with robotics, perhaps suggesting that this is more likely to be a minor barrier than a major barrier for those surveyed here. It may be that teachers feel robotics devices can be integrated with subjects and existing topics, thus costing no extra time.

Alternatively, for teachers in this study, it may be that the limitation of *'time'* is not always in reference to classroom, lesson time and using the robots (as identified by Khanlari, 2016). Instead, teachers here have found it difficult to find the time needed to build their knowledge of how to use these devices. It was highlighted by teachers in this survey and my previous focus group that there is an expectation for teachers to develop their own skills and knowledge in their own time, and consequently, they are not always paid for seeking out additional professional development. This idea is supported by the finding that most teachers here agreed teaching with robotics would increase their workload in the short term (i.e., whilst they develop their knowledge), but the majority disagreed that this would be a long-term effect.

Teacher Education Factors. Following on from the previous focus group study, the findings from this survey again highlighted the lack of teacher education opportunities available to early years primary school teachers within the field of programming and educational technologies. This survey found that two thirds of primary school teachers were not provided with guidance on teaching CT, programming, or robotics during their pre-service training. This is concerning given the Welsh government level focus on developing children's digital skills in early schooling. This finding supports the previously argued notion that early childhood teachers are not given the same teacher education opportunities as those teaching older children, specifically when it comes to STEM subjects (Bers, 2010; Bers & Portsmore, 2005; Sullivan & Moriarty, 2009).

Of course, some teachers who completed this survey *had* previously attended some kind of teacher education program relating to CT, programming, or robotics. Through this survey, these teachers were able to reflect on their previous education experiences. Those who had not attended such courses were also able to comment on characteristics they would like education programs to include. Consequently, the findings of this study may have important implications for what content teacher education programs should include and how they should be delivered to ensure that teachers are able to effectively integrate CT, programming, and robotics within their classrooms. These recommendations are supported by my past focus group findings and published literature (Schina et al., 2021).

Developmentally Appropriate Content. In line with the results of my previous focus group (see Chapter 2), the findings of this survey suggest education programs need to help teachers understand how to deliver lessons that are age-appropriate for their pupils (specifically the 4 to 7 age range). To do so, courses should break down CT and programming concepts to ensure that the content is digestible and accessible for children as young as 4 years old. According to respondents, current programs are not tailored to suit the needs of those teaching children under the age of 8 years old, leaving early years teachers lost as they attempt to navigate this section of the curriculum. As discussed in the previous chapter, teacher education programs should prioritise assisting teachers in developing age-appropriate materials and activities for their students. Moreover, it is important that programs use suitable, jargon-free language to enhance accessibility. Educators should also be equipped to convey this knowledge to their pupils in child-friendly ways. Finally, programs should further tailor their content by showcasing diverse teaching methods suitable for delivering programming lessons to young children. Fostering teachers' awareness of alternative approaches may help address additional barriers. For instance, a workshop covering educational robotics or unplugged learning methods could benefit teachers facing challenges in teaching programming due to limited technology access. Research suggests that improving the relevancy of professional development content is likely to increase information transfer into classroom practices (Axtell et al., 1997).

Interdisciplinary Learning. Current findings suggested that teachers would like to teach programming, not necessarily as an individual skill, but instead as an approach to developing knowledge of other subjects and general CT skills. Therefore, education programs should also provide support for interdisciplinary learning to highlight how CT, programming and robotics can assist teachers in achieving learning goals across subject areas, instead of just being an instructional add-on (Greifenstein et al., 2021). The results of this survey illustrate that teachers have appreciated and enjoyed past programs that have attempted to do so, mainly because an interdisciplinary approach allows educators to utilise their existing body of knowledge.

Knowledge of Different Approaches. In line with findings from my previous focus group study, few teachers discussed using unplugged methods to teach programming and CT content. The current findings suggested that while some teachers had knowledge of alternative approaches to teaching programming, chosen teaching approaches are likely to vary due to teachers' pupil cohorts. As children in early years education possess limited computer skills, informing teachers how to teach CT and programming without the use of computers (e.g., using the Computer Science Unplugged Curriculum; Bell et al., 2009) is likely to prove beneficial. The very few teachers who did mention using unplugged programming activities with their pupils highlighted the lack of guidance available in this area. Teacher education that focuses on unplugged activities or robotics would not only emphasise learning methods suitable for younger children but may also help teachers understand how CT does not need to be paired with computer use in the classroom.

Structure and Approach. The findings of this survey suggest that many Welsh primary school teachers may not be familiar with robotics technologies. Thus, it is important that education programs do not assume teachers have had previous experiences with these devices. Instead, programs should provide opportunities for hands on learning with educational robotics. Past research has evidenced that hands-on training is advantageous (Agatolio et al., 2017; Kim et al., 2015) and has identified hands-on practice as one of the *"best practices"* in educational robotics teacher training courses (Schina et al., 2021). Reflections explored in this study support this and suggest that hands-on learning for teachers is an approach that has worked well in the past. Furthermore, teachers in this study contested that hands-on learning in education programs improved their confidence, further supporting previous research in this area (Konen & Horton, 2000). As discussed in the previous chapter, it is likely that including experiential, hands-on learning within teacher education programs may also save teachers time and reduce their workload as they would no longer need to allocate time post-training to revisit materials to adapt what they have learnt for practice in their classrooms.

Research has highlighted the importance of ongoing and continuous teacher education programs (Schina et al., 2021; von Wangenheim et al, 2017) to help teachers develop their knowledge of CT, programming, and robotics, and to aid their understanding of what it means to think computationally and how to engage their students in computing ideas (Yadav et al., 2017). However, the current findings suggest that teachers usually attend one-off workshops and afterwards, there is a reliance on educators to keep up to date with the latest advice for the delivery of CT, programming, and robotics content. Naturally, this is something teachers find difficult to do due to time restraints and frequent developments in this area (both technological and knowledge advances). Research into teacher education programs shows that this kind of exposure to applied concepts through isolated workshops is unlikely to provide long-term gains in the classroom (Harris & Sass, 2011; Desimone, 2009).

Combining the current survey findings with those from the previous focus group study, it is recommended that education programs offer structured support for teachers as they are first introduced to these topics and technologies (i.e., through interactive workshops). However, that is not to say that educators require a large, permanent support system (i.e., through attending frequent workshops). Discussions from the previous focus group suggested that formal support and contact from experts could be lessened once teachers believed they understood the content. However, it is worth noting that it would likely be beneficial for teachers if they could access additional information and guidance as needed going forwards. Thus, in the intervention study presented in Chapters 5 and 6, teachers attended an afternoon education workshop prior to delivering a robotics curriculum in the classrooms. They were then able to access advice and support from researchers when delivering the curriculum.

Second-order Barriers

Second-order barriers typically comprise of barriers specific to the internal beliefs of the teacher (Ertmer, 1999). Low self-efficacy for programming and robotics is commonly found in samples of primary school teachers (Khanlari, 2016; Ohashi et al., 2018; Ray et al., 2020) and the findings of this survey mirror those from previous research. However, this survey also asked teachers to consider whether they believed they *could* learn to successfully deliver programming and robotics lessons to their pupils. These results showed that teachers had higher levels of confidence when it came to considering their future abilities and skills as most believed they would be able to learn how to use robotics with their pupils. This suggests that teacher self-efficacy beliefs may not be a long-term barrier to robotics integration and teachers remain optimistic regarding their ability to teach robotics at some point in the future. It is likely they need to be provided with effective teacher education programs for them to unlock this potential.

Evaluation and Conclusions

This study was a continuation of the focus group study described in Chapter 2. Thus, the purpose of this study was to further explore Welsh early primary school teachers' beliefs about CT, programming, and robotics education. These ideas are being emphasised more and more in primary school curriculums across the globe (Balanskat & Engelhardt, 2015; Bers 2020; European Schoolnet, 2015; Uzunboylu et al., 2017), and Wales is no different. The enforcement of the Digital Competency Framework (Hwb, 2018) and introduction of the New Curriculum (Hwb, 2024a) have resulted in changes to teaching experiences across Wales, similar to changes experienced by those in other countries with similar curriculum goals. Thus, although this study has utilised a Wales-centric approach, the beliefs and accounts of teachers explored here may be reflective of wider teacher experiences.

Firstly, these findings help produce an overall picture of methods currently used by teachers to teach CT and programming in early primary school. In line with previous focus group discussions, Bee-bot robots and screen-based applications like Scratch Jr are popular teaching tools. Furthermore, both studies provided insight into how these technologies may not work for all teachers and pupils. For example, some pupils find Bee-bot's lack of program visualisation challenging and teachers' concerns about screen time and pupil's technology skills can limit the use of screen-based technologies (e.g., computers and iPads) for programming lessons. Teachers' further reflections on barriers and challenges that may hinder programming and robotics lessons also match those discussed in my previous study. These barriers included pupil factors (i.e., pupils technology skills), teacher factors (i.e., their knowledge and confidence) and external factors (i.e., availability of resources and teaching support).

The findings of this study also provided more insight into teachers' beliefs about CT and digital education following the focus group. By surveying a larger group of teachers, this study has identified additional pupil factors that may hinder programming education. For example, teachers here highlighted that children's spatial knowledge (i.e., knowing left and right) can limit the methods they use to teach programming and related CT skills. Unplugged learning methods may allow children to still engage with CT and programming concepts through activities that do not require these directional skills. For example, children can first be introduced to the concept of algorithm construction through picture sequencing tasks. This point emphasises the importance of ensuring educators are aware of these alternative teaching approaches, however very few teachers in this study and the previous focus group study described using unplugged learning methods. Incorporating unplugged learning content within teacher education programs could be beneficial for several reasons. Firstly, unplugged methods can make programming concepts accessible to lower ability children. Secondly, these approaches can help teachers develop children's CT and digital skills without requiring access to technology (Brackmann et al., 2017). This may be beneficial for teachers who do not have frequent access to a class worth of technical resources.

Both the Digital Competence Framework and guidelines from the New Curriculum for Wales state that teachers must develop children's digital skills and computational thinking abilities, however, teachers feel there is little guidance on how to do so. The findings of this survey and the previous focus group study have highlighted that even as these changes come to fruition, many teachers do not have the opportunity or the means to seek out additional teacher education opportunities in areas like programming and CT. It appears that within early primary education, there are not equal opportunities for additional education in these technological areas. Instead, classroom teachers are expected to pursue additional education courses in their own time. Due to the time pressures teachers face, many do not seek out these additional opportunities and thus do not develop the knowledge needed to deliver CT or programming content, nor the confidence to trial new technologies with their pupils. As a result, teachers involved in these research projects often described utilising internal school support strategies. Here they shared accounts of *teachers supporting teachers* as they sought instructional support for programming and robotics content from more experienced colleagues.

Those who have attended teacher education courses highlighted several short fallings of past programs and made recommendations for improvements that reflect those discussed by focus group participants. Firstly, those designing teacher education programs must ensure that the programming and robotics content covered is developmentally appropriate for children aged 4 to 7 years old. To do so, content should focus on helping teachers create learning materials and activities tailored to their pupils' abilities. Secondly, teachers would like guidance on how to best integrate programming concepts and robotics technologies with other classroom subjects. These two studies have also highlighted the importance of providing hands-on learning opportunities for teachers during professional development sessions. Those who have experienced this kind of training have emphasised its benefits for learning and teacher confidence.

Next Steps

Moving forward with this thesis, I use the knowledge gained from these two teacher focused studies to aid the design of a classroom intervention study (described in Chapters 5 and 6). To suit the needs of primary school teachers, this intervention study provided teachers with a robotics curriculum that was interdisciplinary and engaged children with CT and programming through hands-on, playful methods. As discussed in the previous chapter, the Cubetto robot has been identified as a tool that is developmentally appropriate for children (aged 4 to 7 years) to use in the intervention, mainly due to its tangible programming language and program visualisation platform.

The insights gained from teachers' discussions of teacher education programs were also used in the intervention study. This intervention included a teacher education program that was designed using the insights gathered from primary school teachers in Wales. As recommended by educators, teacher education included guidance on how to adopt an approach to teaching programming that was developmentally appropriate for children between the ages of 4 and 7 years old. For example, it considered how to introduce CT and programming concepts to young children and how to design activities they would find accessible and engaging. The participating teachers were also provided with lesson plans that incorporated other classroom subjects with programming and robotics activities, before being given time within the session to come up with their own interdisciplinary activities. This allowed teachers to gain hands-on experience with the Cubetto playsets whilst also encouraging collaboration between those attending the workshop. Discussions with teachers have highlighted how important it is to allocate time for these experiential learning activities during the session, rather than expecting teachers to find additional time to revisit the content at a later date. Following these recommendations helped provide the structured support and environment teachers desire, consequently improving their knowledge and confidence (see Chapter 6).

Whilst my focus group and survey studies have provided insights into teacher beliefs that aided the design of the teacher education portion of my intervention, further research was needed to investigate programming and robotics integration from the perspective of the children. This ensured that the school-based robotics intervention was effective for children as well as teachers. Thus, my next study (Chapter 4) investigated how children could learn with robotics like Cubetto. This survey highlighted that children's spatial knowledge (i.e., their understanding of left and right) may limit the methods teachers use to teach programming and related CT skills. Thus, the next chapter in this thesis describes a laboratory investigation that aimed to explore the impact of children's spatial skills on programming performance and the potential effectiveness of embodied learning techniques.

Chapter 4. Visual Perspective Taking and Robotics: Can Embodied Learning Aid Programming?

Introduction

So far, this thesis has investigated teachers' beliefs about programming and robotics education. However, as the overall aim of this thesis is to investigate robotics integration from both teacher and child perspectives, in this chapter I now focus on children's learning experiences with an educational robot. The study was motivated by previous findings that have illustrated how spatial ability can impact children's achievement in STEM subjects such as science (Hodgkiss et al., 2018) and mathematics (Gilligan et al., 2017; Gilligan et al., 2019). Specifically, this study focused on the role children's visual perspective taking skills may play in programming and robotics education. It addressed two main questions: (1) Does performance on programming tasks (i.e., robot programming, debugging and prediction) relate to performance on visual perspective taking and executive functioning tasks? (2) What role does embodied learning play in programming performance when accounting for visual perspective taking and executive functioning abilities?

Computational Thinking

Recent changes to primary school curriculums have highlighted the importance of developing children's computational thinking (CT) skills. Within the Welsh curriculum, CT has been defined as *"a combination of scientific enquiry, problem solving and thinking skills"* (Hwb, 2024b). In line with this definition, scientific literature and curriculum guidance frequently highlight algorithm writing, debugging and prediction skills as CT skills to target within early education (Bers, 2020; Hwb, 2024b; see Chapter 1 for more detail). Writing algorithms involves creating *"processes or sets of instructions to be followed in calculations of other problem solving operations"* (Hwb, 2024b). This can include creating sequences of instructions to be followed by a programmable robot. Debugging has been defined as identifying and fixing errors in an algorithm (Bers et al., 2019) and prediction skills involve the ability to follow algorithms, determine their purpose and predict outcomes (Hwb, 2024b).

Although not regularly categorised as a CT specific skill, spatial skills have been suggested to be highly relevant for CT abilities and programming performance (Jones & Burnett, 2008; Parkinson & Cutts, 2018). For example, Città and colleagues (2019) assessed a sample of primary school pupils (*n* = 92, aged 6 to 10 years) and found a positive correlation between programming abilities (using a pencil-paper algorithm writing task) and spatial skills (using a mental rotation task). It is important to note that "spatial skill" does not refer to a singular concept, but instead a collective term for a diverse array of abilities (Linn & Petersen 1985; Newcombe & Shipley 2015; Uttal et al., 2013). For example, both Mental Rotation and Visual Perspective Taking (VPT) are thought of as "spatial skills", however, both require different thought processes. On the one hand, Mental Rotation involves the ability to imagine an object in an orientation different to the one viewed (Frick & Pichelmann, 2023; Parkinson & Cutts, 2018). On the other hand, VPT refers to the ability to mentally adopt a viewpoint different to one's own and involves imagining a change of one's own position (and often orientation; Frick & Pichelmann, 2023).

Flavell (1977) previously defined two levels of VPT. Level-1 VPT involves knowledge about which objects are visible from another point of view. To successfully undertake VPT at this level, children must recognise that someone else may have a view that is different to their own, and so may or may not see the same objects as them. This level of VPT appears to develop between infancy and children's second year of life (Moll & Tomasello, 2006; Sodian et al., 2007). Flavell (1977) also explained that level-2 VPT goes beyond simply recognising that others may see things differently, and instead involves the ability to adopt an alternative visual perspective. At this level, children must be able to imagine what others see from their visual perspective and anticipate *how* the objects and scenes appear to the other person (i.e., what is near, far, left, or right).

The ability to execute level-2 perspective taking is seemingly more complex and there is no evidence showing level-2 perspective taking in infants (Apperly & Butterfill, 2009). Instead, some studies looking at children's understanding of *how* an object appears to an observer suggest that children under the age of 8 default to an egocentric view without recognising *how* their visual perspective may be different to someone else's (Huttenlocher & Presson, 1973; Piaget, & Inhelder, 1956; Yadollahi et al., 2020). However, it is worth noting that there is evidence to suggest that children may be able to demonstrate level-2 VPT from around 4 or 5 years of age (Flavell et al., 1980; Masangkay et al., 1974; Pillow & Flavell, 1986). Thus, it seems likely that there are individual differences in these abilities within the age range of 4 to 8 years old.

VPT and Programming with Educational Robotics

VPT skills may be especially important when using educational robotics (ER), a tool popular with Welsh primary school teachers (see Chapters 2 and 3) due to their physicality within a child's immediate environment. In primary education, most ER devices are controlled through direct interaction with the robot. Examples include the Bee-Bot (www.tts-group.co.uk), the KIBO robot (Sullivan et al., 2017) and the Cubetto robot (www.PrimoToys.com). For example, to use the Cubetto robot, children must plan the movements of the robot by programming an algorithm consisting of movement tokens (i.e., move forwards, left and right). Importantly, children can write and easily view their algorithm using Cubetto's interface board (something that teachers have noted the Bee-bot robot lacks, see Chapters 2 and 3 for discussion). Thus, the Cubetto robot requires children to transform physical movements into symbolic instructions. This may be more difficult when the robot's spatial orientation (i.e. direction it is facing) does not align with the child's. For example, when the child is facing the robot (orientation difference of 180-degrees) the child's left and right are opposite to the robot's. Thus, more advanced VPT skills (i.e., level-2 VPT) may be required for the child to program an accurate algorithm from the perspective of the robot.

Research with older populations has evidenced that, even in adulthood, people struggle with tasks that require them to operate an object from an incongruent perspective (i.e., a spatial orientation that does not align with their own). Cho et al., (2017) conducted research investigating the misalignment problem during drone flying, whereby the drone's left becomes the flyer's right when it is travelling toward them. In this study, they compared participants' performance on an obstacle avoidance task while using a flying program (that would match the drones left/right to the participant's) to performance without this corrective program. Cho and colleagues found that being able to operate a drone from an egocentric perspective (i.e., using the perspective correcting program) resulted in better obstacle avoidance.

Although Cho and colleagues' study employed a constantly moving piece of technology, the same perspective taking principles can be applied to ER. When programming a robot, if its spatial orientation is incongruent with the programmer's (i.e., the child's), then this must be corrected mentally as they pre-plan their algorithm. Thus, it seems possible that poor VPT may hinder young children's ability to program a tangible robot when that robot is operating from an orientation that is different to their own. This study investigated whether children's VPT skills were related to their performance on tangible robot programming tasks.

VPT and Executive Functioning

This study also explored whether children's VPT abilities were related to their executive functioning skills. Executive functioning refers to a set of self-regulatory, cognitive processes that aid in the monitoring and control of thought and action (Carlson, 2005). These skills include working memory, inhibitory control, planning and cognitive flexibility (Dempster, 1992; Welsh et al., 1991; Zelazo et al., 1997).

If level-2 perspective taking is important for successfully programming ER, it seems likely that the ability to ignore conflicting information would be important. For example, when children are presented with a conflict between two perspectives (and thus two differing spatial frames of reference) their own perspective (the conflicting information) must be ignored for the child to be successful at imagining another's perspective. Thus, it is likely that perspective taking ability is associated with executive functioning, more specifically with inhibitory control (Diamond et al., 2002).

Findings from Frick and Baumeler (2017) support this notion. They administered assessments of level-2 VPT and inhibitory control (i.e., the 'Fruit Stroop Task') to 6-year-old children (n = 140). They found a significant correlation between perspective taking and inhibitory control. Furthermore, a linear regression analysis found that, even after accounting for effects of control variables (age, verbal-IQ and socioeconomic status) and other mental transformation abilities (mental rotation abilities), inhibitory control accounted for a significant part of the variance in perspective-taking performance. Given that past research has evidenced a significant effect of inhibitory control on perspective taking abilities, the current study employed a measure of executive functioning abilities that was developmentally appropriate for children aged 4 to 7 years old. This measure was thus included as a control variable that may be associated with individual differences in perspective taking, which may consequently impact robot programming performance.

Can Embodied Learning Aid Programming Performance?

I have described how level-2 VPT skills may be required when programming a tangible robot and highlighted research suggesting that this level of VPT can be difficult for young children. Furthermore, research from Cho et al., (2017) suggested that aligning the participant's body orientation with the orientation of a mobile object can improve the participant's ability to navigate the object as they no longer need to correct the incongruent perspective. Thus, this study applied embodied learning principles to learning with an educational robot.

Embodied learning practice, part of embodied cognition theory, is a learning paradigm that emphasises the use of the body in learning activities (Anderson, 2003; Kosmas & Zaphiris, 2018). Embodiment within education is currently a popular topic amongst researchers, and this is illustrated by the number of recent literature reviews that have investigated EL within a range of educational contexts (Aartun et al., 2022; Fugate et al., 2019; Georgiou et al., 2019; Hegna & Ørbæk, 2021; Zhang et al., 2021). Studies in the embodied education field have shown that EL in the classroom can promote children's cognitive abilities in maths (Cook et al., 2017), science (Lu et al., 2011) and language acquisition (Cassar & Jang, 2010). However, few studies have attempted to examine the benefits of embodied learning within programming and computational thinking education (Kallia & Cutts, 2023). Thus, this study aimed to investigate the role of embodied learning when completing an algorithm writing task with a Cubetto robot.

Research findings have also suggested that embodiment methods can aid visual perspective taking. One study (Huttenlocher & Presson, 1973) investigated whether children (aged 9 to 10 years old) could better imagine what a display array would look like by using bodily movement to remove the incongruence between their own perspective and that of an imagined observer. All children were shown a display which was then hidden from view. After viewing the display, some children physically moved around the table to imagine the display from the perspective of an imagined observer. They then indicated what they believed the array would look like from the new position. Other children remained in place when imagining what the array would look like from an alternative perspective. The results of this study indicated that children who moved around the table to imagine and visualise how the display would look performed significantly better than those who remained in one position. These findings evidence how embodiment can be used to align child-observer orientations and thus remove the need for VPT skills.

It seems likely that embodied learning methods may aid programming abilities with a tangible robot as children would not need to suppress the conflicting orientation and perspective of themselves relative to the robot (using inhibitory control skills). A conceptual paper by Kallia and Cutts (2023) emphasises the importance of embodiment and actions for children's conceptual understanding of programming concepts specifically. In their paper, they define several types of actions. For example, they note the advantages of "physical actions" (i.e., whole body movement or partial body interactions) when the movements have high action congruency. This means that the actions are semantically linked to what is being learned. Additionally, they define "instrumented-symbolic actions" as actions which involve the use of bodily movement to create symbolic instructions to control and manipulate an object (i.e., a robot). The current study combined the notion of physical actions with instrumented-symbolic actions using a Cubetto robot.

In an embodied learning condition, children were instructed to stand in the robot's place on the map, thus embodying the robot's position and orientation. They then took physical steps forward and completed turns left/right as they designed an algorithm for the robot to follow. This engaged children in full body, physical actions which were linked to the movements and goals of the robot. In this study, I then explored whether children in the Embodied Learning condition completed more trials in an algorithm writing task than children in an Incongruent Programming condition (who remained seated with a misaligned orientation to the robot) or children in a Congruent Programming condition (who sat with a child-robot aligned orientation at the start of each trial). I also investigated whether children who were encouraged to use embodied methods when learning with Cubetto at the start of the experiment later performed better on additional programming-related assessments (i.e., debugging and prediction) in comparison to children assigned to Incongruent and Congruent programming conditions. Through these explorations, I assessed whether embodiment helped children transfer their programming knowledge to additional programming-related assessments.

Current Study

To summarise, level-2 VPT is required when operating a mobile object (like a robot) from a visual perspective that is different to one's own. Previous research suggests that this skill develops throughout early childhood (Flavell et al., 1980; Huttenlocher & Presson, 1973). As children in early primary school years (i.e., under the age of 8 years) may not have the ability to consistently demonstrate level-2 VPT, it is possible that young children with poor perspective taking abilities may find programming educational robotics difficult as they must write an algorithm from a perspective that is, at times, incongruent to their own. This study investigated whether children's VPT abilities are related to their programming abilities. That is, I assessed whether children with higher VPT skills performed better in programming tasks (including algorithm writing, debugging and prediction tasks), possibly due to their ability to envision the visual perspective of the robot. Furthermore, given that past research has evidenced a significant effect of inhibitory control on perspective taking abilities, participants in this study completed an executive functioning assessment. This measure was then included as a control variable that may be associated with individual differences in perspective taking, which may consequently impact robot programming performance.

Finally, the explored literature highlights the benefits of embodied cognition for children's learning in maths, science, and literacy. However, research investigating how embodiment can support learning in programming education is limited (Kallia & Cutts, 2023), particularly when using educational robotics as a learning tool. It is possible that embodied learning methods may benefit children's learning with educational robotics, as bodily movement may remove the incongruence between the programmer's visual perspective and the robot's. Consequently, this study investigated whether embodiment techniques designed to support perspective taking resulted in better programming performance with a Cubetto robot. Techniques included sitting children behind Cubetto (which provided a congruent visual perspective with the robot) or getting children to walk through the algorithm as they planned it for the robot (i.e., taking an embodied stance).

The following research questions were explored:

1) Does performance on programming tasks (i.e., robot programming, debugging and prediction) relate to performance on visual perspective taking and executive functioning tasks?
2) What role does embodied learning play in programming performance when accounting for visual perspective taking and executive functioning abilities?

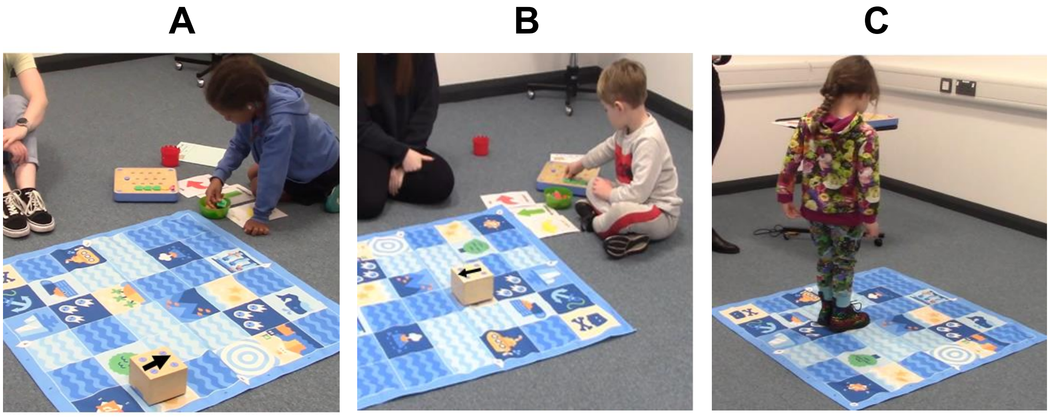
Methods

Design

This study employed a between subjects design whereby participants were randomly assigned to one of three conditions during an algorithm writing task: (1) Incongruent programming (i.e., unmatched child-robot orientation with unaligned visual perspectives, see Figure 4.1a), (2) Congruent programming (i.e., unmatched child-robot orientation with aligned visual perspectives at the start of the task, see Figure 4.1b), or (3) Embodied learning (i.e., participants moved through space whilst programming the device with aligned visual perspectives throughout programming trials, see Figure 4.1c). Participants were randomly allocated to a research condition based on when they attended their lab session, with conditions being consistently rotated between 1, 2, and 3 as children participated. This method of allocation ensured a relatively even distribution of participants across the conditions, minimising the risk of selection bias. These conditions allowed for manipulations of the independent variable, and level of embodiment. Dependent variables measured in this study included programming skills (i.e., algorithm writing, prediction and debugging), visual perspective taking (VPT) and executive functioning.

Figure 4.1

Examples from Each Experimental Condition During the Algorithm Writing Task.



Note: Image A shows the Incongruent programming condition, image B is the Congruent programming condition and image C shows Embodied learning. The black arrow added to the image indicates the robot's orientation. Place cards are positioned near the interface board to remind children of the function of each token.

Ethical Approval

Both parental and informed consent from the child was obtained prior to the study. Both parents and children were made aware that they could cease participation at any time without providing an explanation. This study was approved by the Cardiff University School of Psychology Ethics Committee and is associated with ethics application number EC.18.11.13.5468.

Participants

G*Power 3.1 (Faul et al., 2007) was used to conduct a power calculation to determine the required sample size for MANCOVA analyses. The analysis was conducted using an effect size of .12 measured by Cohen's f^2, an associated alpha of .05, a power of .95 (Cohen, 1988), three independent variables (Incongruent programming, Congruent programming and Embodied learning conditions), three dependent variables (algorithm writing, debugging and prediction assessments) and three control variables (age, executive functioning and visual perspective taking). The estimated total sample size was 69.

Participants for this study were recruited through social media advertisements and a recruitment database. Parents confirmed that their child did not have any significant developmental delays. Seventy-eight children between the ages of 4 and 7 years old (*M* age =

5.73, 39.74% female) participated in this laboratory study, however not all children chose to complete all five assessments. All children completed the algorithm writing and prediction tasks (*n* = 78), however, several children opted out of the debugging (*n* = 3), VPT (*n* = 4) and executive functioning tasks (*n* = 4). Additionally, several children (*n* = 5) did not pass the practice trial at the start of the visual perspective taking assessment and thus their data from this task was excluded from the following analyses. As a result, data was gathered from 75 children in the debugging task, 69 children in the VPT task and 74 children in the executive functioning assessment.

Procedure

Children were tested individually in a single 45-minute laboratory session. This experiment had six components for participants to complete: (1) a robot introduction and free play session, (2) an algorithm writing task, (3) a VPT task, (4) a prediction task, (5) an executive functioning assessment and finally (6) a debugging task. The order of activities three to six were counterbalanced across participants. The laboratory environment provided the opportunity to manipulate participants' engagement with the robot (i.e., their body position), whilst controlling the physical context as well as the linguistic and perceptual cues given by the experimenter. All sessions were videotaped with audio for behavioural coding. A video camera was positioned and moved throughout the laboratory session to provide a view of the participant, the robot programming board and the floor map.

Robot Introduction and Free Play

Children played with a small, tangible robot named Cubetto. Cubetto was created by Primo Toys (www.PrimoToys.com). Its design avoids textual and numerical language thus making it suitable for pre-literate children. Moreover, children can use the playset to learn about coding concepts through hands-on learning and without the use of a screen. Cubetto comes equipped with an interface board, a range of function tokens (forward, right, and left turn functions) and a colourful floor map (see Figure 4.1). Children can navigate the robot around the map by placing the desired tokens in the interface board and pressing the 'Go' button.

Robot Demonstration. At the start of the testing session, children were introduced to Cubetto and its movement tokens. As the primary experimenter, I explained that the robot was unable to move across the floor map on its own and instead needed to follow a sequence of instructions (i.e., coloured tokens). Children's attention was drawn to Cubetto's smiley face and the embossed arrow on top of the device and they were informed that these elements would help them identify *"the way Cubetto is facing."* Next, I introduced the interface board, likening it to a remote control. Once participants understood the function of the interface board, Cubetto's movement tokens were then introduced. I then worked with each child to test each token as they observed that each colour gave the robot a different command (i.e., green = one square forwards, red = right turn, yellow = left turn).

Children in the embodied learning condition were encouraged to complete the token movements with Cubetto (i.e., turn their body left/right or move forward in the space). Children in the other conditions (incongruent and congruent programming) remained seated. All children were given place cards to remind them of the function of each token (see Figure 4.1). After testing each of the movement tokens, the children completed a practice trial. For all children, the Cubetto robot was placed in the same location on the map, and they were tasked with programming Cubetto to move to a predetermined location (this required three 'forward' tokens, with no turn tokens needed). If a child needed guidance to complete the practice trial correctly, I prompted them to count the number of squares Cubetto needed to move and referred to the place cards to remind them of the three token functions.

Free Play. Research has shown that young children learn through repetition and often play with games or watch media content repeatedly when learning (Crawley et al., 2002; Hintzman, 1976; Karpicke & Roediger, 2008; Mares, 2006; Santer et al., 2015). Thus, after being introduced to Cubetto, participants were then given 5 minutes of free, unstructured play time. During this time, children were encouraged to program the robot to move to different locations on the floor map (of their choosing). This allowed them to familiarise themselves with the robot, the interface board, and the movement tokens without being given a set goal. If children asked for help, I prompted them to try their best to see where Cubetto finished on the map. I also reminded them that they could fix their tokens afterwards if Cubetto did not finish where they wanted.

Algorithm Writing Task

This task involved four trials of increasing difficulty. On each trial, Cubetto was placed on a specific spot by the experimenter and the child was given a goal location they needed to program Cubetto to reach. Difficulty was determined by a combination of the number of tokens used (i.e., more tokens = harder trial) and the type of tokens used (only forward tokens = easy, one turn token = medium difficulty, multiple turn tokens = hard). Table 4.1 provides a breakdown of each algorithm writing trial.

Table 4.1

Algorithm Writing Trials.

Trial	Description
1	Forward tokens only (Map 1).
2	Most efficient route included one turn token and several forwards tokens (Map 1).
3	Also using one turn token, however, children were told they need to avoid obstacles whilst moving the robot. Avoidance of obstacles left one available route for children to follow (Map 2).
4	Multiple turn tokens, moving the robot along route whilst avoiding obstacles on the map. Again, this left one available route (Map 2).

To begin the algorithm writing task, I removed the original floor map (used during the free play session) from the space and replaced it with a new Cubetto map (Map 1). I explained to each participant, *"Now Cubetto wants to explore the ocean! Shall we help him/ her? You have your cards to help you remember what Cubetto's tokens do. Remember, you need to put all the tokens in before you press GO."* I then presented children with the first trial, *"Cubetto is over here on this island (gestures to map location), but he/she wants to go on an adventure over here (gestures to map location) to see the fish. Can you help him/ her get there?"*

The positioning of the participant in relation to the Cubetto robot varied between the three conditions outlined above. During the 'Incongruent programming' condition, the participant's bodily orientation did not align with the Cubetto robot. However, in the 'Congruent programming' condition, the robot was always positioned in front of the child (thus aligning their orientations). Finally, in the 'Embodied learning' condition, children were instructed to stand in Cubetto's place on the map, thus embodying the robot's position and orientation. For trial one I explained, *"Cubetto wants to go on an adventure from the island, where you are, to over here (gestures to location) to see the fish. What do you and Cubetto need to do to get there?"* Although the orientation of the participant changed across conditions, the location of the robot on the map remained the same across all conditions. Thus, children completed the same algorithm writing tasks however experimental conditions manipulated their orientation and level of embodiment.

Verbal reflection was encouraged throughout the task. I encouraged reflection by pausing the Cubetto task and prompting children to verbalise their reflective thoughts following errors. An example response given to children following an incorrect trial would be *"Oh no, Cubetto did not make it! What do you need to change to help Cubetto get to the fish?"* Once the child had voiced an alternative solution, Cubetto was placed back onto the map in its original location for children to make another programming attempt. Reflection is discussed in the wider literature as a tool for promoting higher order thinking and learning (Boud et al., 1985) and therefore encouraging feedback following incorrect responses was thought to potentially aid programming performance and understanding. To prevent children from re-running the same incorrect algorithm without reflection, I placed a small cover over the "Go" button on the interface board to prevent children from pressing it repeatedly. Once children had paused to think about their sequence of tokens, the cover was removed (even if children were against changing any of their tokens).

Participants were given three attempts to complete each trial. If they were successful at getting Cubetto to the correct location on the map, they then progressed to the next trial. If they were unsuccessful on the third attempt, the algorithm writing task was ended, and they were directed to the next activity. For each completed trial, children could score up to three points depending on how many attempts it took for them to complete the trial. Higher scores were indicative of better performance. Children were given three points if they completed the trial on their first attempt, two points for their second attempt, one point for their third and zero points if they did not complete the trial. Thus, each child received a score out of three for each of the four algorithm writing trials and could achieve a maximum score of 12 in this task.

Visual Perspective Taking Task

Setup for this task included a table, chair, display tray and a set of coloured blocks for both the primary experimenter (myself) and the participant (see Figure 4.2). At the start of the task, the participant and I each sat at separate tables, facing one another. On each table was a display tray, placed vertically, with a red block and a yellow block. In front of me, one block was close in front of my body, and the other block was positioned far from my body. The same blocks were given to the participant however the blocks were stacked (so as not to bias children when positioning the blocks in their display tray).

Figure 4.2

Set Up for the Visual Perspective Taking Task.

Note: Participant is positioned to the left of the image, and the Experimenter is positioned to the right.

Before the test trials began, I completed a practice trial with each child. Firstly, the task was introduced to the participants,

> *"I have a tray with some coloured blocks on it, and look, you have coloured blocks that match. In this game, you need to imagine that you're sat in my chair, in my body, and that you can see through my eyes."*

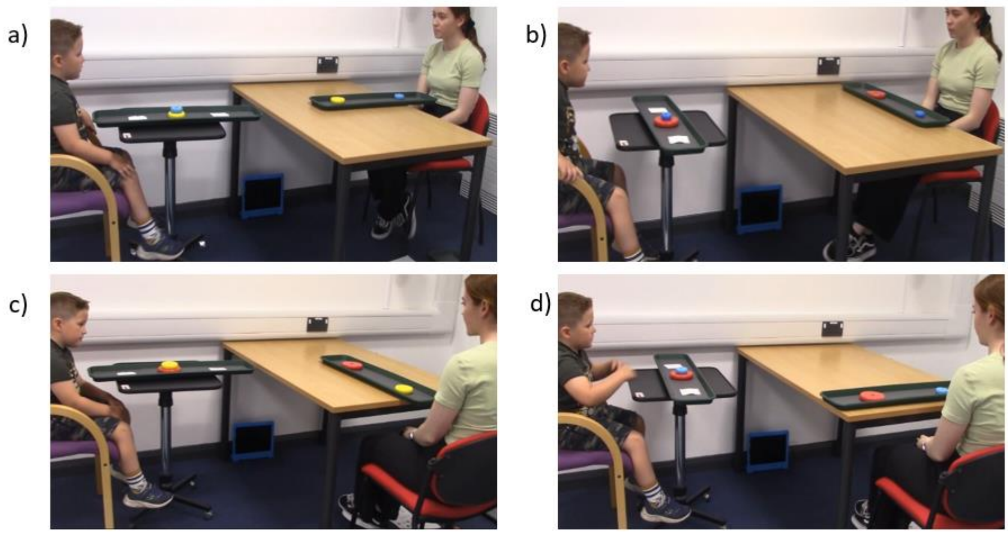
While gesturing to each of the blocks, I provided scaffolding to help children complete the practice trial. *"Do you see how the yellow one is close to me, and the red one is far away? Can you put the blocks how I see them through my eyes?"* To give a correct response, participants needed to place the yellow block close to them and the red block further away. If participants placed the blocks in the incorrect position (i.e., representing how they see the display rather than how I, the experimenter, viewed it), then they were prompted to try again, and the same verbal prompt was repeated. If needed, this prompt was given a maximum of three times. If a participant did not give the correct response after all prompts were given, the secondary experimenter noted this, and test trials began. Data from children who failed this practice trial were excluded from later analyses.

This task contained 12 trials. For half of these trials, children needed to mentally rotate the array of blocks 180-degrees as I was sat on the opposite side of a table. For the remaining trials, children needed to mentally rotate the array 90-degrees as I sat at the side of a table.

These differing orientations were included to manipulate the difficulty of the perspective taking task. Developmental research from Huttenlocher and Presson (1973) found that children made more egocentric errors on a perspective taking task when displays were rotated to 180-degrees than when there was a 90-degree difference. In addition to the orientation of my perspective and body, the positioning of the blocks on the table was manipulated. Blocks were either placed virtually in front of me, or horizontally to either side. Thus, the 12 perspective taking trials were broken down as follows: 3 near-far 180-degree trials (see Figure 4.3a), 3 left-right 180-degree trials (see Figure 4.3b), 3 near-far 90-degree trials (see Figure 3c), 3 left-right 180-degree trials (see Figure 4.3d). When scoring perspective taking performance, each child received a point for each correct trial (achieving a maximum of 12 points).

Figure 4.3

Example trials from the VPT task.



Note: Image (a) x3 near-far 180-degree trials, image (b) x3 left-right 180-degree trials, image (c) x3 near-far 90-degree trials and image (d) x3 left-right 90-degree trials. Child is positioned to the left of the image, experimenter to the right.

This task was an adaptation of Tversky and Hard's (2009) spatial perspective taking task, with stimuli modelled after the original study (i.e., a display that included a model and two objects: colourful circular blocks, see Figure 4.2). In this study, rounded blocks were used so participants did not need to consider the orientation of the shape. Additionally, trays were used to present the array of blocks to the participant. This was to avoid any additional body language cues from myself as the primary experimenter (i.e., reaching across the table to place a block). Children were not required to provide a verbal or written response, instead, children presented their answers visually using coloured blocks matching those in front of me (the experimenter).

Prediction Task

The aim of this task was to assess whether children were able to view an algorithm (a sequence of programming tokens) and predict where the robot would end up on the floor map. They made their prediction by placing a star cut-out on a square of the floor map (see Figure 4.4). Children could complete up to six trials of varying difficulty. Difficulty was determined by a combination of the number of tokens used (i.e., more tokens = harder trial) and the type of tokens used (only forward tokens = easy, one turn token = medium, multiple turn tokens = hard). Children were not allocated to an experimental condition in this task. Thus, instructions for embodiment were not provided and all children were positioned in the same position at the bottom of the robot floor map (see Figure 4.4). If children completed a trial incorrectly, they could attempt the next prediction trial. However, if children answered two trials back-to-back incorrectly, the prediction task was ended, and they were directed to the next activity. When scoring prediction performance, each child received a point for each correct trial (achieving a maximum of 6 points).

Figure 4.4

Example of a Prediction Task Trial (Trial 1, Forward Tokens Only).

Note: Child's answer is illustrated by the placement of the yellow star (right side of the image; this is a correct prediction).

Debugging Task

This problem-solving task modelled debugging tasks used in previous research (see Strawhacker & Bers, 2019, Solve It tasks). However, unlike previous research, this debugging task was done using the robot itself, instead of being a paper-based task. During this assessment, participants were required to fix an algorithm that did not work. As the primary experimenter, I identified a location on the map and programmed the robot to move there. Children watched as the written algorithm did not get Cubetto to the desired location. Participants were then asked to debug and fix the sequence of tokens. This task had six trials for participants to complete (2 easy, 2 medium and 2 hard). The difficulty of each trial was determined by the number of incorrect tokens children needed to correct (i.e., 1, 2 or 3). Children were given two attempts to pass a trial. If they failed at the second attempt, they did not progress to the next trial. For each completed trial, children could score up to 2 points depending on how many attempts it took for them to complete the trial. Higher scores were indicative of better performance. Children were given 2 points if they completed the trial on their first attempt, 1 point for their second attempt, and 0 if they did not complete the trial. Thus, each child received a score out of 2 for each of the six debugging trials and could achieve a maximum score of 12 in this task.

Executive Functioning Task

The Minnesota Executive Function Scale (MEFS; Apple.com; Carlson & Schaefer, 2012) measures cool executive function skills, including working memory, inhibitory control, and cognitive flexibility. This measure was originally adapted from the Dimensional Change Card Sort task (Zelazo 2006) and is shown to be a valid measure of executive functioning for children aged 2 -13 with good test-retest reliability (intraclass correlation = 0.93; Beck et al., 2011; Carlson & Schaefer, 2012).

On screen, children viewed two boxes with target cards on them (see Figure 4.5). They were then given a rule to follow when sorting cards into these boxes: sort by shape, or sort by colour. There were seven levels of varying complexity, and children completed each level in two parts. In Part A, participants were instructed to sort cards on one dimension (e.g., colour), before the sorting rule was then switched in Part B (e.g., sort by shape). At the higher levels, participants were required to adapt to rule changes multiple times and rule changes got more complex.

The participant's age determined their starting level (e.g., 4.5-year-olds start at level 4). To pass each level, children must have accurately allocated cards on at least 4 of 5 trials for both parts A and B (both rule sets). If the participant passed the recommended starting level, the program continued up the scale. However, if the participant was unable to correctly complete 4 out of 5 trials on either part (A or B), the program dropped back to the previous level. The MEFS application calculated a total score which was used for subsequent analyses in this study. The total score (0 – 100) was automatically calculated using an algorithm that takes both accuracy and response time into account. A score near 100 illustrated a participant was both accurate and fast.

Figure 4.5

Screenshot from the MEFS Assessment App.

Note: Acquired from https://apps.apple.com/us/app/mn-executive-function-scale/id967184252.

Embodiment Scoring

As described above, children were allocated to one of three experimental conditions (Incongruent programming, Congruent programming or Embodied learning). Although each condition was designed to encourage different levels of embodiment, children's tendency to engage in embodied actions varied across groups. For instance, children in the embodied learning condition were not required to carry out the prescribed embodiment behaviours to proceed with the programming tasks. Moreover, children were not prevented from engaging in physical movement in the congruent and incongruent programming conditions. As a result, children's actual engagement in embodiment varied across conditions in ways that were not always closely aligned with our intended manipulation. To account for this, videos of the programming tasks were individually coded, and participants each received an 'embodiment score'.

For each trial within the algorithm writing task, participants were given an embodiment score (how much children engaged their body within the programming environment), whereby higher scores were indicative of greater embodiment (see Table 4.2). Average embodiment scores across all trials were then calculated to avoid confounding embodiment scores with the number of trials completed. A random sample of video recordings (i.e., 25%, n = 19) were selected for coding by a second researcher to ensure consistency in scoring. Cohen's Kappa values were calculated for each of the four algorithm writing trials. Analyses showed that there was almost perfect-to-perfect agreement (Landis & Koch, 1997) between the two coders on Trial 1 ($\kappa = 0.84, p < 0.001$), Trial 2 ($\kappa = 0.92, p < 0.001$), Trial 3 ($\kappa = 1.0, p < 0.001$) and Trial 4($\kappa = 1.0, p < 0.001$).

Table 4.2

Rubric Used for Embodiment Coding During the Algorithm Writing Task.

Score	Description	Level of embodiment
0	Participant is sitting/standing with a visual perspective that does **not** align with the robot (incongruent programming).	Participant's physical orientation is not directly related to the robot's position or movements. Thus, engagement relies more on visual and cognitive understanding rather than physical movement and alignment.
1	Participant is sitting/standing with a visual perspective that **does** align with the robot (congruent programming).	Participant's physical orientation is directly related to the robot's position at the start of the programming task. Alignment may enhance their spatial awareness and physical connection with the robot.
2	Participant is standing stationary on the map but is twisting their	Participant's engagement becomes more overt as they physically twist their body. By

	body (in one location on the map) whilst programming.	actively engaging their body, they embody the programming process more fully, incorporating both cognitive and physical elements into their experience.
3	Participant is walking their determined route on the floor map whilst programming (changing locations on the map; embodied learning).	This level involves the most dynamic bodily engagement as the participant physically moves through the programming environment. Participants embody the robot's movements in a more direct and immersive way, leading to complete child-robot alignment and integration of physical, cognitive and spatial aspects of the task.

Statistical Analyses

A Multivariate Analysis of Covariance (MANCOVA) was identified as most appropriate for the experimental design used in this study. A MANCOVA is particularly useful when studies have multiple dependent variables (algorithm writing, debugging, prediction) and want to assess how they are influenced by experimental conditions (Incongruent programming, Congruent programming, and Embodied learning) while controlling for potential confounding variables (age, executive functioning and VPT). Additional correlation analyses were conducted to explore the relation between variables (for example, programming performance and VPT).

All statistical analyses were performed using SPSS (version 28.0; IBM, 2021). A *p*-value of .05 was used to determine statistical significance and correlation analyses were two-tailed. Visual inspection of plots revealed that programming and embodiment scores were positively skewed and not normally distributed; this was supported by Shapiro-Wilk normality testing which was significant for VPT ($W = 0.93$, $p = 0.001$), programming ($W = 0.9$, $p < 0.01$), embodiment ($W = 0.91$, $p < 0.01$), and executive function measures ($W = 0.94$, $p = 0.003$).

Based on this non-normality, non-parametric Spearman's Rank correlation analyses were used to explore relations between variables of interest (e.g., performance on programming tasks and VPT; Research Question 1) and extensive consideration was given to alternative analysis techniques to explore the effects of experimental condition on outcome variables

(Research Question 2). However, a MANCOVA emerged as the most suitable approach given the complexity of the study design. Despite the non-normality of the data, no alternative analysis method could adequately address the multifaceted nature of the research questions and objectives. Furthermore, the decision to utilise a MANCOVA was justified by its resilience to departures from normality when sample sizes are large (Blanca, 2017) The sample size for this study consisted of 78 participants, surpassing the recommended size of 69 as determined by G*Power analysis.

Results

Descriptive Statistics

Children's scores on the algorithm writing task ranged from 0 (no trials answered correctly) to 12 (all trials answered correctly on their first attempt). On average, children scored 4.65 (SD = 3.42, n = 78) on this programming task. Scores on the prediction task ranged from 0 to 9 (all trials correct) and the mean score was 1.69 (SD = 1.60, n = 78). Scores on the debugging task ranged from 1 to 12 (all trials correct on the first attempt) and children's average score was 6.40 (SD = 3.54, n = 75). Finally, scores on the Visual Perspective Taking (VPT) task ranged from 2 to 12 (all trials correct) and the mean score was 6.13 (SD = 2.34, n = 69).

Relation Between Programming, Visual Perspective Taking and Executive Functioning

Firstly, correlation analyses were performed to investigate possible associations between programming outcomes (i.e., scores on the algorithm writing task, the prediction task, and the debugging task) and other possible correlates (i.e., VPT, executive functioning and age). Results from Spearman's Rank correlation analyses are presented in Table 4.3.

Table 4.3

Results from Spearman's Rank Correlation Analyses.

Variables		1	2	3	4	5	6
1	Algorithm Writing	1					
2	Prediction	0.64**	1				
3	Debugging	0.68**	0.60**	1			
4	Perspective Taking	0.18	0.32**	0.16	1		
5	Executive Functioning	0.40**	0.37**	0.37**	0.23	1	
6	Age	0.70**	0.59**	0.70**	0.22	0.46**	1

Note: ** Correlation is significant at the 0.01 level (2-tailed).

The Spearman's Rank correlation analyses revealed several associations between programming related outcomes and various factors in this study. Notably, algorithm writing task scores exhibited significant positive correlations with both prediction scores and debugging scores. Similarly, prediction scores showed significant positive correlations with debugging scores. VPT displayed moderate associations with prediction scores, however, correlations with programming and debugging were not significant.

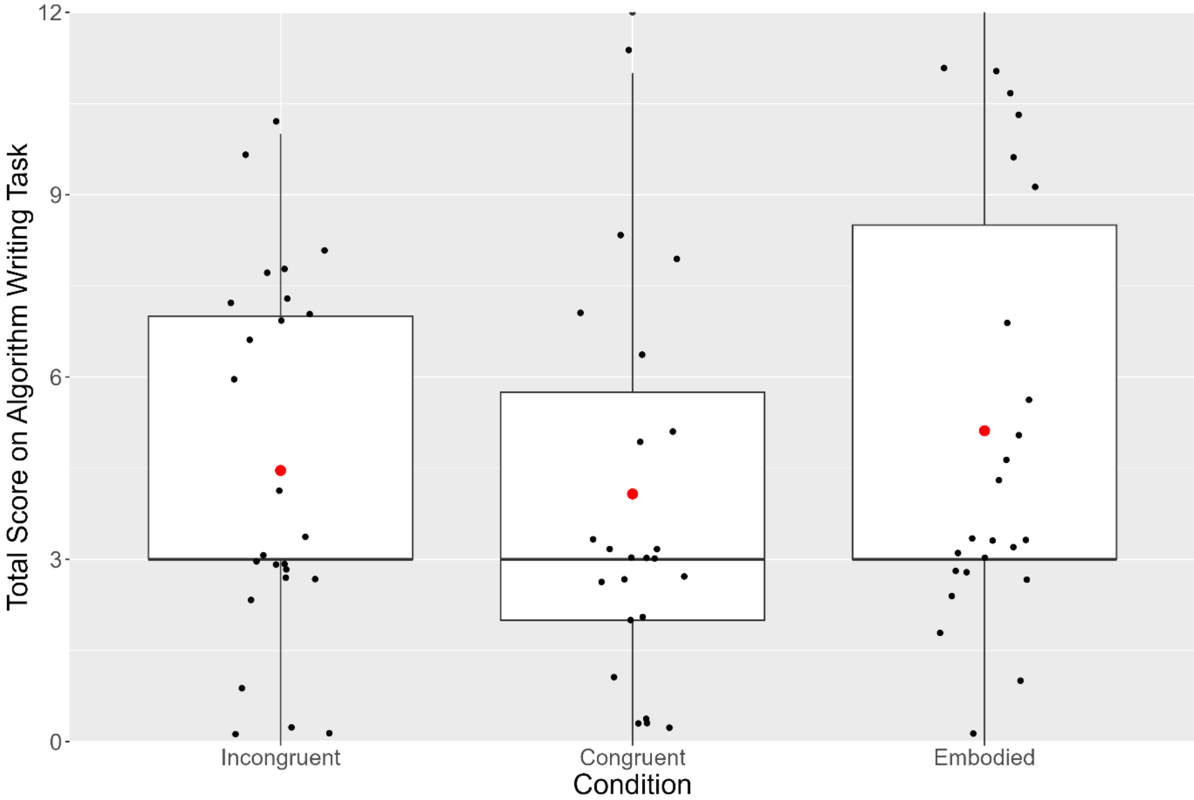
Executive Functioning was significantly correlated with algorithm writing scores, prediction scores and debugging scores. Thus, children with higher executive function scores had higher performance on the three robot and programming-related tasks. Furthermore, age exhibited significant positive correlations with all programming related outcomes (algorithm writing, prediction and debugging scores) and a moderate association with executive functioning.

Embodied Learning and Programming Performance

As a reminder, embodiment condition was assigned for the algorithm writing task, however, embodiment was not encouraged or restricted during the prediction and debugging tasks. Mean and standard deviation scores for performance on the algorithm writing task are displayed in Figure 4.6, grouped by condition.

Figure 4.6

Performance on the Algorithm Writing Task.



Note: Scores have been grouped by experimental condition that was manipulated solely during the algorithm writing task. Condition means are shown in red.

A Multivariate Analysis of Covariance (MANCOVA) was employed to examine the impact of experimental condition (i.e., incongruent, congruent, or embodied learning) on three programming related dependent variables (i.e., performance on the algorithm writing task, the prediction task, and the debugging task). VPT scores, executive function scores and participant age were included as covariates to further isolate the effect of experimental condition on programming outcomes and reduce confounding effects.

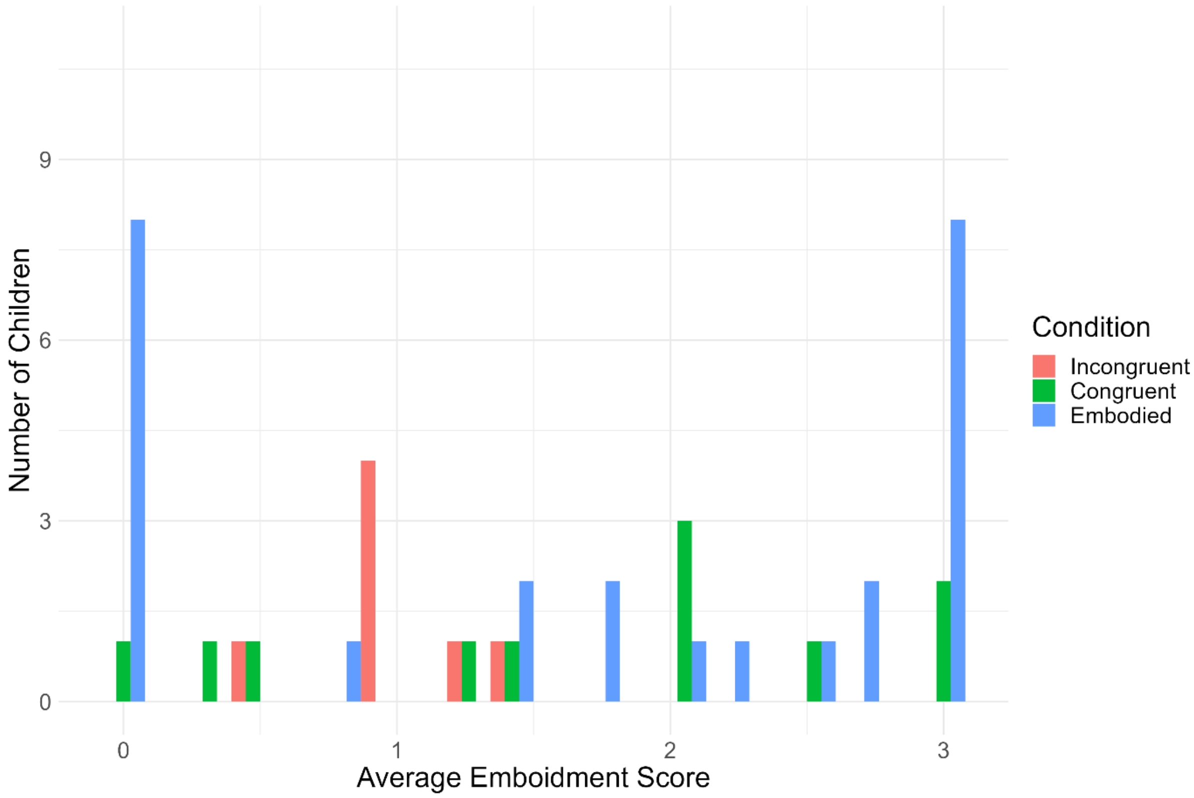
Results showed that the experimental condition children were assigned to did not have a statistically significant effect on the combined programming-related tasks (F (6, 118) = 0.18; p = 0.98; Pillai's Trace = 0.01; partial $\eta 2$ = 0.01). Thus, it appears that embodiment variations between experimental conditions did not lead to significant differences in programming outcomes.

Exploratory Analyses

It was observed that during testing sessions, despite encouragement only within the embodied condition, children's propensity to engage in embodied actions varied across groups during programming. For instance, children were not prevented from independently engaging in embodied methods in the Incongruent and Congruent programming conditions. Likewise, those in the Embodied learning condition were not required to carry out the prescribed embodiment in order to participate. As a result, children's embodiment behaviours while they completed the algorithm writing task were individually coded (as described in the above method section). Observations of variation in embodiment behaviours within conditions were supported by the individualised coding scores (see Figure 4.7).

Figure 4.7

Variations in Average Embodiment Scores During the Algorithm Writing task.



Note: Each experimental condition is shown in a different colour. Higher embodiment scores indicate greater embodiment, while lower scores indicate the child's body orientation (and whether it aligned with the robot's orientation).

To further investigate whether embodied learning methods aided performance on the algorithm writing task, a multiple regression analysis was performed. Embodiment score, participant age, executive function score, and VPT score were entered into a regression equation as predictor variables, with algorithm writing score included as the outcome variable. The test indicated that this model explained 47.8% of the variance in children's algorithm writing scores, $R^2 = .48$, $F(4, 61) = 13.94$, $p < 0.001$. Results indicated that participant age was a strong predictor of children's algorithm writing performance, β = 1.75, $t(61) = 5.22$, $p < 0.001$. The model also showed that embodiment score (β = 0.21, $t(61) = 0.83$, $p = 0.41$), executive function score (β = 0.04, $t(61) = 1.30$, $p = 0.20$) and perspective taking score (β = 0.20, $t(61) = 1.42$, $p = 0.16$) were not significant predictors of children's algorithm writing scores.

The same regression model was used to investigate whether embodied learning impacted performance on the prediction task. Results showed that participant age was a significant predictor of prediction scores (β = 0.55, $t(61) = 2.56$, $p = 0.01$), however embodiment score (β = -0.02, $t(61) = -0.13$, $p = 0.90$), executive function score (β = 0.02, $t(61) = 1.30$, $p = 0.20$) and VPT scores (β = 0.14, $t(61) = 1.59$, $p = 0.12$) were not significant predictors. Similarly, participant age was a significant predictor of debugging scores (β = 2.11, $t(61) = 5.80$, $p < 0.001$), however, embodiment score (β = -0.01, $t(61) = -0.04$, $p = 1.0$), executive function score (β = 0.02, $t(61) = 0.78$, $p = 0.44$) and VPT scores (β = -0.04, $t(61) = -0.26$, $p = 0.80$) were not.

Discussion

This laboratory-based study aimed to explore whether Visual Perspective Taking (VPT) skills were related to children's performance on a selection of programming tasks (i.e., algorithm writing, prediction and debugging) and what role executive functioning plays in VPT abilities. I also investigated whether embodied learning techniques aided algorithm writing with a tangible robot. The following discussion will explore this study's results and will compare these findings to those from previous research. I will then highlight the implications of these findings and examine the strengths and limitations of the current study design.

Does Performance on Programming Tasks (i.e., Algorithm Writing, Debugging, Prediction) Relate to Performance on Visual Perspective Taking and Executive Functioning Tasks?

According to Flavell et al., (1980), level 2 perspective taking requires children to adopt the visual perspective of another observer. Thus, children must be able to imagine how an environment looks from another position within an environment (i.e., what is near, far, left, or right). While there is some disagreement in the literature about when level 2 perspective taking skills begin developing, it is agreed that this occurs during childhood (rather than infancy) and continues to improve in adolescence (Drumontheil et al., 2010). Thus, perspective taking skills are likely to vary amongst primary school aged children.

As educational robotics incorporate spatial movement within a child's physical environment, I proposed that level 2 perspective taking skills may be important for the successful programming of these devices. For instance, children must write an algorithm for the robot to follow, and this may be more difficult when the robot's spatial orientation does not align with the programmer's as this must be corrected mentally as they pre-plan their algorithm. Thus, in this study, children not only completed programming tasks with a robot, but they also completed an assessment of VPT. Contrasting my previous predictions, the results from this study suggested that algorithm writing skills were not significantly related to children's perspective taking skills. Instead, results showed that only age was a significant predictor of success in algorithm writing trials. These findings suggest that perspective taking abilities may not play as big of a role in programming abilities as first thought. Consequently, the intervention study presented in Chapters 5 and 6 did not further explore the role of VPT.

Readings of previous studies also suggested that executive functioning skills may play an important role in VPT (Diamond et al., 2002; Frick & Baumeler, 2017). For instance, when children are presented with a conflict between two perspectives in the same physical environment, children's own perspective (the conflicting information) must be ignored for them to be successful at imagining the alternative perspective. This would require inhibitory control skills. Thus, I proposed that performance on a VPT task would be associated with performance on an executive functioning task. This proposal was supported by findings from Frick and Baumeler (2017) who found that inhibitory control significantly predicted VPT in a sample of 6-year-old children.

The results of this study did not support these past findings as no significant relation was found between performance on a VPT task and a measure of executive functioning. Furthermore, performance on an algorithm writing task was not significantly related to executive functioning in this sample of children aged 4 to 7 years old. It is possible that these results did not replicate those from previous research due to the measure of executive functioning used. For example, Frick and Baumeler used the 'Fruit Stroop Task' to specifically target inhibitory control skills. The executive functioning measure used in this study (the Minnesota Executive Function Scale; Carlson & Schaefer, 2012) was a collative measure of cool executive function skills, including working memory, inhibitory control, and cognitive flexibility. Future research attempting to explore correlations between executive functioning and perspective taking could instead employ specialised assessments of independent executive function skills (i.e., inhibitory control).

Although no associations were found between algorithm writing, VPT and executive functioning, correlation analyses did show that all three programming-related assessments were related. Thus, even though the three programming assessments used in this study targeted different computational thinking skills (i.e., algorithm writing, debugging and prediction skills) the results of this study provide evidence for the coherence and validity of these programming assessments. This suggests that these assessments captured different facets of computational thinking rather than measuring entirely distinct constructs. Consequently, these debugging and prediction tasks were adapted for use in the robotics intervention presented in Chapters 5 and 6.

What Role Does Embodied Learning Play in Programming Performance when Accounting for Visual Perspective Taking and Executive Functioning Abilities?

This study applied embodied learning principles to learning with an educational robot. Previous literature suggested that aligning the orientation of a moving object with the participant's viewpoint could improve the participant's ability to navigate the object as they no longer needed to mentally correct the incongruent perspective (Cho et al., 2017). Thus, I proposed that embodied learning may improve children's perception of robot movements by aligning their visual perspectives, thus helping them write accurate algorithms. In an embodied learning condition, children were encouraged to actively engage in full body locomotion on a floor map as if they were the Cubetto robot. As children's movement and orientation were congruent to the movement of the Cubetto robot in terms of changing direction (turning left or right) and moving forwards, it was proposed that this could help children map their embodied experiences onto programming tasks to potentially improve task performance.

The results of this study did not support this notion. Instead, I found no significant effect of embodiment (be that condition or embodiment score) on children's programming outcomes. Thus, embodied learning methods did not improve children's performance on an algorithm writing task when learning with a Cubetto robot, nor did embodiment aid later performance on prediction and debugging tasks. These findings seem to suggest that embodied learning with robotics may not be as beneficial as first thought, however, since data collection for this study took place, one study (Kwon et al., 2022) has found that embodied learning methods can be used to improve children's computational thinking skills whilst using educational robotics.

In Kwon and colleagues' (2022) research, children (aged 6 to 8 years) repeatedly engaged in full-body learning activities in five (35 to 45 minute) sessions with Bee-bot robots. During these activities, children were instructed to solve a series of path-finding problems, either by programming a Bee-bot directly as an observer outside the map, or by acting as the bee and moving across the map. Kwon et al., found that these embodied learning experiences improved children's spatial awareness and perspective taking in computational thinking tasks that required different levels of mental rotation. For instance, if the child and the Bee-bot faced the same direction, the child could direct the robot's next steps intuitively (i.e., a right turn was the same for both them and the robot). However, more difficult tasks included the robot and the child facing opposite directions which required perspective taking when giving directions. These researchers found that after embodied learning activities, children's ability to take the robot's perspective improved by about three times. Although my study found no significant effects of embodiment on children's learning of programming, findings from Kwon et al., suggest that embodied learning may still be an appropriate method for children learning with educational robotics. However, it is worth noting that there are several methodological differences between these studies that may explain the difference in findings between these studies.

Firstly, children in the classroom-based study by Kwon et al. completed multiple programming sessions (with or without embodiment) before being assessed on computational thinking measures. In contrast, children in this laboratory study were given five minutes of free play with the Cubetto robot before completing test trials. This was due to the time restrictions associated with laboratory testing, but findings from Kwon et al. suggest that, in future research, longer learning sessions with the robot would emphasise differences caused by embodied learning.

Secondly, it is worth noting differences in assessment modalities between this study and research from Kwon and colleagues. Kwon et al. administered computational thinking assessments (i.e., of prediction and debugging skills) that were paper-based rather than robot-based. In my laboratory study, children were assessed using the Cubetto robot. This meant that children in the embodied learning condition were employing embodied learning methods as they completed the algorithm writing task. Upon reflection, this may have been difficult as the trials progressed as these children were not positioned next to the interface board when they planned their route and programming sequence. Instead, they were in the middle of the robot programming map, walking their desired route. Due to the size and weight of the interface board, the board remained on a table to the side of the map. In several embodied learning sessions, I watched as children walked a correct route, spoke the steps and tokens needed out loud correctly, but then forgot their planned algorithm once they had stepped off the robot floor map, and back to the interface board. It may be that for embodied learning to be successful in the moment, children need to be scaffolded to write their algorithm in sections which would thus reduce demands on their working memory.

A final notable difference between this study and Kwon et al's research is the age range of the participating children. This study employed a sample aged 4 to 7 years, in comparison to their sample of children aged 6 to 8 years. The simultaneous demand of physical movement and cognitive engagement in embodied learning may overwhelm younger children, thus making it difficult for them to plan and execute algorithms effectively. Providing scaffolding and segmenting tasks into smaller segments may help alleviate these challenges for younger learners. Future research should continue to explore how embodied learning methods can be used to aid children's understanding of programming concepts when using an educational robot.

Additional Methodological Reflections

One important reflection on the current study design is its complexity, particularly in relation to the three-condition structure. While these conditions were designed to manipulate different levels of embodiment, children's actual engagement in embodied actions varied greatly across the conditions. In retrospect, this variability in how children engaged with the tasks suggests that the manipulation of embodiment was not as tightly controlled as initially intended. If I were to conduct this study again, I would consider simplifying the experimental conditions. Specifically, I would propose focusing on just two conditions: Incongruent Programming and Embodied Learning. This modification would allow for a clearer comparison between a traditional programming condition with limited embodiment and a condition designed to encourage embodied interaction. Additionally, by reducing the number of experimental conditions, I would have been able to allocate more participants to the Embodied Learning group. Given that a notable number of children did not engage in the intended embodiment methods, having a larger sample in this condition would have provided more statistical power to detect the potential effects of embodiment on programming skills.

Ultimately, while the three-condition design provided valuable insights, simplifying the design in future studies would enhance both the clarity of the experimental manipulations and the interpretability of the results. It would also allow for a more focused investigation of embodiment in educational programming tasks, leading to stronger conclusions about its effects on learning outcomes.

Additionally, the power analysis for determining sample size was originally based on the three-condition design rather than the correlational analyses conducted in the study. As a result, the sample size may have been insufficient to detect smaller effects in the correlations between variables. An updated power analysis using G*Power for a correlational design suggested that a sample size of 782 participants would have provided greater statistical power, potentially revealing significant relationships that were not apparent in the current analyses. In future research, a power analysis specifically tailored to the correlational analyses would be important for determining an adequate sample size. This adjustment would ensure that the study is adequately powered to detect even modest effects, thereby increasing the robustness and reliability of the findings.

Although the findings from the laboratory study did not support my initial predictions, this study illustrated the suitability of the Cubetto robot for children aged 4 to 7 years old. Observations made in this study suggested that children enjoyed using the robot. Furthermore, results showed that most children were able to progress through several programming assessments, thus suggesting that children were able to understand how the device worked despite not having used it before. Children's progression on programming assessments, and the absence of ceiling and floor effects, further justifies the use of Cubetto robots for computational thinking and programming education in early primary school classrooms. Consequently, the Cubetto robot was integrated into primary school classrooms during the school-based robotics intervention I present in Chapters 5 and 6. Helping children learn with Cubetto enabled me to advise teachers on how to introduce and use the robot with children aged 4 to 7 years.

Children's performance on the VPT task also suggested that they understood how to complete the task. The perspective taking task used in this study was designed to be developmentally appropriate for the age range used in this study. For example, the experimenter purposefully avoided using positional language like 'left' and 'right' as children and even adults often struggle with this language and associating each word with the correct direction. Rather than having children attempt to identify which coloured block was to the left or right of the model, children were encouraged to place the blocks on how they were visible to the model. Removing the positional language was thought to make the task more accessible for children under the age of 7 (Rigal, 1994).

Conclusion

To summarise, this study investigated whether visual perspective taking (VPT) skills were related to children's performance on a selection of programming tasks (i.e., algorithm writing, prediction and debugging) and what role executive functioning played in VPT abilities. It also investigated whether embodied learning techniques aided algorithm writing with a tangible robot by aligning child-robot orientations. At the point of data collection, research investigating embodied learning within the context of programming and computational thinking was lacking (Kallia & Cutts, 2023). The results of this study did not align with the ideas I proposed at the start of this chapter. Instead, results illustrated that children's performance on programming-related tasks was not related to their performance on a VPT task. Furthermore, children's programming and perspective taking abilities were not related to their executive functioning skills. Finally, embodied learning did not aid algorithm writing abilities in this study. Following these findings, the intervention study presented in chapters 5 and 6 did not further investigate perspective taking and embodied learning. Thus, the school intervention continued to explore skills more traditionally associated with computational thinking (i.e., prediction and debugging). The results of this study provided evidence for the coherence and validity of the programming-related assessments used. Variations in task performance and correlations between tasks suggested that these assessments successfully measured different computational thinking skills rather than measuring entirely distinct constructs. Consequently, these assessments were adapted for use in a large-scale, classroom-based intervention (see Chapters 5 and 6). Finally, while these findings suggested that perspective taking, executive functioning or embodiment did not play a significant role in children's programming success, this study did evidence the suitability of the Cubetto robot for children aged 4 to 7 years old. Observations made in this study were that children enjoyed using the Cubetto robot and were able to understand how to program the device. Thus, following this study, I was able to advise teachers on how to use the robot with children in the intervention study presented in Chapters 5 and 6.

Chapter 5. Exploring the Effects of a Teacher-Led Robotics Intervention on Children's Learning and Beliefs.

Introduction

The next two chapters of this thesis present a school-based robotics intervention, designed for children aged 4 to 7 years old and their classroom teachers. This intervention included three experimental conditions: Intervention+ (with a robotics curriculum for pupils and an education workshop for teachers); Intervention (the same robotics curriculum for pupils, but no workshop for teachers) and Control (pupils completed the robotics curriculum after collection of post-intervention data and teachers were offered the education workshop). Two intervention conditions were included in this study's design to explore the role of the teacher and investigate whether providing additional teacher education (prior to curriculum delivery) impacted pupils' learning outcomes (i.e., computational thinking skills).

Computational thinking (CT) and Programming in Primary Education

Programming and computational thinking (CT) is now a distinctive part of primary school curriculums in Wales, as it is in classrooms across the world (Balanskat & Engelhardt, 2015; Bers 2020; European Schoolnet, 2015; Uzunboylu et al., 2017). Both the Digital Competence Framework (Hwb, 2018) and the New Curriculum for Wales (Hwb, 2024a) emphasise *'science and technology'* as a key area of learning. Within this area, curriculum guidance has emphasised the importance of incorporating CT and programming skills, even at the early stages of primary education. For educators, CT is described as "*a combination of scientific enquiry, problem solving and thinking skills*" (Hwb, 2018). Thus, CT is thought to be a set of skills that can benefit everyone, not just those working in technical roles or children learning with computer technologies (Wing, 2006).

The New Curriculum for Wales highlights several CT skills that teachers can target in early primary school classrooms. These include debugging, prediction and sequencing skills. These ideas are often targeted through programming education as programming education has been shown to have a positive effect on CT skills (Jaipal-Jamani & Angeli, 2017). Within programming education, debugging has been defined as identifying and fixing errors in an algorithm (Bers et al., 2019). Guidance also states that children should learn to *"follow algorithms to determine their purpose and predict outcomes"* (Progression step 2; Hwb, 2024b). For example, a child may use the knowledge they have about the function of different programming instructions to anticipate what an algorithm will do. Finally, sequencing refers to a series of individual steps or instructions ordered to achieve a desired outcome (Brennan & Resnick, 2012). This skill is required as children write algorithms within programming activities. Curriculum guidelines define algorithms as *"Processes or sets of instructions to be followed in calculations of other problem-solving operations"* (Hwb, 2024b).

Educational Robotics for CT, Beliefs and Programming Skills

Earlier studies presented in this thesis (see Chapters 2 and 3) have found that educational robotics (ER) are popular amongst primary school teachers as a method of teaching programming and CT to their pupils. ER are devices that combine hands-on learning with tangible technology. Recent examples include the Bee-Bot (www.tts-group.co.uk), the KIBO robot (Sullivan et al., 2017) and the Cubetto robot (www.PrimoToys.com). Children can operate these devices by creating sequences of instructions (i.e., algorithms) out of simplified movement commands (i.e., move forward, move backwards, turn left, turn right etc.). As ER have been identified as an appropriate learning tool for children in early primary education, past research has found that ER can be used to aid the development of CT skills (see Wang et al., 2023 for meta-analysis).

Debugging Skills

Firstly, researchers have used ER interventions to develop children's debugging abilities. One study (Pugnali et al., 2017) found advantages of using ER (i.e., a KIBO robot) to develop children's debugging skills over a screen-based programming application (i.e., Scratch Jr). This study assessed the impact of a ~15-hour programming curriculum and explored whether learning outcomes varied based on the method of instruction. Results showed that children learning in the ER condition performed significantly better on paper-based debugging tasks at the end of the course than children using the screen-based interface. Furthermore, large scale research from Misirli and Komis (2023) investigated children's (n = 526, aged 4 to 6 years) engagement in debugging practices with a Bee-bot robot. They found that teaching children programming skills using a Bee-bot robot encouraged children's development of CT skills, including debugging. These findings not only suggest that ER can be used to engage children as young as four in debugging activities (Bers et al., 2014), but that these tangible methods improve debugging skills more than screen-based programming tasks. However, it is worth noting that neither study employed control groups thus limiting the ability to draw definitive conclusions about the causal relationship between the robotics interventions and the observed outcomes.

Prediction Skills

Research has also investigated how ER can be used to help children develop their prediction skills. One study (Slangen et al., 2011) explored children's (aged 10 to 11 years) learning as they worked with Lego Mindstorms NXT robots. During programming lessons, children were presented with algorithms of increasing difficulty and were tasked with predicting the robot's behaviour based on these sequences. Researchers used qualitative methods to observe children's learning and concluded that experience with ER challenged children to predict, hypothesise and then test their assumptions. These observations suggested that ER can be used as a tool to develop children's prediction skills through tasks that encourage children to predict the outcome of algorithms. However, further research is needed to investigate the relation between ER learning and prediction abilities using quantitative methods, a control comparison group, and samples of younger children. Thus, this study investigated the impact of a 6-week robotics program on prediction skills in a sample of children aged 4 to 7 years. It also employed a control group and assessed children's prediction skills using quantitative assessments.

Sequencing Skills

Finally, research has also shown that ER can be used to develop children's sequencing skills. For example, one study (Strawhacker et al., 2013) investigated whether programming interventions across three different interfaces impacted performance on a sequencing assessment. Kindergarten children (n = 36, aged 4 to 5 years) were allocated to one of three conditions: tangible interface (i.e., robot learning), graphical interface (i.e., screen-based learning) or hybrid interface (i.e., combined robot/ screen-based learning). They found significant differences between the tangible and graphical interface conditions after the intervention, whereby children in the robot condition answered a higher percentage of sequencing assessment trials correctly (73.3%) compared to children in the screen-based learning condition (12.5%). Supporting these findings, another study (Kazakoff et al., 2013) concluded that ER could improve children's (aged 4 to 5 years) sequencing abilities after just a weeklong intervention (relative to a control 'learning as normal' group who continued with their regular curriculum). The findings of these studies suggest that children as young as 4 years-old can improve their sequencing abilities through ER activities and that these tangible interfaces may provide better learning outcomes than screen-based interfaces.

Beliefs

Alongside CT skills, past research has also assessed the effect of ER interventions on children's beliefs about programming and robotics. Beliefs towards these topics are thought to be multidimensional, involving a combination of self-efficacy beliefs (how confident a child is in their ability to use a robot or complete programming tasks) and beliefs in the value or importance of these topics (Eccles & Wingfield, 2002). It is important to investigate children's beliefs in this area as past research has found children's early STEM (Science, Technology, Engineering, Mathematics) beliefs can impact later learning outcomes and continued interest (Simpkins et al., 2006).

Past evidence on whether ER exposure improves children's value and self-efficacy beliefs is mixed. On the one hand, some have found that using robotics in education can improve children's beliefs. For example, one study (Zviel-Girshin et al., 2020) implemented a robotics curriculum (with lessons once or twice a week for seven months) and surveyed 84 children (aged 5 to 7 years). Researchers reported that children felt confident in their ability to use robots and held positive beliefs about continuing learning with robotics in the future. On the other hand, others have reported negative effects on children's beliefs (Hussain, 2006; Leonard et al., 2016). For example, Leonard et al, (2016) found that 10 to 11-year-old children's self-efficacy decreased following a robotics program that lasted 6 to 10 weeks. It is important to investigate whether ER exposure in primary school may positively or negatively impact children's self-efficacy and value beliefs as these changes may have a long-term impact on children's perceptions of programming and robotics. This current study aimed to do just that. As part of this school-based intervention, children's self-efficacy and value beliefs were assessed (using Likert-scale style responses) pre- and post-intervention to investigate potential changes following controlled robotics exposure during a six-week program.

Transfer of Programming Skills

CT is recognised as a versatile thinking competency that is important for being successful in all areas of STEM learning (Grover & Pea, 2018). Thus, CT is not just about learning how to use a particular programming tool but is instead about developing a set of problem-solving skills and strategies that can be applied across different situations. This study investigated whether children who completed the 6-week Cubetto robot intervention showed signs of spontaneous transfer of CT skills. For example, it explored whether children transferred skills developed through learning with the Cubetto to a new programming language. To do so, children in this study completed a programming assessment using the Lightbot Jr programming application (Apple.com). The Lightbot Jr app was chosen as it shared similarities with the Cubetto programming language. For example, while using the app children were required to manoeuvre the new robot to an end goal using simplified programming instructions. These instructions (i.e., forwards, left and right turn) were presented to children using image tokens rather than text, thus making the application suitable for pre-literate children.

To transfer knowledge effectively, it is essential that children develop a flexible mental representation of the education content and identify the similarities between previously learned solutions and new problems (Barr, 2013; Fisch et al., 2005). This would enable them to transfer CT and programming knowledge from a physical programming language to a virtual programming language. In their conceptual paper, Kallia and Cutts (2023) note that moving from physical programming to virtual programming is a large conceptual step for children, but that combining physical actions with symbolic instructions to control an object (i.e., a robot) can help bridge the gap between physical and virtual learning environments. Thus, this study investigated whether children could transfer programming knowledge from an interactive Cubetto robot to Lightbot Jr, a virtual programming environment.

Evaluating Past Educational Robotics Interventions

The final section of this introduction evaluates past robotics interventions with primary aged children. I now describe previous studies' choice of educational robotics, intervention administration, sample age, experimental design, and sample size. I also explore the potential limitations of past research and outline how the current intervention study aimed to improve upon them.

Choice of Educational Robotics

Literature reviews have highlighted several types of user interfaces used in research to develop CT in primary school children. Bakala and colleagues' review (2021) found that researchers used either robotics with physical buttons on top of the robot, tangibles (i.e., interfaces whereby algorithms are written using coloured materials or blocks) or hybrid interfaces (i.e., a combination of tangible and graphical interfaces that utilise screen-based programming languages). They highlighted that Bee-bot (with buttons on top; www.tts-group.co.uk) was the most popular device used in empirical studies aimed at exploring how robot-mediated activities could promote the development of CT in children under the age of 8 years old. This aligns with findings from my previous survey study (see Chapter 3) which found that Bee-bot was the most popular robot used by Welsh primary school teachers inside the classroom.

However, in earlier chapters, I have explored the relative advantages and disadvantages of certain user interfaces, as experienced by primary school teachers (see Chapters 2 and 3). Teachers have found that robots with physical buttons (i.e., Bee-bot robots) do not generate the structured, visual algorithm needed for skills like debugging and algorithm prediction. This kind of user interface can make programming cognitively demanding due to the high load on working memory, whereby the child is expected to hold their programming sequence in their memory (Bakala et al., 2021). The accounts of teachers supported this notion as they found that children would quickly forget what they had instructed the robot to do (see Chapters 2 and 3). Bakala and colleagues (2021) also suggested that these interfaces therefore limit programming activities to sequencing only tasks. In contrast, Cubetto, the programmable robot developed by Primo Toys (www.PrimoToys.com) facilitates all stages of programming by providing children with an interface board in which they place tangible movement tokens. Thus, Cubetto's interface board may overcome Bee-bots' problems of program visualisation by allowing children to physically manipulate their tokens in its large interface board. In doing so, children no longer need to remember the commands they have given to the robot. Instead, they have a clear view of their algorithm to help them predict the outcome of their sequences and detect errors in their program.

Furthermore, the laboratory study presented in Chapter 4 engaged children aged 4 to 7 in several programming and CT tasks with a Cubetto robot. It appeared that children enjoyed playing with the Cubetto robot and were able to begin engaging with basic concepts of different programming and CT skills while using this device. As a result, the study presented in this chapter aimed to integrate the Cubetto robot into everyday classrooms to assess how learning with this device may improve children's CT skills and their beliefs about programming and robotics.

Intervention Administration

A recent paper by Ching and Hsu (2023) systematically reviewed 22 ER studies. They noted that a strength of past research is the volume of studies conducted in naturalistic school settings, which in turn strengthens findings and conclusions due to the increased ecological validity and applicability of findings. The current study also implemented a robotics intervention within classrooms. Moreover, this study further recruited classroom teachers to deliver the robotics and programming curriculum. This is something that previous research has lacked. Previously, robotics intervention curriculums have been delivered to children by primary researchers or research assistants (e.g., Sullivan & Bers, 2015; Sullivan & Bers, 2016; Sullivan et al., 2013; Strawhacker & Bers, 2015). In some cases (Sullivan & Bers, 2016), it was thought that teachers would learn and gain confidence in teaching programming and robotics content by observing a research assistant delivering the content.

However, findings from my focus group and survey research (see Chapters 2 and 3) suggested that learning through observation is not favoured by teachers. Instead, my findings have highlighted the importance of hands-on, experiential learning for effective teacher education and the improvement of teacher self-efficacy. This leaves open questions regarding the role of the researcher as both the expert and the teacher, and whether similar benefits of ER would be found if the classroom teacher (who is likely to have less knowledge of programming and robotics) delivered the curriculum. Moreover, past research has suggested that having classroom teachers deliver the robotics content would prompt pupils to engage more and ask more intricate questions (Hussain et al., 2006; Lindh & Holgersson, 2007). This would also provide teachers with the knowledge, experience, and ability to continue teaching programming and robotics education once the intervention had concluded.

In addition to being teacher-led, this intervention study also investigated whether attending an additional teacher education workshop (prior to the delivery of the robotics curriculum) positively impacted pupils' learning outcomes (i.e., CT abilities). Thus, teachers assigned to an Intervention+ condition attended a half day workshop at Cardiff University. This education workshop aimed to improve teachers' knowledge of CT, programming, and robotics in early education. It also aimed to improve their confidence, engagement, and ability to implement the prescribed robotics curriculum in their classrooms.

As my previous studies have found that primary school teachers often struggle to fit additional teacher education into their already busy schedules (see Chapters 2 and 3), it is important to explore the impact of teacher education workshops (attended prior to curriculum delivery). For example, do these sessions have additional benefits for pupils' learning outcomes, or do pupils learn just as well from a teacher who has not attended an education workshop? To investigate this, I included a second Intervention condition that did not provide an additional education workshop for participating teachers. I then explored whether improvements in pupils' learning outcomes differed depending on whether their teacher attended an education workshop or not. Implementing these conditions within the experimental design allowed for further explorations into the role of the teacher and whether additional teaching support improved pupils' CT outcomes.

Sample Age

Although investigating ER within primary school classrooms has been a popular research topic in the last decade, very few studies investigate robotics in early years settings (see meta-analysis by Wang et al., 2023). In a recent systematic literature review, Mangina et al., (2023) investigated robotics specifically in the context of primary and pre-school education. They found that only 2 of 21 studies tested robotics in early primary education with samples of children under the age of 6. Moreover, these two studies did not appear to involve children in formal learning activities with set goals, but instead simply encouraged playful interactions with the robotics kits.

Tselegkaridis and Sapounidis (2022) found similar results in their systematic review of common study designs used in robotics and STEM interventions. They reviewed 36 papers and found that two thirds of papers employed samples of children over the age of 7 years. Authors highlighted in their discussion that older samples may be favoured as it has previously been argued that STEM activities are best suited to older pupils due to their progressions in cognitive development. However, as evidenced above, more recent research has in fact illustrated that programming concepts can be successfully introduced to children as young as 4 years old (Bers, 2020). Thus, researchers should make a conscious effort to include younger samples within their interventions. This study aimed to contribute to the literature by designing and implementing a developmentally appropriate robotics intervention for children ages 4 to 7 years old.

Conducting research within this age range is even more important given the changes being made to primary school curriculums. Early childhood educators are required to teach CT, programming and robotics content to their pupils, however, findings from my previous research (see Chapters 2 and 3) have suggested that they are unsure how best to do so. By employing samples of children under the age of 7, researchers would be able to provide guidance to practising teachers.

Lack of Control Groups and Sample Size

The lack of research employing experimental designs is another limitation highlighted by several review papers. For example, Tselegkaridis and Sapounidis (2022) found that 61.9%

of the 21 papers included in their review and meta-analyses used non-experimental designs (defined as using no control group or not collecting multiple measurements across time). Similarly, Xia and Zhong (2018) reviewed 22 studies investigating the integration of ER within K-12 education. They found that 59% of the studies reviewed used non-experimental designs. The authors of these review papers suggested that by not utilising control groups in their research design, researchers cannot fully assess the validity of their interventions (Tselegkaridis & Sapounidis, 2022). Thus, the intervention study presented in this Chapter implemented an experimental design by utilising a control group and collecting data pre- and post-intervention.

Bakala and colleagues' (2021) review explored how robotics could promote the development of CT in preschoolers and found only four papers that successfully combined control groups and a pre-test post-test design (i.e., Kazakoff et al., 2013, Muñoz-Repiso & Caballero-González, 2019, Nam et al., 2019, Roussou & Rangoussi, 2019). However, only one of these studies (Nam et al., 2019) included a sample size of over 100 children and instead, most studies recruited samples of fewer than 80 children (Xia and Zhong 2018). In a review paper (Cheung & Slavin, 2013) that aimed to explore the effectiveness of educational technology interventions on mathematics achievement in K-12 classrooms, authors categorised small studies as those utilising a sample of less than 250 participants. They also reported a statistically significant difference in effect size between small and large studies, with the mean effect size of small studies (i.e., < 250 participants) being approximately twice that of large studies. Findings from small studies with larger effect sizes should be interpreted with caution, particularly if results are not replicated in studies with larger sample sizes that also employ similar experimental designs. Thus, this intervention recruited a sample of > 250 children to further investigate how robotics can be used to promote the development of CT skills in early education.

To summarise, this study aimed to investigate the effects of an educational robotics curriculum on primary school children's CT skills and beliefs. To address the gaps in the literature explored above, this study implemented a Cubetto intervention that was (1) delivered by classroom teachers, (2) designed for children between the ages of 4 and 7 years old, (3) employed a sample of > 250 pupils, and (4) utilised a control group along with pre and post outcome measures. Furthermore, this study aimed to investigate whether the addition of a teacher education program benefits children's learning outcomes. To do this, some teachers were invited to attend an education workshop, and others were not. This study aimed to answer the following research questions:

1) Can a 6-week Cubetto curriculum improve children's computational thinking skills (i.e., debugging, prediction, sequencing), and how do outcomes differ if teachers attend an additional education workshop?
2) Can a 6-week Cubetto curriculum improve children's beliefs about programming and robotics, and how do beliefs differ if teachers attend an additional education workshop?

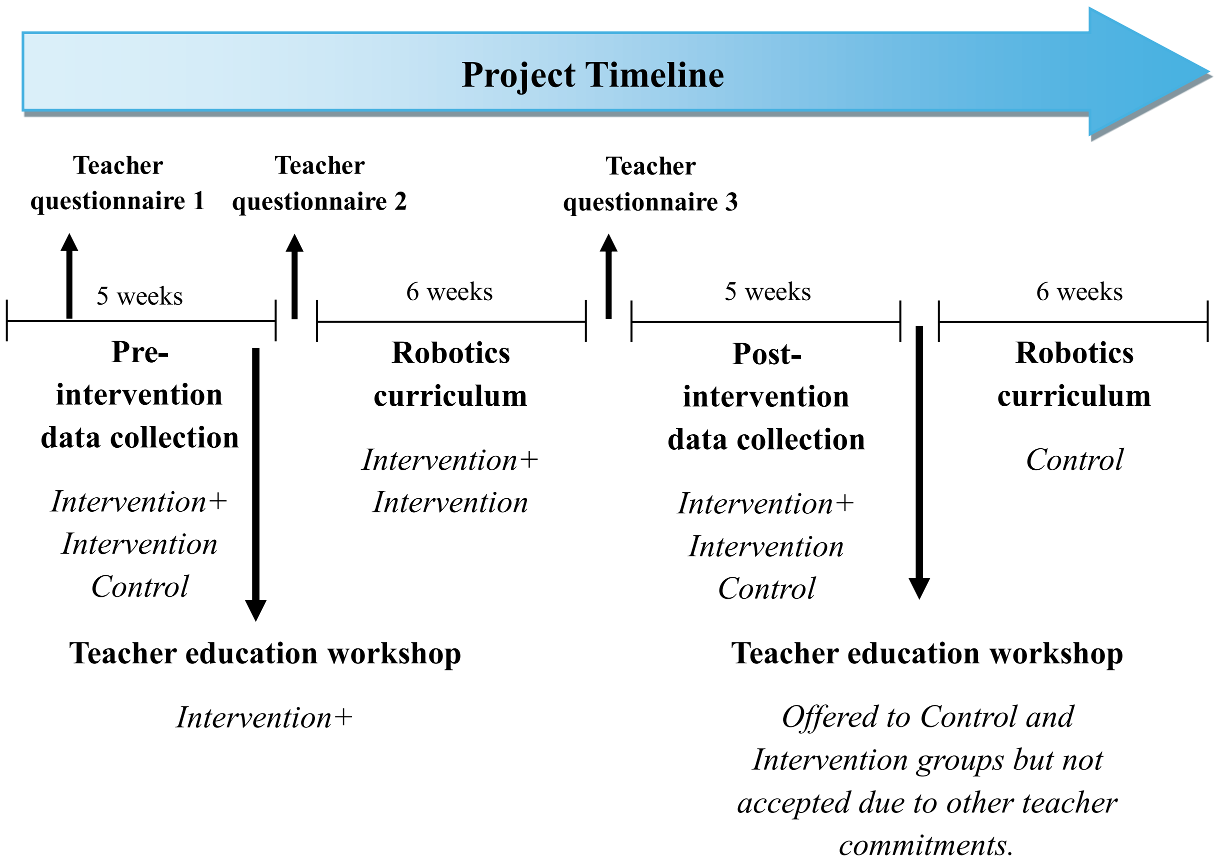
Methods

Design

This study employed a pre-test post-test design whereby classrooms were assigned to one of three conditions: no intervention (Control), classroom intervention (Intervention), or classroom intervention plus teacher education (Intervention+). Teachers assigned to the Intervention+ condition attended a teacher education workshop prior to the intervention, designed to give them the time and support necessary to learn about the programmable robot (Cubetto) and its accompanying lesson plans. Not all teachers delivering the robotics lessons received this additional input. Instead, the teachers assigned to the Intervention condition simply received an information pack and a selection of lesson plans to follow. Neither teachers nor pupils in the control condition received any experience with the robots until after post-intervention assessments (see Figure 5.1). Overall, this study investigated whether the teacher education workshop and/or a classroom intervention improved children's computational thinking skills *and* teacher's beliefs and competence in this area, relative to a control group that did not gain any experience with programming and robotics. This chapter focuses on pupil outcomes; the next chapter (Chapter 6) investigates teacher outcomes and how teacher outcomes may relate to pupil performance.

Figure 5.1

Illustration of the Project Timeline.



Ethical Approval

Online opt-in consent forms were completed by the head teacher, participating teachers and parents at each school. These forms explicitly stated that the data collected from schools (including teachers and children) could be withdrawn at any time. Additionally, as one of the pupil measures was screen recorded on an iPad, the parental consent form allowed parents to consent to their child's five minute session being screen recorded with audio. Children provided verbal consent at the start of each testing session. This study was approved by the Cardiff University School of Psychology Ethics Committee and is associated with ethics application number EC.21.11.09.6434.

Participants

Given the complex nature of the study design, which involved multilevel modelling (MLM) with repeated measures and hierarchical data, a formal power analysis was not conducted to determine the sample size. MLM requires accounting for multiple levels of nesting (e.g., pupils within classrooms, classrooms within schools) and the correlation of repeated measures within individuals, which complicates the power analysis process (Hox, 2017). Existing tools for MLM power analysis are limited, particularly in handling repeated measures (Hox, 2017). Consequently, the sample size was determined based on prior research and practical considerations (e.g., quantity of resources, availability of schools) to ensure robust analysis while acknowledging these methodological challenges.

Initially, fifteen schools across Wales volunteered to take part in this study. Teachers from these schools had previously completed an online survey (see Chapter 3) investigating teachers' experiences with programming and robotics in the classroom. Through the survey, they expressed interest in the current intervention. Those interested received a study information pack (see Appendix D) and were invited to attend a series of virtual Q and A sessions with me. During these sessions, staff were able to ask questions and clarify what would be required of their schools should they choose to take part. Of the fifteen schools that registered their interest, seven were chosen to take part in the project. Physical resources and time restraints for data collection restricted the number and locations of schools selected to participate.

In total, nineteen foundation phase classrooms (a mix of Reception, Year 1, and Year 2) participated in this study, from seven schools across South Wales. Each school was allocated to one of the three conditions mentioned above. Allocation was not completely randomised due to the size of the schools taking part. Two of the participating schools were particularly large (each with ~180 pupils spanning Reception to Year 2), so all classrooms within one of these large schools were randomly assigned to the Intervention+ condition and all classrooms within the other large school were assigned to the Control condition. Thus, the five remaining schools were assigned to the same condition to equate numbers in the Intervention condition. Whole schools were assigned to the same condition as it would not have been possible to control the spread of information between teachers if individual classrooms within a school were assigned to varied conditions. Specifically, it would have been difficult to ensure teachers who attended the education workshop did not pass information on to teachers who did not attend the workshop.

Across all schools, 17 teachers (92% female, mean age = 39.3) and approximately 550 pupils completed the robotics curriculum. All children in each classroom participated in the curriculum, however, each family had the option to allow or decline data collection pre- and post-intervention. As a result, data was collected from 430 children who had parental consent

(50.93% female, mean age = 5.9, range = 4.41 to 7.65). Data was excluded if children had a formal diagnosis that entitled them to 1:1 support for their additional learning needs, as they did not complete CT assessments independently. In most cases, children with additional learning needs did not partake in assessments given in this study. Participant distribution across intervention conditions was as follows: Intervention+ = 149, Intervention = 137, and Control = 144.

Procedure and Materials

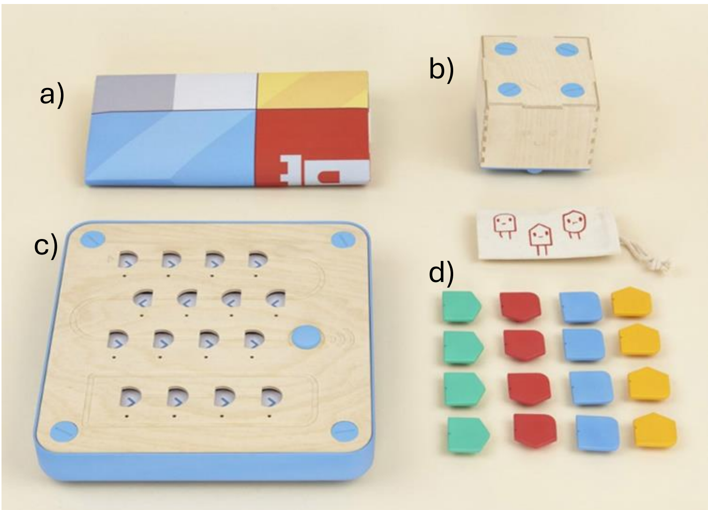
The intervention groups (Intervention+ and Intervention) participated in the robotics program while the Control group continued with their typical curriculum. All children were pre-tested before the intervention groups began the program and post-tested after the intervention groups completed the program. All the children who participated as controls had the opportunity to participate in robotics lessons after post-testing.

Robotics Curriculum

During this project, participating schools were provided with Cubetto robotics kits for the classroom. Cubetto was created by UK company Primo Toys (www.PrimoToys.com). Its design avoids textual and numerical language thus making it suitable for pre-literate children. Moreover, children can use the playset to get to grips with coding concepts through hands-on learning and without the use of a screen (Ching & Hsu, 2023). Cubetto comes equipped with an interface board, a range of function tokens (forward, right, and left turn functions) and a colourful floor map (see Figure 5.2). Children can navigate the robot around the map by placing the desired tokens in the interface board and pressing the ‘Go’ button. Through play, Cubetto introduces and provides the foundations for a host of programming concepts including ‘sequencing’ and ‘debugging’.

Figure 5.2

Cubetto Playset



Note: Each set comes with a floor map (a), a Cubetto robot (b), an interface board (c) and a range of movement tokens (d).

Lesson Plans. This study used a collection of lesson plans that were created by Primo Toys with the support of an early year's teacher. Participating teachers were given access to a total of 53 lesson plans. Six of these lessons were made compulsory and teachers were asked to complete at least one plan for each week of the program (see Appendix E). The plans selected were chosen to ensure that pupils gained experience with a range of programming skills (e.g., debugging, sequencing and prediction). Over the 6-week course, I did not limit the number of lessons or programming activities that teachers could deliver. Instead, they were welcome to use the additional lesson plans and were encouraged to create their own programming tasks for their pupils. Teachers could note any additional lesson plans used in their weekly feedback form. This informal feedback form prompted teachers to record how much time they spent teaching with robotics that week, whether they made changes to the lesson plans and whether they used any additional lesson plans. Teachers were also asked to note how the session(s) went, reflecting on both positive and negative aspects.

The first compulsory lesson plan introduced children to the robot. The classroom teacher explained that, as Cubetto is a robot, it cannot think on its own and needs to follow instructions given by a human. They also explored what the different tokens made the robot do. The main objective of this lesson was to understand how algorithms are implemented on devices, and by the end of the lesson, pupils should have been able to put instructions in order (sequencing) and understand how to follow an algorithm.

The central activity in lesson two focused on pupils programming the robot to move around the activity map to collect various shapes. This task allowed children to practice writing algorithms and reinforced the function of each movement token. This lesson also integrated mathematics concepts as children discussed and described the properties of different shapes.

The objective of week three's lesson was for children to predict the outcome of simple algorithms. In this lesson, teachers supported pupils as they learnt to predict what an algorithm would make the robot do. Snakes and ladders games were used to introduce this concept as children were encouraged to roll the dice and predict where the robot would finish once the robot had travelled across that many squares. For example, they were asked to think about whether Cubetto would land on a particular themed square (i.e., a tree or a boat), would travel up a ladder or slide down a snake. They then programmed Cubetto to move to the required number of squares.

Lesson four built upon this introduction to prediction. In this session, children wrote their own algorithms and asked their friend to work out where the robot would finish on the map (see Figure 5.3). This task aimed to advance their prediction skills as they worked with the movement tokens rather than counting squares after the rolling dice.

Figure 5.3

Child Using Cubetto Robot for Prediction Activity.

Note: A sequence of tokens has been placed into the interface board (left). The child has placed an "x" on the map where they think Cubetto is going.

The fifth and sixth sessions formally introduced debugging. In week five, children wrote their own algorithms on a whiteboard to get the robot to a destination on the map. They then rubbed out one of the tokens and handed the board to a friend. Their friend then identified the correct token to fill the gap, thus completing the sequence.

In the sixth and final (compulsory) session, children designed a course or maze for Cubetto. They then had to write an algorithm to help Cubetto get through the course. One of their peers would then place an obstacle in their path. The first child then needed to debug their old algorithm so that Cubetto avoided the new obstacle.

Teacher Feedback Forms

Teachers delivering the robotics curriculum were asked to complete feedback forms after delivering each lesson plan. In their weekly feedback forms, they recorded how much time they spent teaching with Cubetto that week and whether they made any changes to the set lesson plans. Teachers also had space to provide information about how the session(s) went, noting both positive and negative aspects. These forms were given to teachers so that we could monitor how they were implementing the robotics curriculum.

Control group teachers were also given a weekly feedback form to complete. The guidance attached to this form explained that we wanted to know whether they and their class had taken part in any programming or computational thinking activities that week. A clear disclaimer was included stating that although we were not encouraging teachers to teach such activities before our next visit (for post-intervention data collection), we wanted to know if they had. Teachers were informed that even if their answer was "No", they should still complete the form. In this feedback form, teachers provided their name and the name of the school they worked at. They were then provided with the following information:

> "*Over the last week, has your class done any activities linked to coding, programming, or computational thinking more broadly? (This can include class work, games, videos etc).*
>
> *Note: Computational thinking skills include recognising patterns, creating step-by-step instructions (algorithms), identifying errors, and fixing them (debugging) and planning."*

Those who responded with "Yes" or "Maybe" were then asked to describe the activities they delivered to pupils in an open-ended text response.

Classroom Assessments

As the primary investigator, I delivered three pencil paper tasks to children in their classrooms. Previous research (see Bakala et al., 2021, for review) has favoured portfolio style assessments of children's abilities, often including hands-on testing with the chosen programming equipment. Bakala et al., (2021) defined portfolio style assessments as an *"Evaluation of student's products during robotics activities through the use of rubrics or* *checklists"* (p. 10). These assessments differ to 'Traditional' assessments which can include multiple choice tests evaluated by accuracy. The decision was made not to include hands on activities with the robot in the current pupil assessments, as participants in the Control condition would be at a disadvantage during testing having not used the robot toy. Similarly, all children would have been disadvantaged during pre-intervention assessments. Instead, the paper measures placed all participants on a level playing field as they only needed to understand the function of three coloured tokens. Additionally, for classroom use in the future, paper assessments are advantageous for several reasons. Firstly, they are more time effective as a large group of children can be assessed at the same time. Secondly, paper handouts can be marked easily and can be used by teachers to evidence pupils' learning.

Visual Perspective Taking. The first paper-based task children completed was an adaptation of the Visual Perspective Taking (VPT) task used in the laboratory study described in Chapter 4 (previously adapted from Tversky & Hard's 2009 study).

Children watched a video demonstration of this task being completed by a model participant and model experimenter. In this video, two people sat opposite one another. Model A (the experimenter) sat behind a display of coloured blocks. Model B (acting participant) sat with an answer booklet and coloured stickers. Instructions for the task were narrated in the video. Children watched as Model B followed these instructions, filling out the answer booklet. During their first attempt at the task, the Model B answered incorrectly. The narrator acknowledged this and prompted them to try again. On the second attempt, the actor completed the trial correctly. The narrator congratulated them, and the demonstration video ended. Children viewed this video twice in their classrooms.

Children then completed this task themselves. The experimenter repeated the instructions given to the acting participant in the demonstration video.

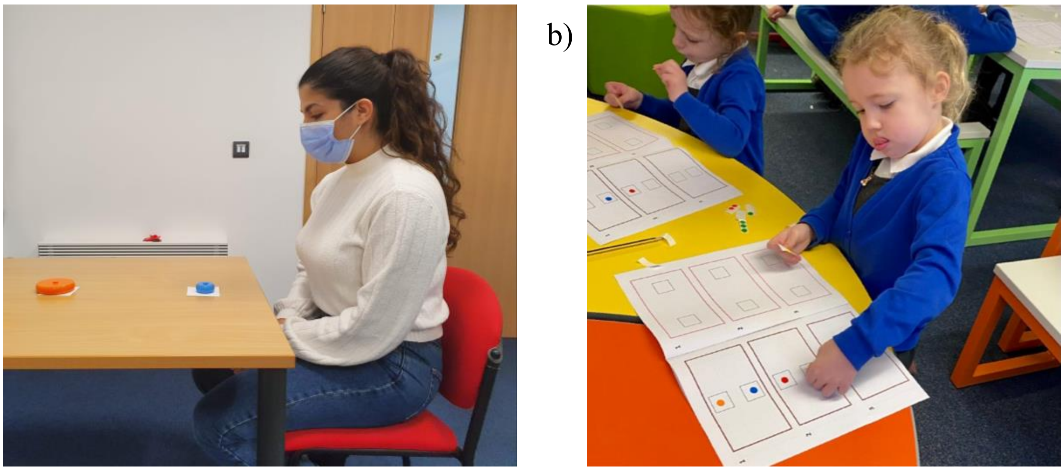
> *"Now it's your turn to play this game with* Model A. *You need to imagine that you are sat in her chair, in her body, and that you can see through her eyes."*

This task contained 12 trials: 3 near-far 180-degree rotation trials, 3 left-right 180-degree rotation trials, 3 near-far 90-degree rotation trials (see figure 5.4a), 3 left-right 180-degree rotation trials.

Figure 5.4

Visual Perspective Taking (VPT) Task.

a) b)



Note: Image a is an example VPT trial shown to children (a 90-Degree, Near-Far Trial). Image b is a pupil placing matching stickers into their answer booklet. In Image b, child's answer in box one is correct as the blue block was close to the Model and the orange block was further away.

For each trial, an image of *Model A* and her coloured blocks was displayed on the class whiteboard (see Figure 5.4a). The experimenter explicitly stated what colour stickers they needed for each trial:

> *"Look at the picture on the screen.* Model A *has some coloured blocks in front of her and you have stickers to match! First, you need your green and blue, your blue and green stickers."*

The colour pairing was given twice to avoid any positional bias (i.e., children placing a certain sticker first because the experimenter verbalised that colour first). Children were then instructed to put their stickers in their workbook. *"Imagine you're looking through* ***her*** *eyes. Can you put your colours how* ***she*** *sees them?"* (see Figure 5.4b). Answers were recorded by placing the stickers in boxes printed on the page. Participants were given a score of 1 for each correct trial, and a score of 0 for incorrect trials. An example of a correct answer can be seen in Figure 5.4b.

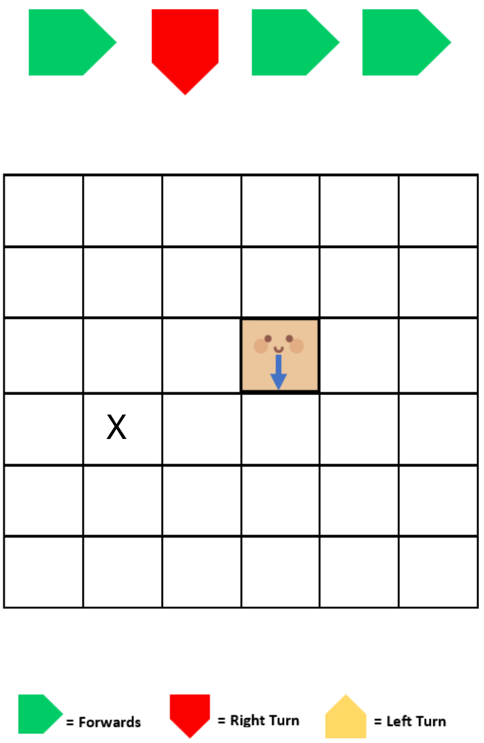
This data was collected in parallel with the laboratory study. Since then, analysis of the laboratory data suggested that children's VPT abilities and their robot programming skills were not related (see Chapter 4). For this reason, this VPT data was not analysed further in this pupil intervention chapter.

Robot Introduction. Prior to the next assessment tasks, participants were given a short presentation that introduced a cartoon robot and its movement tokens. I explained that the robot was unable to move across a grid on its own and instead needs to follow a sequence of instructions (i.e., coloured tokens). Participants heard that each token gave the robot a different command (i.e., green = one square forwards, red = right turn, yellow = left turn) and watched video demonstrations for each of the three tokens. Participants viewed these demonstrations twice and were given a rule check (e.g., *"what does the green token do?"*) in-between to check their understanding of the token functions.

Prediction. The aim of this task was to assess whether children were able to view a sequence of tokens and predict where the robot would end up on a grid (see Figure 5.5). To introduce this task, I first completed two demonstration trials on the class whiteboard, during which participants worked with me to work out where the robot would stop. An "X" was drawn on the grid to signify where the robot would be on the grid after following the algorithm. Participants then watched as the robot on the screen followed the algorithm. The first demonstration trial contained only "forward" tokens and the second featured a "turn" token.

Figure 5.5

Example Prediction Trial.



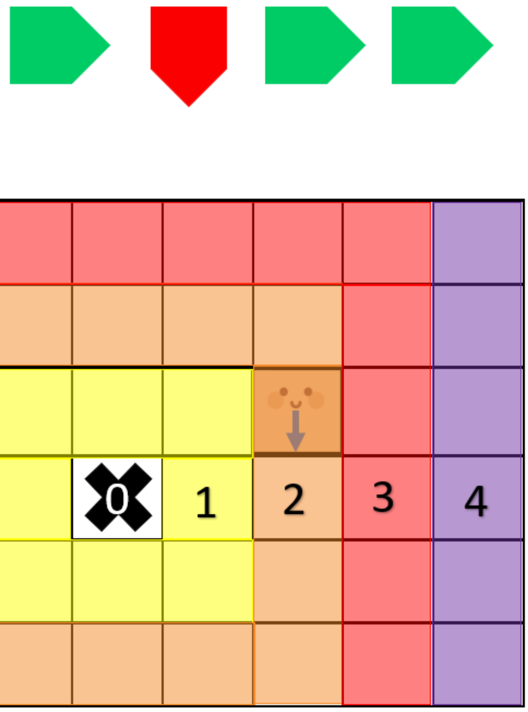
Note: The robot's instructions are shown at the top of the page. The blue arrow on the robot's front represents the orientation of the robot. Written reminders of what each token does appears at the bottom of the page. The "X" on the grid shows the correct answer for this trial.

During this paper and pencil task, participants were required to draw their answers in their individual answer booklets. Inside the booklets there were 10 trials of varying difficulty. Difficulty was determined by a combination of the number of tokens used (i.e., more tokens = harder trial), the type of tokens used (only forward tokens = easy, one turn token = medium, multiple turn tokens = hard) and the orientation of the robot (i.e., in harder trials the robot was facing the participant, meaning the participant's perspective was not aligned with the robot's). In-lab pilot trials with children from this age group helped determine the difficulty level of each trial.

When coding this data, participants were given a score of 1 for each correct trial, and a score of 0 for incorrect trials. Thus, a proportion of correct prediction trials was calculated from the trials they attempted. The accuracy of each participant's prediction on each trial was also assessed. Using a framework made of concentric squares (see Figure 5.6) participants received a score that signified how close their "X" was to the correct answer. The lower the score, the more accurate the prediction. An average accuracy score for each child was calculated across the trials they attempted.

Figure 5.6

Coding Diagram Example for Prediction Trial.



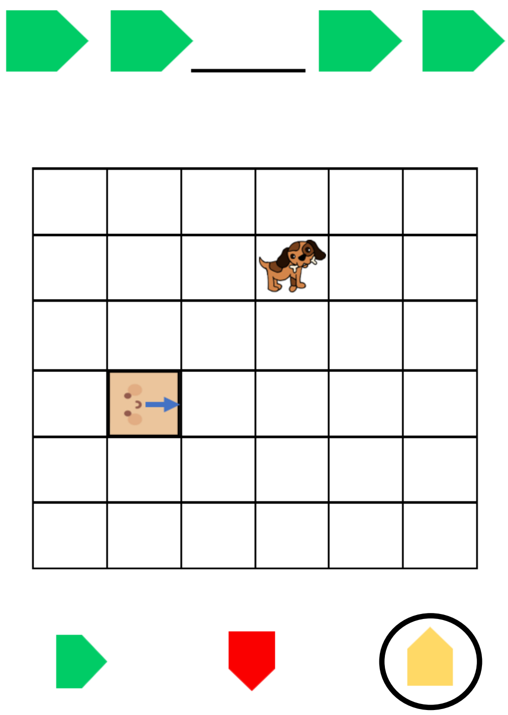
Note: Participants were given a higher score for a less accurate prediction. Therefore, a score of 0 signified the participant provided the correct answer. Answers in the yellow squares were given a score of 1. Answers in the orange squares a score of 2, red squares 3 and purple squares 4.

Debugging. This task modelled debugging tasks used in previous research (see Strawhacker & Bers, 2019, Solve It tasks). During this assessment, participants were required to choose a command that would resolve an error in an algorithm. To begin, I completed a demonstration for the class. Participants viewed a robot animation on a PowerPoint presentation. They watched as the algorithm provided did not get the robot to a target object. I then removed the incorrect token from the sequence, leaving a gap in the algorithm. Working with me, participants chose the correct token to fill this gap. I then ran the algorithm again with the correct token, and participants watched as the robot reached the target object. Two demonstration trials were completed in this manner (the first with a missing forward token, the second with a missing turn token).

In their individual answer booklets, participants then completed 10 trials of varying difficulty which increased as trials progressed (see Figure 5.7 for an example). In this task, difficulty was determined by the type of token used in the sequence and the number of missing tokens. At the start of each trial, the children watched a robot animation on a PowerPoint presentation. They watched as the robot followed the algorithm on the screen and failed to reach the target object. Like the demonstration presentation, errors in the algorithm were highlighted for the participants to correct. In their answer booklet, children were instructed to circle the token needed to fill the gap in the algorithm. Participants were given a score of 1 for each correct trial, and a score of 0 for incorrect trials.

Figure 5.7

Example of a Debugging Trial.



Note: The robot's algorithm is shown at the top of the page. Children selected the correct token to fill the gap. The circled token is the correct answer for this trial ("left turn" token).

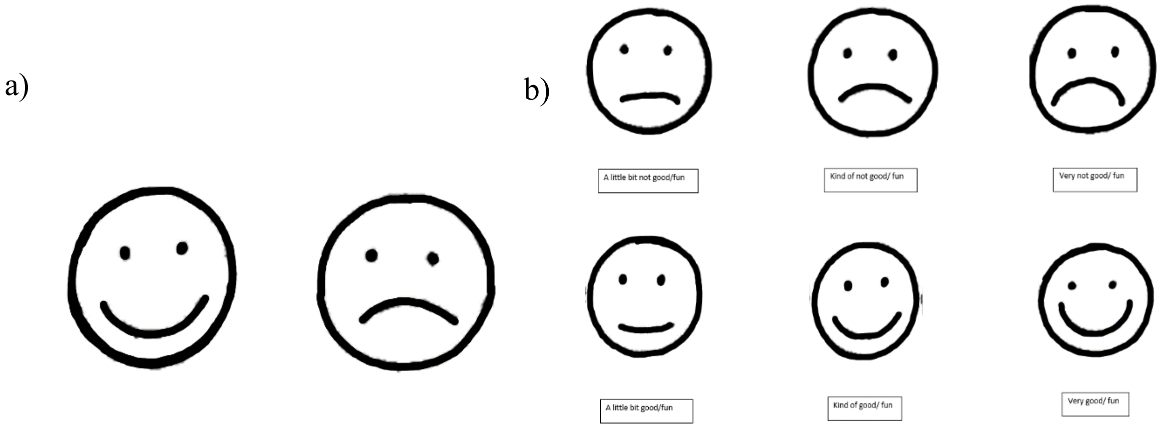
Individual Assessments

Due to time restraints during testing days, a random subsample of participants (Intervention+ = 110, Intervention = 103, Control = 126) completed a range of tasks during a 1:1 assessment with me or a research assistant. These four tasks took approximately 20 minutes in total to complete.

Pupil Beliefs. Participants answered questions about interest and self-efficacy on a 6-point Likert scale. To familiarise participants with this scale, they first responded to two practice items (previously used by Master et al., 2017). Each question item was asked in two stages (known as "branching") to keep the number of choices simple and age appropriate (Master et al., 2017). Designed to familiarise children with the positive side of the scale, the first practice item asked children whether playing outside is fun or not fun (step 1). They were asked to point to a smiley face or a frowning face to indicate their answer (see Figure 5.8a). Depending on their answer, we then asked the participant how "fun" or "not fun" it was (step 2). During step 2, Master et al. (2017) previously used a card showing faces with three sizes of smiles (or frowns), accompanied by the labels a little, medium and a lot. For the purposes of this study, the scale labels in step 2 were changed due to concerns that the wording was awkward and unnatural (i.e., "playing outside is medium fun" and "getting hurt is a lot not fun"). Instead, the following labels were used: a little bit, kind of and very (see Figure 5.8b). As the second practice question, children were asked if getting hurt is fun or not fun to familiarise them with the negative scale.

Figure 5.8

Measure Used to Assess Pupil's Beliefs.



Note: Image a shows step 1 of the attitude measure and image b shows the negative and positive scales presented at Step 2.

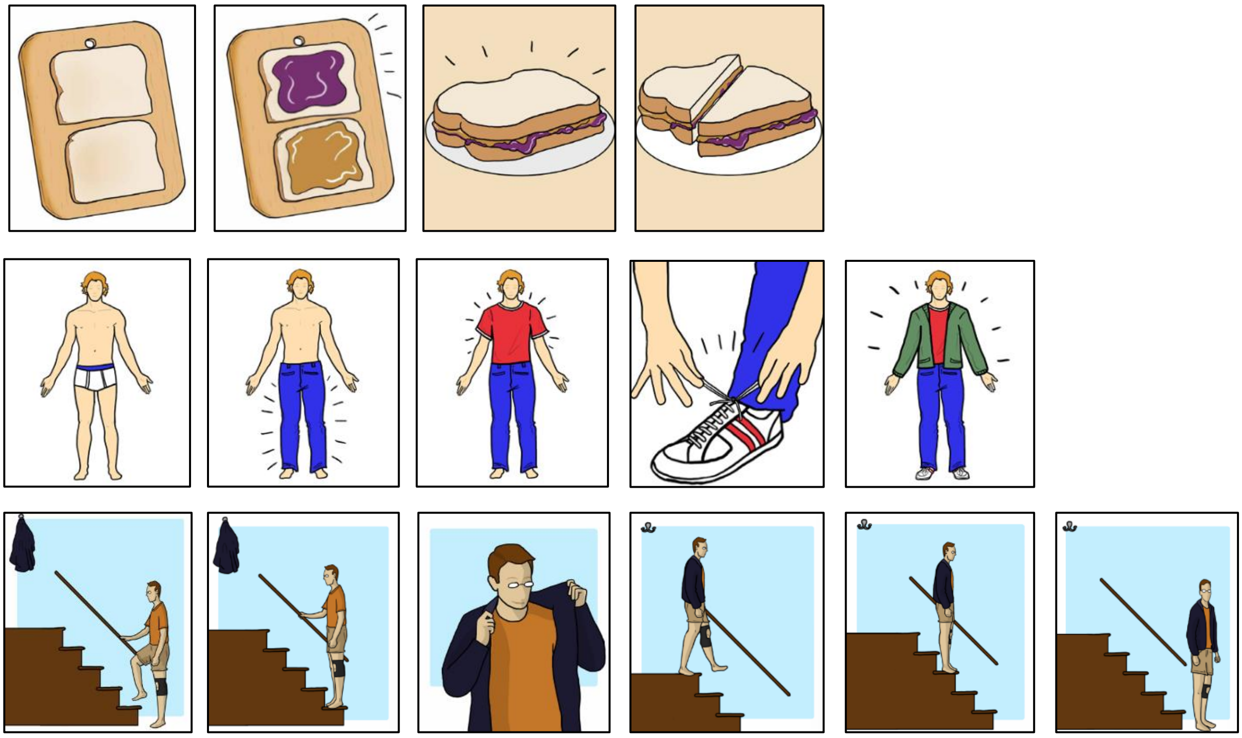
Following the practice items, technology motivation was measured using three items, developed by Master et al (2017) and adapted from Mantzicopoulos and colleagues' (2008) scale assessing young children's interest and liking for science. These items assessed interest in programming *("Is programming fun or not fun?")*, interest in robots *("Are robots fun or not fun?")* and self-efficacy with robots *("Are you good, or not good with robots?")* Before these items were administered, we provided all children with a definition of programming, *"Programming is when you tell a computer or a robot what to do"* (Master et al., 2017, p. 6). When scoring steps 1 and 2 were combined to create a 6-point scale with 3 positive values and three negative values. Thus, the scoring of participants' responses ranged from 1 (*very not fun/good*) to 6 (*very fun/good*).

Sequencing. Children completed a picture-sequencing task. Picture sequencing assessments are common educational tools and assessment measures for sequencing skills in early childhood classrooms (e.g., Brown & Murphy 1975; Linebarger & Piotrowski 2009). A picture sequencing assessment was chosen due to its similarities with writing algorithms. For instance, a foundational concept of algorithms is that order matters as the order in which children place programming tokens dictates the behaviours of the robot (Bers, 2019). Thus, to write algorithms successfully, children must understand that steps of a program must be completed in a certain order to achieve the desired outcome. The same can be said for picture-sequencing assessments.

The sequencing assessment used in this study featured 6 picture sequences. Sequencing picture cards have previously been used by Baron-Cohen et al., (1986) and more recently by Kazakoff et al., (2013). The picture sequencing tasks in this study were specifically chosen as they featured activities or tasks that young children are familiar with. Trials varied in difficulty, with difficulty being determined by the number of picture cards in the sequence. A total of two 4-card sequences (easy), two 5-card sequences (medium) and two 6-card sequences (hard) were used. Correct answers are shown in Figure 5.9.

Figure 5.9

Picture Cards Used in the Sequencing Assessment.



Note: Images downloaded from SOAR Therapy (https://www.etsy.com/uk/shop/SOARTherapyPDX).

Participants were told, *"I need your help to put these cards in the right order."* As they gestured from the left side of the table to the right, the researcher instructed them to place the cards from left to right. There was no time limit for children to complete this task. Participants were awarded a score of 1 for each correct sequence and a score of 0 for each incorrect sequence. Scores could range between 0 and 6 for both the pre-test and the post-test.

Executive Functioning. The Minnesota Executive Function Scale (MEFS; Carlson & Schaefer, 2012) was also used in the laboratory study described in Chapter 4. Like in the previous laboratory study, this measure was included as a measure of individual differences. The tool assessed cool executive functions (a collective assessment of working memory, inhibitory control, and cognitive flexibility) and was adapted from the Dimensional Change Card Sort task (Zelazo 2006). Children sorted cards on a touchscreen tablet based on different rules, i.e., sort by shape or by colour (see Figure 5.10). There were seven levels of varying complexity. Starting levels were determined by the child's age. To progress, children had to accurately sort cards in at least 4 out of 5 trials. If they failed to do so, they dropped back to the previous level. A Total Score (ranging from 0 to 100) was calculated based on accuracy and response time. A higher Total Score indicated both accuracy and speed in completing the task.

Figure 5.10

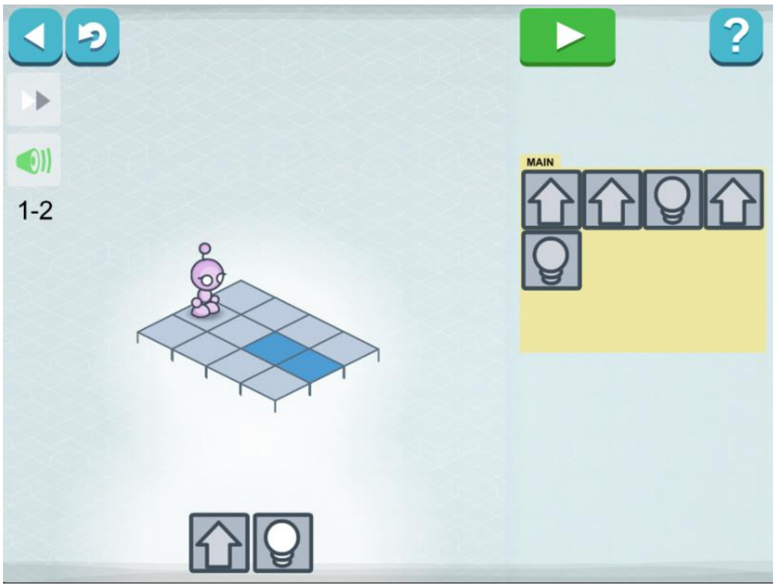
Screenshot Taken from the MEFS Assessment App.

Note: Image acquired from Apple.com (https://apps.apple.com/us/app/mn-executive-function-scale/id967184252)

Transferal of Programming Skills. Lightbot Jr is a programming puzzle game designed for children aged 4 to 8 (Apple.com). This game was chosen due to its similarities with the Cubetto robot. Like Cubetto, children programmed Lightbot using visual tokens and thus reading skills were not required. To introduce the game to participants, the onscreen instructions were read to the child by myself or a research assistant. The aim of the game was to help the unfamiliar robot reach a goal (i.e., to light up all the blue tiles; see Figure 5.11). Learners attempted to direct a robot around the screen to achieve this goal. They did so by creating a sequence of pre-planned instructions (i.e., forward, turn and lightbulb). Participants were given five minutes of uninterrupted play with the game and could complete up to nine levels. As they progressed through the levels, the on-screen instructions were read aloud by the researcher. Children did not receive help during this task. If a child asked for help, prompts like *"what do you think?"* were used.

Figure 5.11

Screenshot Taken from the Lightbot Jr App.



Note: Children must use the tokens at the bottom of the screen to complete the task. The box on the right side of the screen displays the answer for this trial.

It is important to note that a limitation of the Lightbot Jr app is that the game, in some instances, allowed children to proceed past a trial even if the algorithm they gave was technically incorrect (i.e., they achieved the intended goal but also had extra tokens added onto the end). For this reason, screen recording videos were later coded for programming efficiency.

Completing a trial "efficiently" meant that participants completed the task with minimum complexity (i.e., the exact, minimum number of tokens required; also used by Chen et al., 2017). Participants received 2 points for each level completed efficiently, and 1 point for each level completed but with additional tokens. Children were able to make as many attempts as they needed to complete each trial (within the 5-minute time limit). Points for each trial were totalled to provide a total programming score (at pre and at post).

Multilevel Modelling

Multilevel modelling (MLM) is an appropriate statistical technique to use when data is nested within a hierarchal structure (Hox, 2017). In this study, pupils are nested within classrooms (represented by teachers) which are nested within schools. MLM has been used in this analysis as the robotics intervention was administered at the group level (i.e., across individual schools), and outcomes (i.e., pupil's computational thinking skills and beliefs) were measured at the individual level (Koepsell et al., 1992). While there may be variability in pupil outcomes due to condition allocation, some of this variability may be due to differences between classrooms or schools. Additionally, pupil data has been collected at two time points (pre- and post-intervention), thus measurement occasion is nested within the individual. MLMs account for the fact that observations made on the same individual are likely to be correlated.

Fitting Models

All models were built and analysed using R Statistical Software (R Core Team, 2023), specifically the *lme4* package (Bates et al., 2015). In fitting mixed-effects models using this package, the methodology primarily relies on Maximum Likelihood Estimation (MLE), with Full Information Maximum Likelihood (FIML) employed by default (Grund et al., 2018). MLE seeks parameter values that maximise the likelihood function, providing unbiased estimates. FIML, integrated into *lme4*, accommodates missing data by utilising all available information, including cases with missing values, during parameter estimation. In this study, data was missing at times when children were absent or left testing sessions early due to other school obligations.

To achieve the aims of this study, all base models typically included fixed effects of condition (Intervention+, Intervention, Control). Additionally, for binomial assessment measures, interaction effects of condition and assessment time-point (i.e., pre intervention and post intervention) were also included. An interaction of Time and Condition could not be implemented for measures that were not scored binomially as difference scores (i.e., between pre-intervention and post-intervention) were used when individual trials could not be inputted into the model. In addition, a random effects structure was fit with random intercepts only. School-level factors (i.e., school funding, resources, and organisational structure) have been found to affect pupil achievement (Hofman et al., 2002). However, these factors were unlikely to influence the administration or effectiveness of this intervention for a few reasons. Firstly, robotics playsets were provided for all participating classrooms. Additionally, structured lesson plans were provided to ensure that teachers took the same approaches to programming and robotics lessons.

It is possible that classroom and teacher factors resulted in variations in the delivery of this robotics intervention. For example, children may have had different learning experiences due to classroom factors like class size, and teacher factors such as years teaching experience and teacher beliefs. Additionally, variance may have been caused by individual differences between children. For these reasons, to account for maximum variability, both TeacherID (representing individual classrooms) and PupilID were both included as random intercepts. However, this raised singularity errors in R. Such errors occur when the random effect structure is too complex to be supported by the data. Recommendations for handling such errors include removing only terms required to allow for a non-singular fit (Barr, 2013).

As pupils were nested within teachers (or classrooms) the decision was made to keep TeacherID (and remove PupilID) in the random effects structure for the following analyses. Additionally, Akaike Information Criterion (AIC; Akaike, 1974) values can be used to compare models and determine which random effect structure is the best fit for the data. The best fitting model is the one that explains the greatest amount of variation using the fewest possible independent variables. Smaller AIC values indicate a better model fit. Across all five pupil measures, including TeacherID as a random intercept provided lower AIC values (than PupilID) and thus proved to be a better fit for the data. These analyses further supported the proposed random effect structure. Furthermore, teacher effects (i.e., teacher competence) have been found to be an important factor influencing pupil achievement (Gustafsson, 2003).

Random slopes were not included in models due to the objectives and constraints of the study. The primary aim of this study was not to explore variations in the effects of the intervention between individual schools. Instead, it investigated the impact of the overall condition on child outcomes. Furthermore, in some instances, specific conditions consisted of only one school. Thus, the distribution of schools across conditions limited the ability to assess how the effects of condition may vary between schools.

Results

Delivery of Cubetto Curriculum

As explained above, teachers completed weekly feedback forms during the intervention period (six weeks = six forms). All six feedback forms were completed for the six classrooms participating in the Intervention+ condition. In the Intervention condition, all six forms were completed for three of the five classrooms. For the remaining two classrooms, teachers completed five of the forms but the sixth and final form was not completed after delivering the lesson (although teachers confirmed they had delivered the final session). No pupils (in either intervention conditions) completed any of the additional Cubetto lesson plans. Overall, time spent delivering individual Cubetto lessons seemed to range between 1 and 2 hours. Short feedback forms were also sent to Control group teachers to track the delivery of any computational thinking activities as they delivered their regular curriculum. These forms had a very low response rate, with only two teachers returning forms across the intervention period.

Preliminary Analyses

Year Group

For each of the outcome variables, the effect of Year Group was tested as a main effect to investigate whether differences in task performance or pupil attitudes varied across year groups. Additionally, models including a three-way interaction of Condition x Time x Year group were tested. I had no theoretical reason to believe that intervention outcome would differ based on year group as this study was investigating individual pupil changes between pre- and post-intervention time points. The results of this analysis supported this assumption as Year Group and its interactions with Condition and Time were not statistically significant for any of the pupil measures (both computational thinking related and beliefs). Thus, data was pooled across year groups and analyses involving Year Group were not reported in this Chapter. There were main effects of age overall (i.e., when not also exploring effects of Condition x Time), however these results will not be discussed further here as they were not relevant to the aims of this study (instead, see Appendix F).

Gender

The effect of Gender was also tested as a main effect and as a three-way interaction of Condition x Time x Gender. Results showed main effects of Gender for the number of correct trials on the debugging and prediction tasks. Post-hoc analyses revealed that boys were more likely to get trials correct on the debugging task (odds ratio = 1.11, $z = 1.96$, $p = 0.05$) and the prediction task (odds ratio = 1.26, $z = 3.98$, $p < 0.001$). No main effect of Gender was found in the remaining outcome measures (i.e., prediction accuracy scores, sequencing, or Lightbot programming) nor amongst pupil beliefs. Additionally, there were no significant interaction effects of Condition x Time x Gender. Thus, data was pooled across male and female pupils and analyses involving Gender were not reported.

Computational Thinking

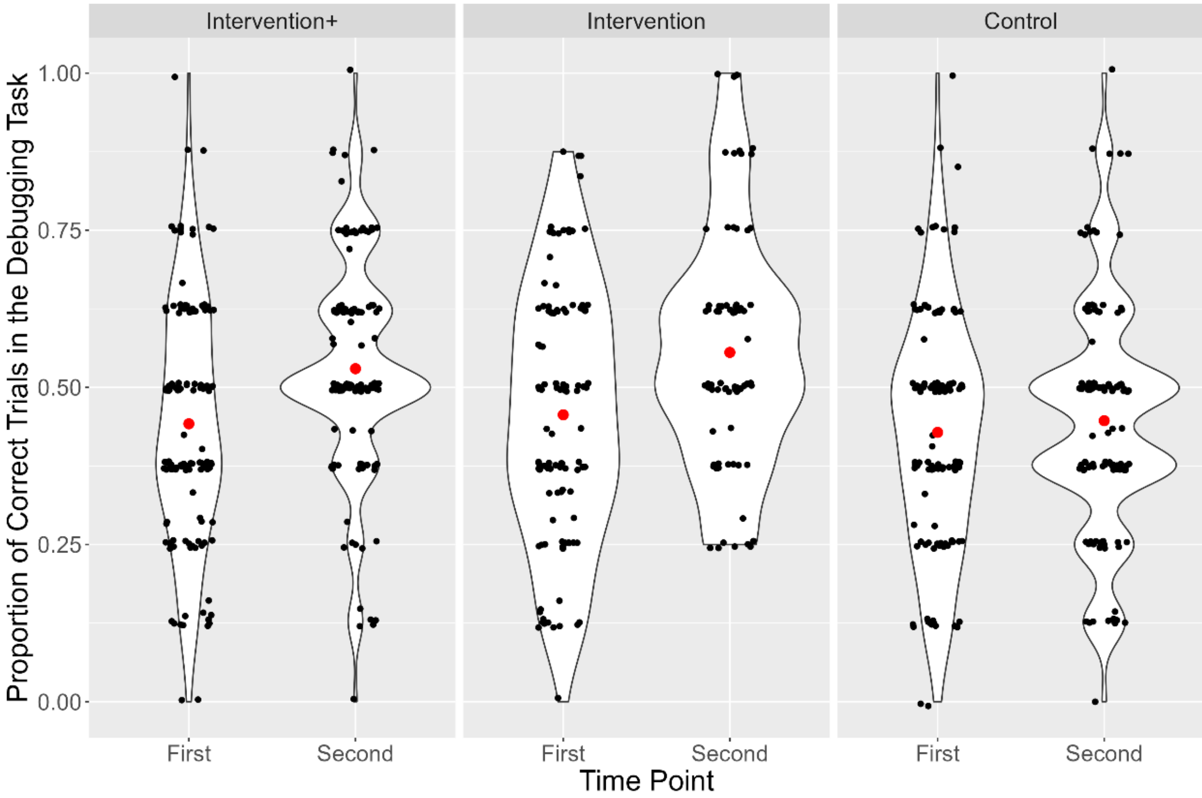
The following section explores differences in children's computational thinking outcome measures (i.e., debugging, prediction, picture-sequencing, and Lightbot Jr programming) and their beliefs between pre and post intervention testing. Differences in performance are explored between the three experimental conditions (Intervention+, Intervention and Control).

Debugging

Debugging task data was collected from 409 children at pre- or post-testing (21 children did not complete this task at both pre- and post-intervention and thus provided no data at all). Although 10 debugging trials were completed by children (pre- and post-intervention), 8 trials were included in the final analyses. This was due to a misprint in the assessment booklet which occurred at trial 9. As trial 9 could not be answered correctly, trials 9 and 10 were removed from the final analyses. The decision was made to also remove trial 10 in case the misprint in trial 9 mislead pupils. Figure 5.12 illustrates the proportion of correct trials on the debugging task, grouped by condition and time point. Within R, plots were created with the *Tidyverse* packages (Wickham et al., 2019).

Figure 5.12

Portion of Correct Trials in the Debugging Task, Grouped by Condition and Time Point.



Note: Group means are displayed in red.

A Generalized Linear Mixed-effects Model (GLMM) from the *lme4* package in R (Bates et al., 2015) was used to analyse these differences. The current model included debugging performance as a binary outcome variable for each trial (0 = incorrect trial, 1 = correct trial). The interaction between Condition (Intervention+, Intervention and Control) and Time (pre- and post-test) was included as a fixed effect and observations were grouped by Teacher. Confidence intervals were computed with the *confint()* function. P-values were obtained using the *lmerTest* package (Kuznetsova et al., 2017). The model that was estimated used the following *lme4* structure:

Debugging Performance ~ Condition*Time + (1|TeacherID).

Table 5.1

GLMM model results: Fixed effects.

	Model summary					
	β	*SE*	*z*	95% CI	*p*	
Intercept	-0.33	0.16	-2.00	[-0.64, 0.01]	0.04	*
Intervention+ Vs Control	0.07	0.25	0.30	[-0.43, 0.58]	0.76	
Intervention Vs Control	0.11	0.22	0.49	[-0.35, 0.56]	0.63	
Time Point	0.07	0.09	0.78	[-0.11, 0.25]	0.44	
Intervention+ Vs Control (pre- vs post-intervention)	0.30	0.13	2.35	[0.05, 0.55]	0.02	*
Intervention Vs Control (pre- vs post-intervention)	0.22	0.15	1.49	[-0.07, 0.50]	0.14	

Note: * $p < .05$. Intervention+ Vs Control (post-intervention), illustrates whether there is an interaction between Conditions (Intervention+ and Control) and Time (pre- and post-intervention).

The fixed effects from the model results are presented in Table 5.1. The model revealed a significant interaction effect of Condition x Time on children's debugging performance. On average, improvements in children's scores in the Intervention+ condition were significantly larger than those in the Control condition ($p = 0.02$). There were no significant differences in improvements on the debugging task between the Intervention condition and the Control condition ($p = 0.14$). Similarly, between the Intervention+ and the Intervention condition, there were no significant differences in improvements in debugging ($B = 0.09$, $SE = 0.14$, $z = 0.59$, CI 95% [-0.37, -0.20], $p = 0.56$).

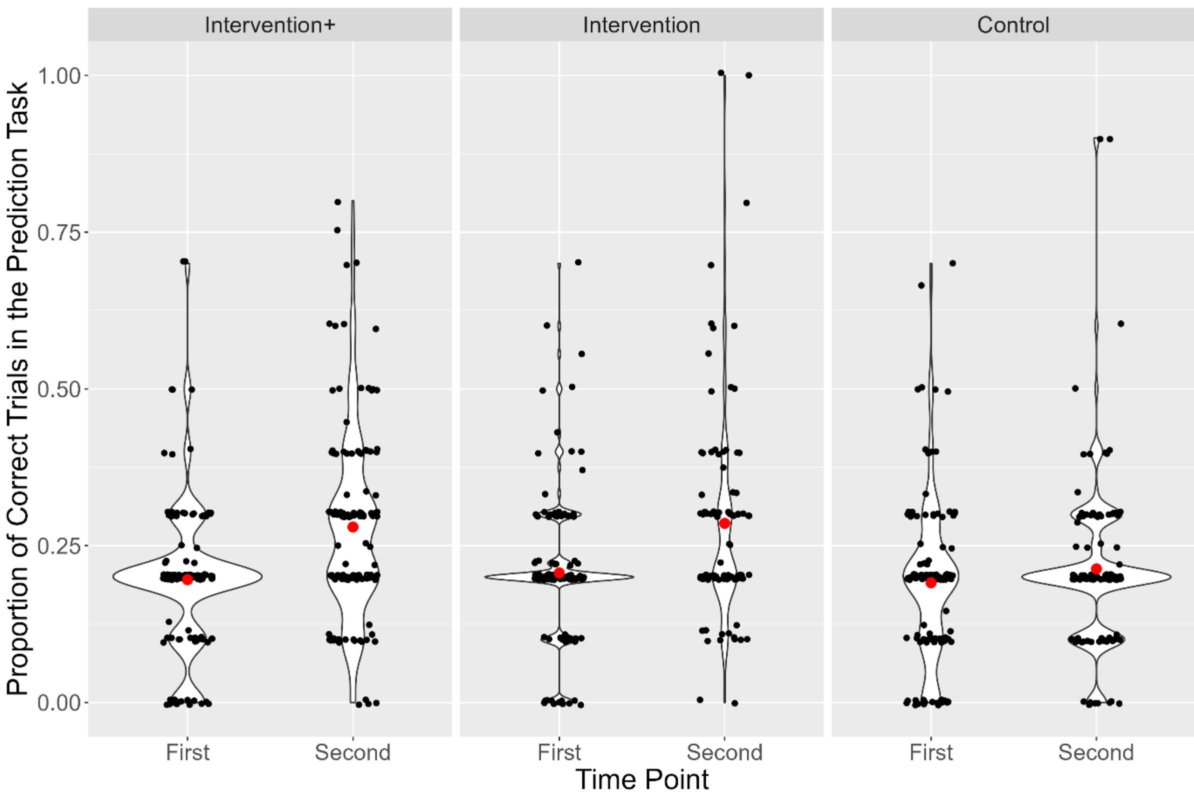
The Condition x Time interaction was followed up using the *pairs()* function in *emmeans* package (Lenth, 2021) to perform pairwise comparisons of the estimated marginal means at each timepoint, within each condition. Results showed that the odds of a child performing better on the debugging task at post-intervention were significantly higher than at pre-test for children in the Intervention+ condition (odds ratio = 0.69, $z = -4.16$, $p < 0.0001$) and the Intervention condition (odds ratio = 0.75, $z = -2.54$, $p = 0.01$). On the other hand, odds of success on the debugging task did not improve for children in the Control condition at the end of the robotics intervention (odds ratio = 0.93, $z = -0.78$, $p = 0.44$). That is, in terms of debugging performance, children in the Intervention+ condition and Intervention condition were more likely to perform better at post-testing than pre-testing, whereas children in the Control condition showed no such improvement in debugging performance.

Prediction (Correct Trials)

Prediction task data was collected from 397 pupils (at pre- or post-testing). Data are visualised in Figure 5.13. Thirty-three children did not complete this task at both pre- and post-intervention and thus provided no data at all.

Figure 5.13

Portion of Correct Trials on the Algorithm Prediction task, Grouped by Condition and Time point.



Note: Group means are displayed in red.

Further analysis employed a GLMM to analyse prediction scores as a binary variable for each trial (0 = incorrect trial, 1 = correct trial). The interaction between Condition and Time was included in the model as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following *lme4* structure:

Prediction performance ~ Condition*Time + (1|TeacherID).

Table 5.2

GLMM Model Results: Fixed Effects.

	Model summary					
	β	*SE*	*z*	CI 95%	*p*	
Intercept	-1.52	0.16	-9.41	[-1.86, -1.19]	<0.001	
Intervention+ Vs Control	0.08	0.25	0.31	[-0.44, 0.59]	0.76	
Intervention Vs Control	0.14	0.22	0.62	[-0.32, 0.59]	0.53	
Time Point	0.16	0.10	1.56	[-0.04, 0.36]	0.12	
Intervention+ Vs Control (pre- Vs post intervention)	0.33	0.14	2.41	[0.06, 0.61]	0.02	*
Intervention Vs Control (pre- Vs post intervention)	0.25	0.15	0.71	[-0.04, 0.54]	0.09	.

Note: **.** p < 0.1, * p < .05. Intervention+ Vs Control (post-intervention), illustrates whether there is an interaction between Conditions (Intervention+ and Control) and Time (pre- and post-intervention).

The fixed effects from the model results are summarised in Table 5.2**.** The model revealed a significant interaction effect of Condition x Time on children's prediction performance. On average, improvements in children's scores in the Intervention+ condition were significantly larger than those in the Control condition (p = 0.02). Differences in improvements on the prediction task between the Intervention condition and the Control condition were marginal (p = 0.09). Finally, between the Intervention+ and the Intervention condition, there were no significant differences in improvements in task performance (B = 0.56, SE = 0.14, z = 0.58, CI 95% [-0.19, 0.04], p = 0.56).

The Condition x Time interaction was followed up using the *pairs()* function in *emmeans* package to perform pairwise comparisons of the estimated marginal means at each timepoint, within each condition. Results showed that the odds of a child performing better on the prediction task at post-test than at pre-test were significantly higher for children in the Intervention+ condition (odds ratio = 0.61, z = -5.24, p < 0.001) and the Intervention condition (odds ratio = 0.66, z = -3.86, p <0.001). On the other hand, odds of success on the prediction task did not improve for children in the Control condition at the end of the robotics intervention (odds ratio = 0.85, z = -1.56, p = 0.12). That is, in terms of prediction performance, children in the Intervention+ condition and Intervention condition were more likely to perform better at post-testing, whereas children in the Control condition were not.

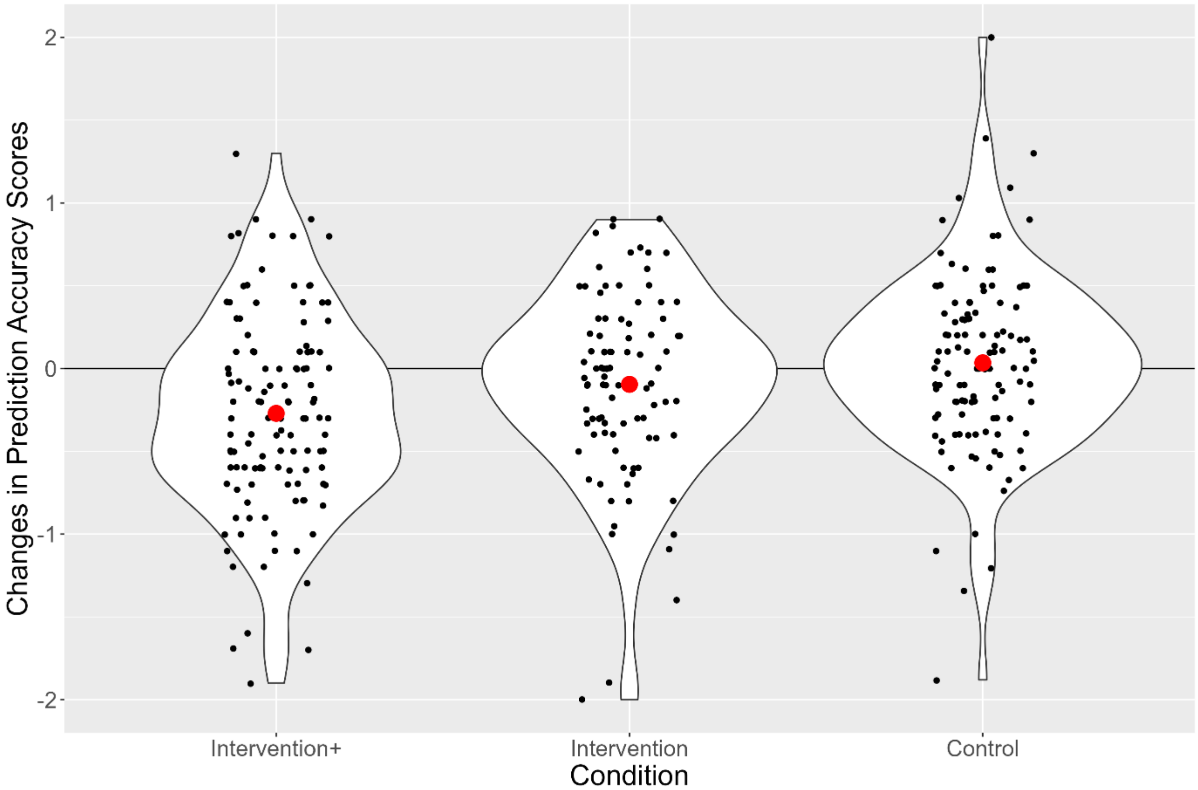
Prediction (Accuracy)

397 pupils were included in this analysis, having completed the prediction assessments at both pre-and post-intervention. If children provided multiple answers to a trial, this was recorded as missing data as an accuracy score could not be calculated. As prediction accuracy scores were not binomial (i.e., they could range from 0 – 3, 0 – 4 or 0 – 5 depending on the trial), performance was not analysed on a by trial basis like the analyses above. Instead, average scores were calculated for pre- and post-intervention assessments. To calculate average scores on this task, children's total accuracy score was divided by the number of trials they attempted. This was done to account for missed trials and missing data. Thus, values could range from 0 (i.e., all trials were answered correctly) to 4.2 (if attempts were furthest away from the correct answer for all trials). Lower scores indicated higher accuracy and thus better performance.

Difference scores were calculated by subtracting average post-intervention from average scores pre-intervention. Figure 5.14 illustrates changes in prediction accuracy scores pre- and post-intervention. Lower accuracy scores post-intervention indicated improved accuracy.

Figure 5.14

Changes in Prediction Accuracy Scores (Between Pre- and Post-Intervention), Grouped by Condition.



Note: Scores below 0 indicate improved accuracy scores post-intervention. Group means are shown in red.

This analysis employed a Linear Mixed-effects Model (LMM) from the *lme4* package in R (Bates et al., 2015). The current model included prediction accuracy as a continuous difference score (post-intervention score – pre-intervention score). Condition (Intervention+, Intervention and Control) was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following *lme4* structure:

Prediction accuracy ~ Condition + (1|TeacherID).

Table 5.3

LMM Model Results: Fixed Effects.

	Model summary					
	β	*SE*	*t*	CI 95%	*p*	
Intercept	0.02	0.08	0.38	[-0.11, 0.17]	0.71	
Intervention+ vs Control	-0.32	0.11	-2.80	[-0.05, -0.12]	0.02	*
Intervention vs Control	-1.12	0.11	-1.12	[-0.33, 0.08]	0.28	

Note: * p < .05. Intervention+ vs Control, illustrates whether there is an effect of intervention between Conditions (Intervention+ and Control).

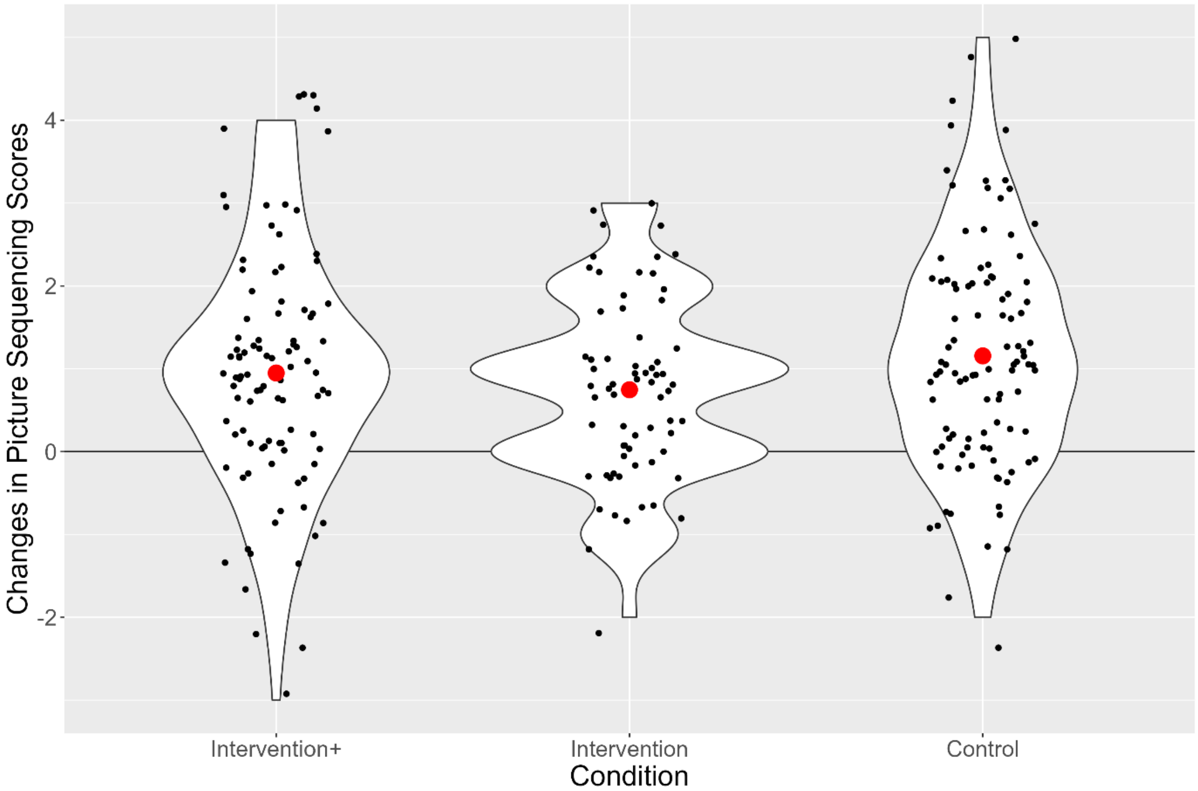
The fixed effects from the model results are summarised in Table 5.3. The model revealed a significant main effect of Condition on children's prediction accuracy scores. On average, children's answers in the Intervention+ condition were more accurate post-intervention than children's answers in the Control condition (p = 0.02). Prediction accuracy scores in the Intervention condition were not significantly different to scores in the Control condition (p = 0.28). Similarly, there was no significant difference between the Intervention+ condition and the Intervention condition (B = -0.20, SE = 0.12, t = -1.67, CI 95% [-0.42, 0.02], p = 0.12).

Picture-Sequencing

267 children were included in this analysis, having completed the sequencing assessments at both pre-and post-intervention. At each time point, total scores could range from 0 to 6. Difference scores were calculated by subtracting post-intervention scores from pre-intervention scores. Figure 5.15 illustrates changes in sequencing scores pre- and post-intervention.

Figure 5.15

Changes in Picture Sequencing Scores (Between Pre- and Post-Intervention), Grouped by Condition.



Note: Group means are displayed in red.

This analysis employed a LMM from the *lme4* package in R. The model included picture-sequencing as a continuous difference score (post-intervention score – pre-intervention score). Condition was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following *lme4* structure:

Picture sequencing ~ Condition + (1|TeacherID).

Table 5.4

*LMM Model Results: Fixed Effects***.**

	Model summary					
	β	*SE*	*t*	CI 95%	*p*	
Intercept	1.15	0.15	7.65	[0.87, 1.43]	< 0.001	***
Intervention+ vs Control	-0.18	0.22	-0.80	[-0.60, 0.25]	0.44	
Intervention vs Control	-0.41	0.23	-1.78	[-0.85, 0.02]	0.09	.

Note: **.** p < 0.1, *** < 0.001. Intervention+ vs Control, illustrates whether there is an effect of intervention between Conditions (Intervention+ and Control).

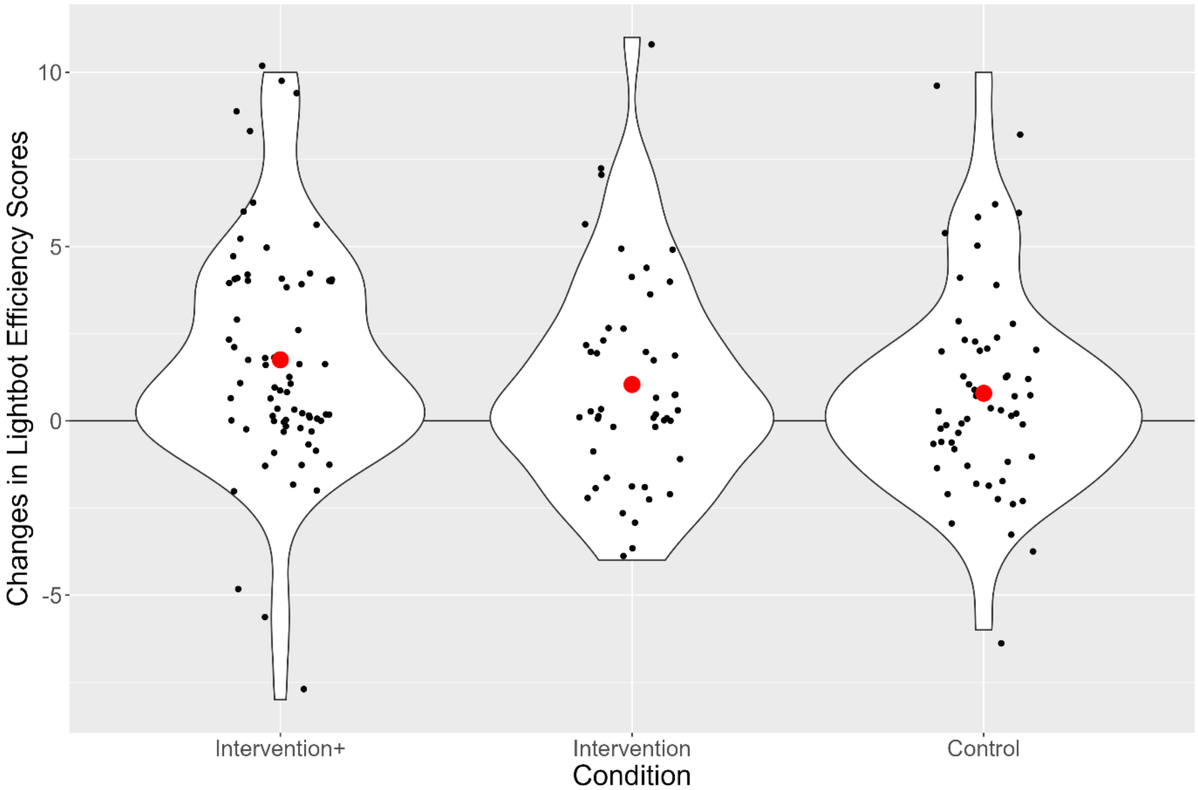
The fixed effects from the model results are described in Table 5.4. The model did not show a significant main effect of Condition on children's picture-sequencing performance. Neither the Intervention+ nor the Intervention condition significantly differed from the Control group. Similarly, scores in the Intervention+ condition did not differ significantly from the Intervention condition (B = 0.23, SE = 0.24, t = 0.97, CI 95% [-0.22, 0.69], p = 0.35).

Lightbot Programming

267 children were included in this analysis, having completed the Lightbot assessments at both pre-and post-intervention. Participants received 1 point for every level completed, but 2 points for each level completed efficiently (i.e., with the least number of blocks needed to program the virtual robot). At each time point, total scores could range from 0 to 9. Difference scores were calculated by subtracting post-intervention scores from pre-intervention scores. Figure 5.16 illustrates changes in Lightbot programming scores pre- and post-intervention.

Figure 5.16

Changes in Lightbot Jr Programming Scores (Between Pre- and Post-Intervention), Grouped by Condition.



Note: Group means are displayed in red.

This analysis employed a LMM from the *lme4* package in R. The model included Lightbot programming performance as a continuous difference score (post-intervention score – pre-intervention score). Condition was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following *lme4* structure:

Lightbot programming ~ Condition + (1|TeacherID).

Table 5.5

LMM Model Results: Fixed Effects.

	Model summary					
	β	*SE*	*t*	CI 95%	*p*	
Intercept	1.64	0.73	2.24	[-0.43, 2.03]	0.05	.
Intervention+ vs Control	-0.55	0.99	-0.55	[-1.01, 2.68]	0.59	
Intervention vs Control	-0.84	0.98	-0.85	[-1.47, 2.05]	0.41	

Note: **.** $p < 0.1$. Intervention+ vs Control, illustrates whether there is an effect of intervention between Conditions (Intervention+ and Control).

The fixed effects from the model results are described in Table 5.5. The model did not show a significant main effect of Condition on children's Lightbot programming scores. Neither the Intervention+ nor the Intervention condition significantly differed from the Control group. Similarly, scores in the Intervention+ condition did not differ significantly from the Intervention condition ($B = 0.55$, $SE = 0.99$, $t = 0.55$, CI 95% [-1.52, 1.41], $p = 0.59$).

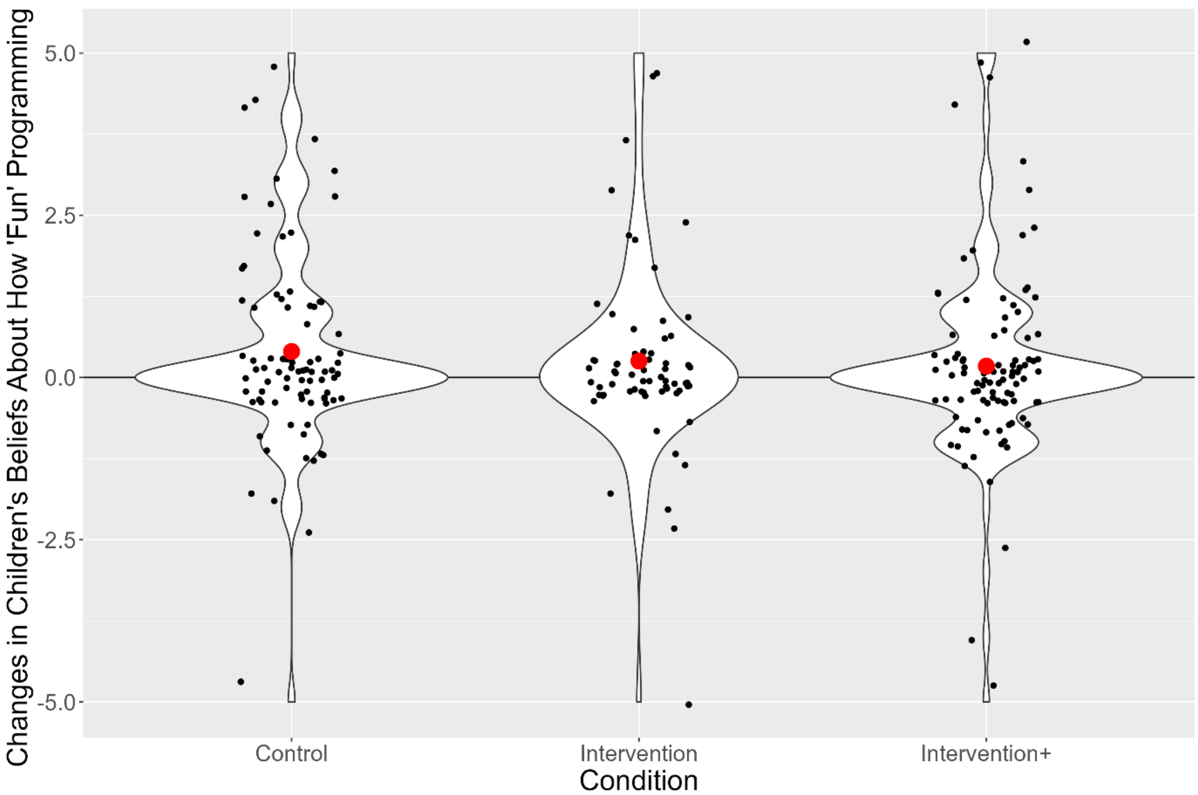
Pupil Beliefs

"Programming is Fun."

267 children were included in this analysis having completed pupil belief assessments at both pre- and post-intervention timepoints. Mean scores show that prior to the intervention, children believed programming to be fun. On average, children in the Intervention+ condition scored 5.16 on the 6-point Likert scale (SD = 1.19), the Intervention group averaged 5.26 (SD = 1.23) and finally the control group scored on average 5.14 (SD = 1.34). Figure 5.17 illustrates changes in programming belief scores pre- and post-intervention.

Figure 5.17

Changes in Programming Belief Scores (Between Pre- and Post-Intervention), Grouped by Condition.



Note: Scores above 0 suggest improved beliefs. Group means are shown in red.

This analysis employed a LMM and the model included children's belief scores (i.e., how fun they believed programming was) as a continuous difference score (post-intervention score – pre-intervention score). Condition was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following *lme4* structure:

Pupil belief score ~ Condition + (1|TeacherID).

Table 5.6

LMM Model Results: Fixed Effects.

	Model summary					
	β	*SE*	*t*	CI 95%	*p*	
Intercept	0.40	0.17	2.34	[0.09, 0.71]	0.04	*
Intervention+ vs Control	-0.21	0.24	-0.89	[-0.65, 0.20]	0.40	
Intervention vs Control	-0.14	0.25	-0.57	[-0.61, 0.32]	0.57	

Note: * $p < 0.05$. Intervention+ vs Control, illustrates whether there is an effect of intervention between Conditions (Intervention+ and Control).

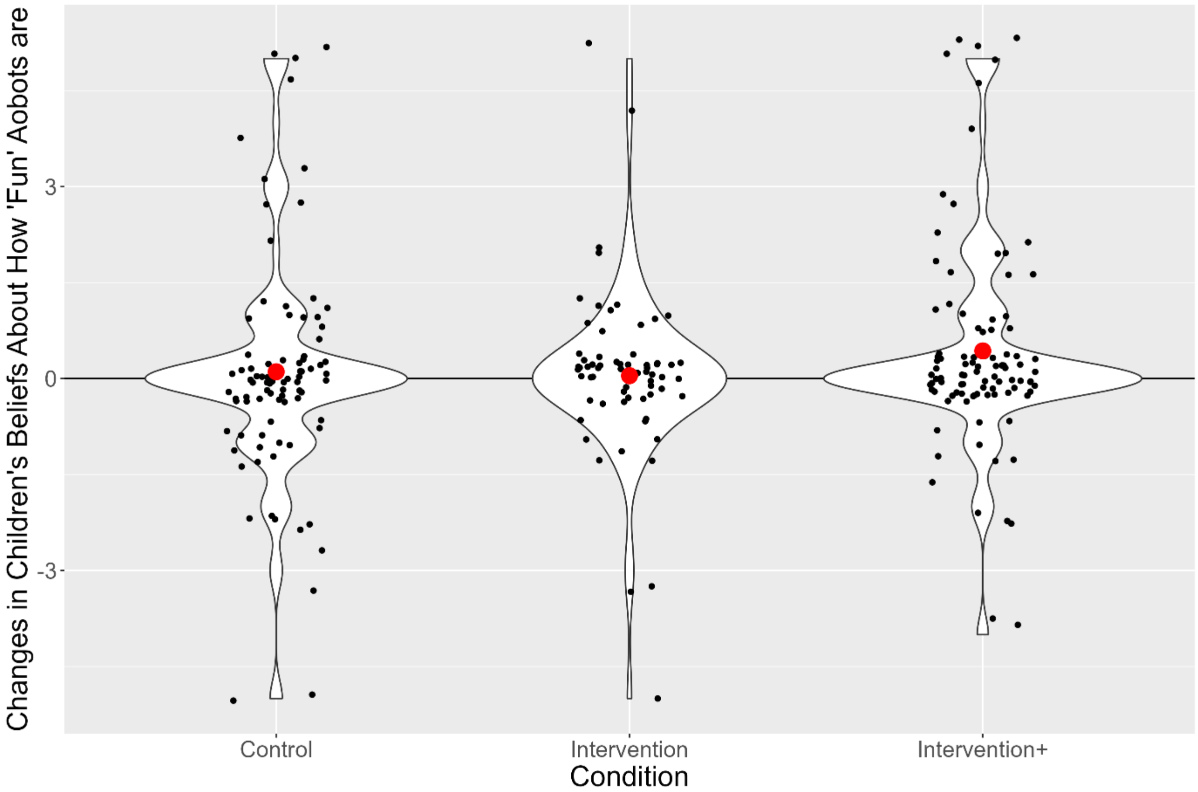
The fixed effects from the model results are described in Table 5.6. The model did not show a significant main effect of Condition on children's changes in programming beliefs. Neither the Intervention+ nor the Intervention condition differed from the Control group. Similarly, changes in belief scores in the Intervention+ condition did not differ significantly from the Intervention condition ($B = 0.55$, $SE = 0.99$, $t = 0.55$, CI 95% [-0.53, 0.38], $p = 0.59$).

"Robots are Fun."

Mean scores suggest that prior to the intervention, children believed robots were fun. On average, children in the Intervention+ condition scored 4.96 on the 6-point Likert scale (SD = 1.57), the Intervention group averaged 5.41 (SD = 1.07) and finally the control group scored on average 5.13 (SD = 1.44). Figure 5.18 illustrates changes in robot belief scores pre- and post-intervention.

Figure 5.18

Changes in Robot Belief Scores (Between Pre- and Post-Testing), Grouped by Condition.



Note: Scores above 0 suggest improved beliefs. Group means are shown in red.

This analysis employed an LMM, and the model included children's belief scores (i.e., how fun they believed robots were) as a continuous difference score (post-intervention score – pre-intervention score). Condition was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following *lme4* structure:

Pupil belief score ~ Condition + (1|TeacherID).

Table 5.7

LMM Model Results: Fixed Effects.

	Model summary				
	β	*SE*	*t*	CI 95%	*p*
Intercept	0.10	0.18	0.57	[-0.24, 0.45]	0.58
Intervention+ vs Control	0.33	0.25	1.31	[-0.14, 0.79]	0.23
Intervention vs Control	-0.06	0.27	-0.21	[-0.58, 0.45]	0.84

Note: Intervention+ vs Control, illustrates whether there is an effect of intervention between conditions (Intervention+ and Control).

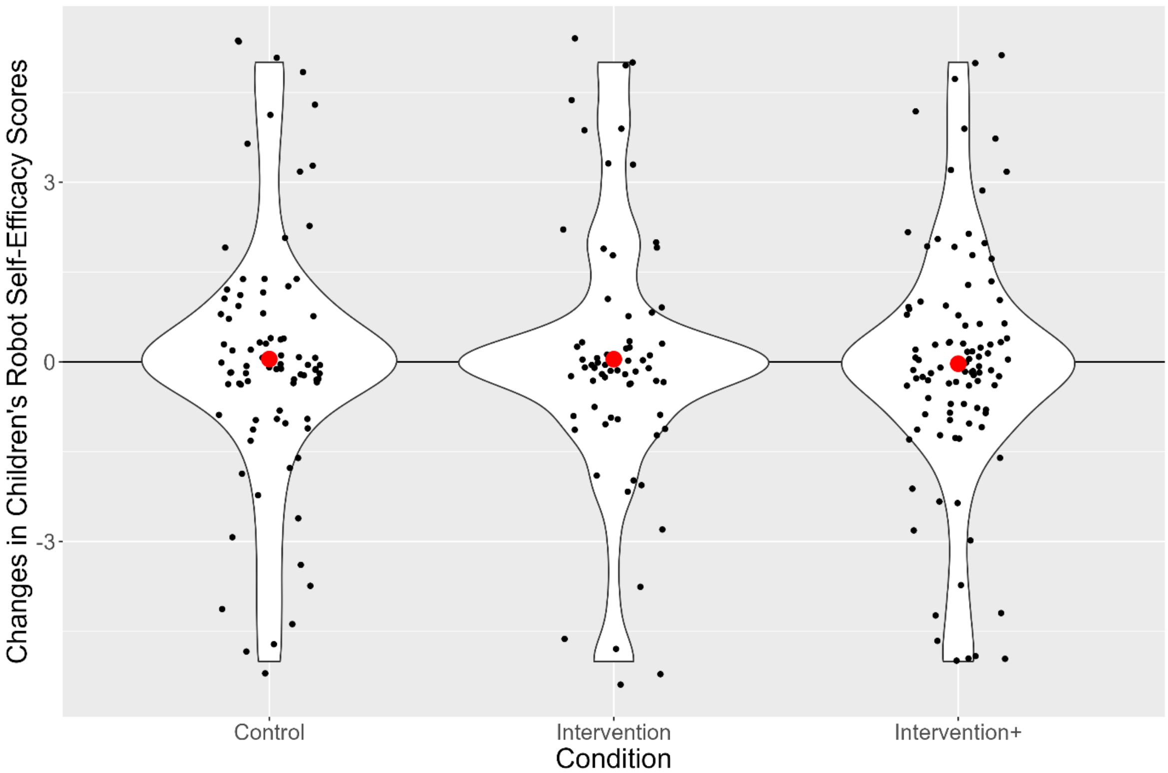
The fixed effects from the model results are described in Table 5.7. The model did not show a significant main effect of Condition on children's changes in robot beliefs. Neither the Intervention+ nor the Intervention condition differed from the Control group. Similarly, belief scores in the Intervention+ condition did not differ significantly from the Intervention condition (B = 0.38, CI 95% [-0.11, 0.89], SE = 0.26, t = 1.45, p = 0.17).

Robot Self-Efficacy

Mean scores suggest that, prior to the intervention, children were confident in their ability to use robots. On average, children in the Intervention+ condition scored their confidence 4.45 on the 6-point Likert scale (SD = 1.77), the Intervention group averaged 4.61 (SD = 1.81) and finally the control group scored on average 4.48 (SD = 1.83). Figure 5.19 illustrates changes in robot self-efficacy scores pre- and post-intervention.

Figure 5.19

Changes in Robot Self-Efficacy Scores (Between Pre- and Post-Intervention), Grouped by Condition.



Note: Scores above 0 suggest improved beliefs. Group means are shown in red.

A LMM was used and included children's confidence scores as a continuous difference score (post-intervention score – pre-intervention score). Condition was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following *lme4* structure:

Pupil confidence score ~ Condition + (1|TeacherID).

Table 5.8

LMM Model Results: Fixed Effects.

	Model summary				
	β	*SE*	*t*	CI 95%	*p*
Intercept	0.04	0.24	0.16	[-0.42, 0.49]	0.88
Intervention+ vs Control	-0.10	0.34	-0.30	[-0.76, 0.54]	0.77
Intervention vs Control	0.007	0.36	0.02	[-0.67, 0.69]	0.98

Note: Intervention+ vs Control, illustrates whether there is an effect of intervention between Conditions (Intervention+ and Control).

The fixed effects from the model results are described in Table 5.8. The model did not show a significant main effect of Condition on changes in children's self-efficacy scores. Neither the Intervention+ nor the Intervention condition differed significantly from the Control group. Similarly, belief scores in the Intervention+ condition did not differ significantly from the Intervention condition (B = -0.11, SE = 0.36, t = -0.30, CI 95% [-0.80, 0.58], p = 0.77).

Discussion

The aim of this study was to investigate the impact of a 6-week robotics curriculum on pupil's computational thinking (CT) skills (i.e., debugging skills, prediction skills, picture-sequencing skills, programming skills) and beliefs. Reviews of previous research have highlighted that previous robotics intervention studies lacked control groups, were typically delivered by researchers instead of teachers (Tselegkaridis & Sapounidis, 2022), and focused on children over the age of 8 rather than younger children (Mangina et al., 2023). This intervention study addressed each of these gaps in the literature. To do so, this study employed three experimental conditions: no intervention (Control), classroom intervention (Intervention), and classroom intervention plus teacher education (Intervention+). Using this experimental design, the study investigated whether the addition of a teacher education workshop impacted the success of a 6-week robotics curriculum. This section discusses the findings in relation to the two research questions outlined in the introduction and additional literature.

Can a 6-week Cubetto Curriculum Improve Children's Computational Thinking Skills, and How Do Outcomes Differ if Teachers Attend an Additional Education Workshop?

Previous research has shown that children as young as 4-years-old can demonstrate debugging skills with a programmable robot toy (Bers et al., 2014). Furthermore, Misirli and Komis (2023) engaged a sample of children (n = 526, aged 4 to 6 years) to investigate how they engaged with debugging practices whilst using a tangible robot for 2 to 3 weeks. They found that teaching children programming skills with a robot encouraged the development of children's debugging skills. These studies evidence the benefits of using ER with young children to promote debugging skills. However, neither study employed a control comparison group, thus the findings must be interpreted with caution as this limits the ability to draw definitive conclusions about the causal impact of robotics on the observed debugging outcomes.

The study presented in this Chapter included not only a control group, but also a pre- and post-intervention measure of children's debugging skills using a quantitative assessment. The findings of the current intervention study support these findings to a certain extent. The results of this study showed that a 6-week robotics curriculum significantly improved children's debugging skills, however, only when combined with a teacher education workshop (Intervention+ condition). Children whose teachers did not attend an additional teacher education workshop (the Intervention condition) did not show significant improvements in debugging skills when compared to the Control group.

With regards to prediction skills, a previous study (Slangen et al., 2011) asked children (10 to 11 years) to predict the behaviour of a programmable robot based on the algorithm they were given. Using qualitative methods, researchers observed children's learning and concluded that experience with robotics challenged children to make predictions before testing their assumptions. The results from the current intervention study support this suggestion that learning with educational robotics can encourage children's prediction skills. Furthermore, this study found improved prediction skills in younger children, and employed quantitative assessment measures, meaning skill development was tracked and could be easily compared across time points. However, like the results surrounding debugging improvements, the results of this study showed that combining a 6-week robotics curriculum with a teacher education workshop significantly improved children's prediction skills compared to the Control group and children whose teachers did not receive additional support (Intervention condition).

It is important to note that the results of this study did not show significant improvements on all pupil assessment measures. When compared to Control group performances, the 6-week intervention did not significantly improve children's picture-sequencing skills or pupil beliefs, even when combined with a teacher education workshop. Therefore, these results do not support or replicate those found in previous studies. For example, one study (Kazakoff et al., 2013) previously found that educational robotics could improve children's sequencing abilities following a 1-week long intensive robotics and programming curriculum compared to a control group who continued with their regular activities.

The current intervention study did not replicate these findings, this may be due to several reasons. With the current intervention being six times as long as the one implemented by Kazakoff and colleagues, it was more likely that teachers in the Control condition would also deliver sequencing activities to their pupils given the focus on sequencing in guidance from the New Curriculum for Wales (Hwb, 2024b). Attempts were made to track whether teachers in the Control group delivered programming and CT activities (including sequencing) to their pupils on a weekly basis, however, teachers did not complete these short forms when prompted. Informal discussions with some teachers during post-intervention data collection suggested that they used picture sequencing activities with their pupils during the intervention period, but there is no formal record of this. This may explain why differences in picture-sequencing performance were not found between the three intervention conditions. Instead, descriptive trends suggested that there was some improvement in picture-sequencing performance across all conditions.

The results of this study also showed no significant differences in programming-transfer between conditions, thus suggesting that completing the 6-week curriculum did not help children transfer knowledge to a new programming language. The wider literature appears to lack research that investigates whether children can transfer programming learning with educational robotics to alternative screen-based programming languages. However, the results of this study are consistent with past research that has suggested that young children may struggle when trying to transfer information between different modalities (Moser et al., 2015; Zack et al., 2009; Zack, et al., 2013). In their conceptual paper, Kallia and Cutts (2023) note that transferral of programming skills from a physical environment to a virtual environment should not be targeted with children under the age of 8 years.

It is worth noting, however, that although the linear mixed effect model did not find significant improvements in Lightbot programming performance in either of the intervention conditions compared to the Control condition, the descriptive statistics were encouraging. Average scores on the Lightbot Jr task suggest that children in the Intervention+ condition improved more than those in the Intervention and Control conditions. Although these differences were not significant, these trends were promising.

Overall, results showed that the 6-week robotics intervention combined with a teacher education workshop positively impacted children's debugging and algorithm prediction skills. However, the intervention did not significantly improve children's picture-sequencing skills or digital programming skills, even when combined with a teacher education workshop. Interestingly, the results for these CT measures collected can be split into two categories: (1) measures closely related to the Cubetto programming curriculum (i.e., debugging and prediction tasks) and (2) measures not explicitly linked to the Cubetto programming curriculum (i.e., picture sequencing and Lightbot Jr programming).

Theories of near versus far transfer effects (Barnett & Ceci, 2002) may explain why improvements were found for certain CT skills but not all. Such theories suggest that transfer of skills to other contexts is highly situation specific (Greeno et al., 1998). Thus, transfer of knowledge and skills from one context (i.e., physical Cubetto activities) to another (i.e., CT assessment measures) may depend on the similarity and overlap between contexts in which the skills were acquired and how they were presented later (Schunk, 2012). For instance, near transfer requires similar contexts and the performance of similar skills and strategies. In this study, both the debugging and algorithm prediction tasks assessed skills specifically targeted within the given Cubetto curriculum. On the other hand, far transfer occurs between contexts that are dissimilar and may require different skills or strategies (Perkins & Salomon, 1992). In this study, the picture-sequencing assessment and Lightbot Jr programming tasks provided different contexts to the Cubetto curriculum activities, as they were not explicitly linked to the robot. Thus, for transfer of knowledge to have occurred, this process would have had to have been an automatic and spontaneous process. However, a classic and repeated finding is that that such 'transfer' is difficult and does not happen spontaneously (Gutiérrez-Núñez et al., 2022; Hajian, 2019; Salomon & Perkins, 1989). Thus, it is likely that children require additional scaffolding to successfully transfer knowledge to an unfamiliar context.

It is important to note that the differences between the two intervention conditions suggest that improvements in debugging and prediction skills are not entirely due to familiarity with the Cubetto robot. If this were the case, we would have expected to find similar improvements in the Intervention condition. Instead, significant improvements were only found within the Intervention+ condition, suggesting that teachers may have played a large role in the success of the robotics intervention and the impact of the curriculum on children's CT outcomes.

Previous research has suggested that teachers' beliefs (i.e., value and self-efficacy beliefs) can impact pupil outcomes. Studies have shown that teacher self-efficacy beliefs in STEM subjects like science and mathematics may be associated with pupil's self-efficacy beliefs in these subjects (Midgley et al., 1989; Opperman et al., 2019; Stipek et al., 2001). Furthermore, additional findings have suggested that teachers with higher self-efficacy may be more effective at increasing pupil achievement (Klassen et al., 2021). For example, there is evidence that teacher self-efficacy beliefs are associated with children's achievement in subjects like mathematics (Ashton, 1983; 1986; Goddard et al., 2000). Thus, it is possible that the teacher education workshop may have improved teacher beliefs in the Intervention+ condition, which in turn had positive effects on children's learning. This leaves unanswered questions about the role of the teacher in the success of this intervention. The next chapter in this thesis (Chapter 6) explores these potential effects.

Can a 6-week Cubetto Curriculum Improve Children's Beliefs about Programming and Robotics, and How do Outcomes Differ if their Teacher Attends an Additional Education Workshop?

Past research has also assessed the effect of ER curriculums on children's beliefs about robotics. On the one hand, some have found that using robotics in education can improve children's attitudes (Zviel-Girshin et al., 2020). On the other hand, others have reported negative effects on children's attitudes (Hussain, 2006; Leonard et al., 2016). This intervention study did not improve children's beliefs about programming and robotics; however, this may be because children held positive beliefs at pre-intervention with most scores at ceiling prior to robotics exposure. Thus, they could not then increase their scores post-intervention. Alternatively, children's understanding of what programming is may have influenced their responses. For example, programming was described to children as *"Programming is when you tell a computer or a robot what to do"* (Master et al., 2017), and some children responded negatively, believing that *"telling someone what to do is bossy."* Additionally, regarding robots, some children relied on depictions of robots they had seen in books and media (e.g., *"robots blow stuff up"*) rather than recognising that they had been working with and learning about the Cubetto robot in lessons. In some cases, this impacted children's responses about how fun they perceived robots to be. However, it is important to note that children's beliefs were not negatively impacted by the intervention. Results showed that on average, children's confidence did not decrease after robotics exposure, and neither did their overall interest in robotics and programming. This is a positive finding to take away from this study given that some studies have previously found that robotics exposure can negatively impact children's self-efficacy (Leonard et al., 2016).

Methodological Reflections

Upon reflection, there are several methodological strengths and limitations that should be noted and may help guide future research. One strength of this intervention study was its use of pre- and post-intervention quantitative assessments that measured a range of CT skills. The design of these measures meant that children's learning could be tracked over the course of the intervention. Furthermore, by including a selection of trials which varied in difficulty, all CT assessments were accessible for children across the 4 to 7 age range. However, it is worth noting that the length of classroom testing sessions impacted data collected in some cases. The three paper-based assessments (i.e., debugging, prediction and visual perspective taking) were completed during back-to-back classroom sessions and thus took approximately 1.5 to 2 hours to complete. Although children were typically given a short break after one hour, having to sit in one place and complete written activities for this long was difficult, especially for some of the younger children in the sample (i.e., Reception class children). Furthermore, this group approach to testing may have been difficult for children used to small group learning (i.e., younger children). This may be why previous research has favoured hands-on, portfolio style assessments with robotics devices (Bakala et al., 2021). However, this style of assessment was not possible with such a large sample. The style of testing sessions delivered in this study allowed for a much larger sample size as data could be collected from ~30 children at once. Upon reflection, the benefits of assessing a larger sample outweighed the disadvantage of potentially nosier testing sessions.

Reflecting further on the methods used, collecting more data about individual differences would have been helpful. For instance, some schools were more ethnically diverse than others, meaning there were increased numbers of children for whom English was not the primary language spoken at home. This may have impacted children's understanding of the assessment tasks and thus future studies should endeavour to collect this information. However, it is worth highlighting some of the positive reflections collected from teachers with regard to how children with additional learning needs (ALN) engaged with the robotics curriculum. One teacher (a Reception Intervention+ teacher) shared,

> *"Cubetto really engaged our ALN child. Even when his group were not using Cubetto he wanted to sit and watch Cubetto move along the board. This is very unusual as he often cannot sit still and concentrate for more than a few minutes at a time."*

Similarly, another teacher (Year 2, Intervention+) reflected on how the Cubetto robot engaged a child for whom English was not their primary spoken language. They shared,

> *"The highlight of this lesson was when I worked with a group to programme Cubetto and sat next to a child with Additional Learning Needs. This child struggles to communicate in English, to write and access the learning of most of his peers. However, he was amazing at programming Cubetto, telling the other children whether to select right or left (he knew the difference between right and left), often correcting them and successfully navigating Cubetto through the maze. He knew which blocks to choose for each command. He was noticeably excited and animated and spoke more English than I have previously heard."*

Children with additional learning needs were not included in the analyses of intervention data as they either did not attempt the assessment measures at all or did not complete them individually. Instead, these children had additional support from teaching staff to complete answer booklets, thus we could not guarantee that responses were free from teacher influence. However, these reflections suggest that learning with the Cubetto robots positively impacted their learning experiences. Further research is needed to investigate how robotics can be used to improve learning outcomes in pupils with additional learning needs using an intervention design like the one used in this Chapter. Research aimed at interventions for children with additional learning needs is essential to ensure that educational programs address the diverse requirements of all learners, potentially leading to more inclusive educational practices.

Conclusion

In conclusion, this intervention study aimed to investigate the impact of a 6-week robotics curriculum on pupils' computational thinking skills (i.e., debugging skills, prediction skills, picture-sequencing skills, digital programming skills) and beliefs. Several aspects of this study's design were unique and innovative. This study employed three experimental conditions: no intervention (Control), classroom intervention (Intervention), or classroom intervention plus teacher education (Intervention+). Using this experimental design, the study investigated whether the addition of a teacher education workshop impacted the success of the 6-week robotics curriculum. This design sets this intervention apart from previous research as past robotics intervention studies have lacked control groups, were typically delivered by researchers instead of teachers (Tselegkaridis & Sapounidis, 2022), and focused on children over the age of 8 rather than younger children (Mangina et al., 2023).

The results of this study showed that a 6-week robotics intervention combined with a teacher education workshop significantly improved children's debugging and prediction skills, but not their picture-sequencing, digital programming skills or beliefs. These findings raise interesting questions about the role of the teacher in the success of the robotics curriculum as significant improvements were only found within the Intervention+ condition. Past research suggests that teacher beliefs are often associated with children's learning outcomes (Klassen et al., 2021). Thus, it is possible that the teacher education workshop may have improved teacher beliefs in the Intervention+ condition, which in turn had positive effects on children's learning. The next chapter in this thesis (Chapter 6) explores these potential effects. In the second half of this study, I investigated the impact of the intervention on teacher beliefs before then exploring whether teacher beliefs appeared to be associated with children's learning outcomes.

Chapter 6. Exploring the Effects of Robotics Intervention on Teachers' Beliefs.

Introduction

In previous chapters, I have summarised how curriculum guidance from the Welsh Government has highlighted the importance of developing children's digital skills (Hwb, 2018; Hwb, 2024a). For example, the Digital Competence Framework highlights *"data and computational thinking"* as a key learning area within early primary education. Curriculum guidance for teachers defines computational thinking (CT) as *"a combination of scientific enquiry, problem solving and thinking skills."* (Hwb, 2018). Furthermore, both school curriculums and academics have emphasised the notion that CT encompasses a broad set of analytic and problem-solving skills that can serve everyone, not just those working in technical roles or children learning with computer technologies (Barr et al., 2011; Barr & Stephenson, 2011; Computer Science Teachers Association, 2020; Lee et al., 2011, see Chapters 1 and 5 for more details).

In Chapter 5, I presented a school-based intervention which recruited early primary school teachers and their pupils to participate in a 6-week robotics curriculum in their classrooms. In that chapter, I explored whether the 6-week curriculum improved children's CT, programming skills and beliefs about programming and robotics. CT measures assessed debugging skills (e.g., identifying, and fixing errors in a sequence; Bers et al., 2019); prediction skills (e.g., the ability to use knowledge about the function of different programming instructions to anticipate what a sequence will do); and sequencing skills (e.g., creating a series of individual steps or instructions ordered to achieve a desired outcome; Brennan & Resnick, 2012). Pupil beliefs included their beliefs about how fun programming and robotics were, and how confident they were using robotics. I found that a 6-week robotics curriculum improved children's prediction and debugging skills when the curriculum was delivered by a classroom teacher who attended a teacher education workshop prior to curriculum delivery (see Chapter 5). These effects were only found when teachers received additional education prior to the intervention compared to pupils in a second condition who also completed the robotics curriculum but whose teachers did not receive additional education. This suggested that the teachers themselves and their experiences may have played an important role in the success of the robot intervention.

The education workshop delivered in this intervention study aimed to (1) improve teachers' beliefs about the importance of CT and programming education (i.e., their value beliefs); (2) improve their confidence in teaching these topics (i.e., their self-efficacy beliefs); (3) decrease their anxiety about teaching these topics and (4) increase their enjoyment when teaching CT, programming, and robotics content. Thus, it is possible that positive changes in teachers' beliefs following the workshop increased the effectiveness of the intervention on pupil's learning outcomes. Using data from the same intervention, this chapter explores whether additional teacher education improved teachers' beliefs relative to a group who delivered the robotics curriculum without additional education and a control group who did not receive education or deliver the curriculum. I also investigate whether any changes in teacher's beliefs relate to subsequent pupil outcomes.

Exploring the Impact of Teacher Beliefs

As well as looking at how teacher beliefs change over the course of this intervention, the simultaneous collection of pupil and teacher outcome data provided an opportunity to explore whether pupil outcomes were related to teacher beliefs. Bandura's influential theory has previously been used to argue that teachers with higher self-efficacy will be more effective at increasing pupil achievement (Klassen et al., 2021). For example, there is evidence that teacher self-efficacy beliefs are associated with children's achievement in subjects like mathematics (Ashton, 1983; 1986; Goddard et al., 2000). Thus, it is likely that teacher self-efficacy beliefs could impact pupil outcomes in other STEM subjects, including areas of CT (i.e., programming and robotics education).

Researchers have theorised that teacher self-efficacy may impact pupil achievement in two ways. Firstly, higher teacher self-efficacy may improve teachers' behaviours and classroom practices (Lauermann & Butler, 2021). For example, in the current study, teachers with higher self-efficacy may have felt more comfortable implementing the set robotics curriculum but may also have felt confident creating and integrating their own activities into their teaching to tailor the curriculum to their pupils' needs. Research has previously linked teacher self-efficacy with teacher effectiveness and instructional behaviours (Mok & Moore, 2019). For example, education research has found that high teacher self-efficacy can show positive links with classroom practices such as planning lessons that advance children's abilities, making opportunities for meaningful learning and effectively managing classroom behaviour (Chacon, 2005; Woolfolk et al., 1990). Secondly, it has been proposed that increased teacher self-efficacy may be transferred to pupils via role-modelling processes, whereby teachers' confidence is reflected by pupils and thus this increased self-efficacy in pupils then increases pupil persistence. This is then thought to have subsequent benefits for pupil achievement (Lauermann & ten Hagen, 2021).

Research investigating possible links between teacher beliefs and pupil achievement has provided mixed results. Meta-analyses have evidenced significant main effects of teacher self-efficacy on pupil achievement (Kim & Seo, 2018; Klassen & Tze, 2014). However, more recently, Jerrim et al., (2023) carried out their own large-scale international study with pupils aged 9 to 10 and 13 to 14. They found no evidence of a link between teacher self-efficacy and pupil achievement. In their paper, Jerrim and colleagues discuss possible reasons for their null findings. They highlighted findings from Lauermann and ten Hagen (2021) which suggested that the relation between teacher beliefs and pupil achievement is likely to be higher when pupils are younger, know the teacher well, and are faced with difficult tasks. The current intervention study employed samples of younger children (between the ages of 4 and 7 years), who participated in challenging tasks (robotics), with an intervention delivered by well-known teachers (their everyday classroom teachers, a dynamic distinctive to primary education). This intentional design, echoing Lauermann and ten Hagen's (2021) suggestions, was expected to increase the likelihood of detecting an effect of teacher beliefs on pupil outcomes.

Jerrim et al., also proposed that teacher self-efficacy may be a stable concept, thus limiting the potential for variation in longitudinal designs. The experimental design for this current study involved manipulations of teachers' experiences with programming and robotics education. For example, some teachers delivered a robotics curriculum to their pupils after attending a teacher education workshop. These education and classroom teaching experiences were manipulated to produce belief variations between teachers across the span of the intervention. These manipulations may have consequently increased the likelihood of finding a link between teacher beliefs and pupil achievement.

Current Understanding of Teacher Beliefs about Programming and Robotics

The findings from my focus group study (Chapter 2) and online survey (Chapter 3) suggest that, generally, Welsh primary school teachers hold positive value beliefs about the importance and relevance of teaching CT, programming, and robotics in early education. Thus, they appeared supportive of recent curriculum changes in these areas. Furthermore, another study (Khanlari, 2016) also found that a sample of Canadian primary school teachers ($n = 11$) held positive beliefs about the value of using robotics and the potential benefits for their pupils' learning. For example, when asked whether early primary school children were too young to understand and work with robotics, 64% of teachers surveyed believed pupil age was not an obstacle. Findings from my past survey research supported these findings with a larger sample (n = 68) as early primary education teachers believed their pupil's age was not a barrier to learning with educational robotics. These findings indicated teachers' positive value beliefs.

Previous research has found that low self-efficacy for programming and robotics is common in samples of primary school teachers (Khanlari, 2016; Ohashi et al., 2018; Ray et al., 2020). For example, in a survey study of 142 Japanese primary school teachers (Ohashi et al., 2018), only 4% of teachers reported they felt confident about teaching programming to their pupils. Similarly, survey findings from Khanlari (2016) illustrated that 82% of primary school teachers (n = 11 teachers in Canada) identified teacher confidence as a major obstacle to robotics use in the classroom. Findings from my focus group and survey studies (see Chapters 2 and 3) supported the notion that early primary school teachers typically held low self-efficacy beliefs when it came to teaching CT, programming, and robotics content. However, my survey results also suggested that, although most teachers were not confident in their ability to teach programming and robotics content at the time of questioning, they were confident that they could learn how to deliver these lessons. Thus, this intervention study aimed to provide practising teachers with the tools and experiences they required to teach CT, programming, and robotics lessons with confidence. This was done in two ways. Firstly, all participating teachers were provided with a collection of robotics resources which included a 6-week robotics curriculum and additional materials such as a booklet outlining guidance on how best to introduce the robot and curriculum to their young pupils. Secondly, some teachers in this study were invited to attend a teacher education workshop prior to delivering the robotics curriculum. Thus, this study aimed to explore whether improvements in teacher beliefs were best achieved via a combination of teacher education and in-classroom teaching experiences, or whether in-classroom experiences (with provided curriculum guidance and resources) were just as effective at improving beliefs. This study also employed a Control group to explore whether any changes occurred across time, without the workshop or curriculum experiences.

Improving Teachers' Beliefs

This teacher education workshop aimed to improve teachers' beliefs (i.e., value, self-efficacy, anxiety, and enjoyment beliefs) regarding teaching CT and programming with educational robotics. Several studies have utilised pre-test-post-test designs to explore changes in teacher beliefs about programming and robotics education and have demonstrated that teacher education workshops can improve teachers' beliefs in these areas (Castro et al., 2018;

Chang & Peterson, 2018; Kim et al., 2015). One study (Kim et al., 2015) recruited 16 pre-service primary school teachers (i.e., teachers who had not yet completed their teaching qualification) and had them complete a 3-week robotics and programming course. Through pre and post course surveys and interviews, it was concluded that the 3-week program improved teachers' motivation, enjoyment and interest towards programming and robotics. Research from Chang and Peterson (2018) suggested that even a single 2-hour workshop may improve teachers' beliefs. They delivered a 2-hour educational technology course to 59 pre-service primary school teachers. This session focused on how to teach CT, robotics, and programming to children aged 4 to 12. Based on written reflections from teachers, the authors concluded that the session increased teachers' understanding of CT and its teaching applications. They also believed it improved teachers' relevance beliefs. Research findings from these studies indicated positive shifts in pre-service teachers' beliefs after participating in teacher education workshops. However, it is important to interpret these findings cautiously, recognising the potential bias in self-report measures and the lack of control comparison groups. Moreover, the context of pre-service training may contribute to these positive outcomes.

It is important to also employ samples of practising teachers in intervention research due to their unique teaching experiences in comparison to pre-service teachers. To give one example, the support pre-service teachers receive during their pre-qualification training might create a somewhat artificial environment, as they are shielded from the full spectrum of challenges and responsibilities faced by experienced, full-time classroom teachers (Laker et al., 2008). Classroom dynamics and the evolving demands of day-to-day teaching may significantly shape educators' beliefs and practices, and pre-service teachers may not yet have had the opportunity to fully navigate these complexities. This limited exposure to real-world teaching scenarios may bias their perspectives on incorporating robotics into their (future) teaching practices.

One study from Bers, Seddighin and Sullivan (2013) has investigated whether a teacher workshop could improve programming and robotics beliefs in a sample of fully qualified teachers ($n = 32$). Changes in teacher beliefs were measured following a 3-day intensive teacher education program. This program aimed to help teachers understand how new robotics technologies could be used with young children and integrated within subjects fundamental to early childhood education. Content was delivered to teachers via a combination of lectures, small group discussions and hands-on learning with educational robotics. Using surveys and semi-structured interviews, this study found that participation in these workshops resulted in increased self-efficacy and improved beliefs towards using technology in teaching.

The current robotics intervention project continued to contribute to this literature by investigating changes in beliefs amongst a sample of fully qualified classroom teachers. Collecting measurements of teacher beliefs before and after the administration of teacher education programs was important for assessing the effectiveness of these courses as pre-workshop measurements provided a baseline for teachers' beliefs. These baseline measurements helped determine whether high self-efficacy and positive value beliefs post-workshop were due to intervention manipulations or due to individual differences (i.e., teachers' positive beliefs beforehand and sample bias). Furthermore, in contrast to the studies outlined here, my intervention study also employed a control group of teachers who did not complete a robotics intervention with their pupils, nor did they attend a teacher workshop.

To create an effective teacher workshop for this intervention study, insights from previous chapters were used to aid its design. Firstly, my focus group and survey chapters (Chapters 2 and 3) highlighted the importance of workshop content and goals that focused on teacher *education* rather than teacher *training*. It has been argued by Stephens et al., (2004) that the terms 'education' and 'training' signify different pedagogical approaches. Comparing these two approaches to teacher development, they argued that teacher 'education' has a broad focus on intellectual and personal development and typically includes a combination of theoretical and practical learning to encourage reflection, analysis, and a deeper understanding of content. This may help teachers apply knowledge to a diverse range of contexts as they feel more confident in adapting instructional strategies to suit the specific needs of their classrooms. On the other hand, teacher 'training' typically emphasises the development of practical skills that are required to complete specific tasks (Stephens et al., 2004). Thus, training is characterised as involving drills and simulations to ensure that teachers can execute specific tasks. This approach may make it more difficult for teachers to apply the new knowledge within their own classroom. Thus, the teacher workshop in this study was designed with the principles of teacher education in mind.

Furthermore, in earlier studies, I explored teachers' perceptions and experiences of teacher education in the areas of programming and computing. These discussions highlighted several limitations of teacher education opportunities in early primary school years, specifically relating to CT, programming, and robotics education. The recommendations that emerged from these studies aligned with the principles framed by teacher *education* and included (1) providing robotics and programming knowledge that is easily applicable within early childhood classroom contexts; (2) teaching content that focuses on interdisciplinary learning and (3) providing structured support while adopting experiential learning approaches. These recommendations from Welsh primary schools in my previous chapters reflect theoretical recommendations from past research (i.e., Desimone, 2009).

Developmentally Appropriate Content

Firstly, teachers in my previous studies (see Chapters 2 and 3) highlighted the lack of education opportunities available to early primary school teachers within the field of programming and educational technologies. It seemed that past programs have lacked developmentally appropriate content for children under the age of 8 years. As a result, teachers have not been able to transfer this knowledge into the classroom and into practice. Thus, the workshop in this study featured content and learning activities specifically designed for children aged 4 to 7 years. To further cater to lower primary school educators, this workshop introduced programming and robotics knowledge using simplified, jargon free language to ensure the content was accessible for both inexperienced teachers and their pupils. This session also provided teachers with time and space to discuss ideas for age-appropriate materials and activities for their own pupils (considering their abilities and interests). Research suggests that improving the relevance of training would likely improve classroom outcomes (e.g., implementation of the robotics curriculum; Axtell et al., 1997).

Interdisciplinary Learning

In my previous studies, teachers also highlighted the importance of cross-curricular learning and shared their thoughts on integrating programming and robotics with other subject areas. Thus, another focus of this workshop was to demonstrate how teachers could combine robot activities with other classroom subjects (e.g., mathematics, literacy, art) to teach programming, not as an individual skill, but instead as an approach to developing broader knowledge and general CT skills. This aimed to positively impact teachers' relevancy beliefs as they explored the versatility of programming education and robotics technologies (Greifenstein et al., 2021).

Experiential Learning

Finally, findings from previous chapters have also suggested that, to ensure effectiveness, teacher education programs should provide experiential learning experiences.

This is likely to positively impact teachers as active learning is more impactful than passive learning (Burke & Hutchins, 2007). In this workshop, teachers were given free time to use the robot playsets themselves. This hands-on learning opportunity allowed them to learn how the playsets worked and to share ideas for lesson planning. These experiential learning opportunities aimed to not only increase teacher confidence (Konen & Horton, 2000), but also increase the likelihood of education transfer into the classroom. As discussed above, Bers et al., (2013) included hands-on learning activities in their teacher education workshop (along with lectures and small group discussions) and afterwards found an improvement in teachers' self-efficacy and beliefs about using robotics technologies in the classroom. These findings support the inclusion of hands-on robotics tasks within teacher education workshops.

In the current study, teachers who attended the education workshop were not the only teachers who had experiential learning opportunities with the educational robot used in this study. Although teachers in a second experimental condition did not attend a workshop, they still delivered a set 6-week robotics curriculum to their pupils independently. Study designs that have included an additional element of experiential teaching inside of classrooms have proved to be advantageous. Ensign (2017) not only delivered a robotics workshop to teachers but also encouraged teachers to implement robotics activities within their classrooms. After comparing pre and post questionnaire responses, results showed that education workshop and teaching experiences increased teachers' confidence and improved their value beliefs as they reported that educational robotics improved pupils' motivation and engagement.

Few other studies appear to investigate changes in teacher beliefs following classroom-based robotics interventions that provide experiential teaching opportunities for teachers. This is likely because, when investigating the impact of robotics interventions, researchers have typically focused on pupil outcomes rather than teacher outcomes. This is evidenced by the number of recent literature reviews that each aim to summarise pupil centred studies (e.g., Bakala et al., 2021; Tselegkaridis & Sapounidis, 2022; Xia & Zhong, 2018). The lack of review papers surrounding teacher outcomes following robotics interventions appears to reflect the wider literature. The lack of teacher-led interventions will have contributed to the lack of teacher-centred robotics research. For instance, few studies have used classroom teachers to deliver robotics intervention curriculums to early primary school pupils, instead relying on researchers and research assistants to deliver robotics content (e.g., Sullivan & Bers, 2015; Sullivan & Bers, 2016; Sullivan et al., 2013; Strawhacker & Bers, 2015). Thus, researchers would not have had the opportunity to analyse and explore potential changes in teacher beliefs.

Engaging teachers in intervention research as active participants is important as they are responsible for delivering these lessons to their pupils in the long term. By doing so, this study sought to administer a robotics intervention that had the potential of benefitting teachers and their pupils beyond the timeline of the intervention itself.

Current Study

In summary, this study recruited practising early years teachers to deliver a 6-week robotics curriculum to their pupils within their own classrooms. To do so, this study utilised several experimental conditions. As part of one condition (Intervention+), a sub-sample of teachers attended a teacher education workshop prior to delivery of the robotics curriculum. Discussions and findings from previous chapters were used to aid the design of the teacher education workshop in this study. Taking this tailored approach is likely to improve the effectiveness of this teacher intervention, and thus positively impact teacher knowledge, teacher confidence (Konen & Horton, 2000) and pupil achievement (Sims et al., 2021).

Teachers in a second condition (Intervention) delivered the prescribed robotics curriculum without attending a workshop, thus any changes in teacher beliefs were likely due to personal experiences from teaching robotics sessions instead of any formal education of the teachers themselves. A sample of teachers also participated as a Control group, a design strength that previous studies lack (e.g., Bers et al., 2013; Jaipal-Jamani & Angeli, 2017; Kim et al., 2015). This is a limitation highlighted in a review paper by Tselegkaridis and Sapounidis (2022). They argued that control groups are necessary for accurately assessing the effectiveness of robotics interventions. Teachers in the current Control group also completed teacher belief questionnaires despite not receiving a robotics curriculum nor additional teacher education. Including a control group allowed for further investigations into the effectiveness of the intervention and its impact on teacher beliefs and pupil outcomes.

To measure changes in teacher beliefs, this study combined design elements from past studies discussed above. For example, teacher beliefs were assessed at multiple time points throughout the intervention. Firstly, teachers completed a questionnaire at the start of the intervention, prior to any teacher education or robotics lessons within the classroom. Secondly, all teachers were asked to complete the beliefs questionnaire after teachers in the Intervention+ condition had attended a teacher education workshop. Finally, questionnaires were administered again at the end of the intervention (i.e., after the robotics lessons concluded). By collecting this data at multiple timepoints, this study aimed to provide detailed insights into how and when teachers' beliefs changed over the course of the intervention (i.e., after teacher education versus after teachers delivered the robotics curriculum). This data was also combined with pupil data from the previous chapter to explore the effects of teacher beliefs on pupil learning outcomes.

Thus, this chapter answers the following research questions:

1. Can a 6-week robotics curriculum improve teachers' beliefs regarding programming and robotics and how do changes vary as a result of whether teachers attended an additional teacher education workshop?
2. Do teacher beliefs (post-intervention) relate to changes in pupil outcomes following a 6-week robotics curriculum?

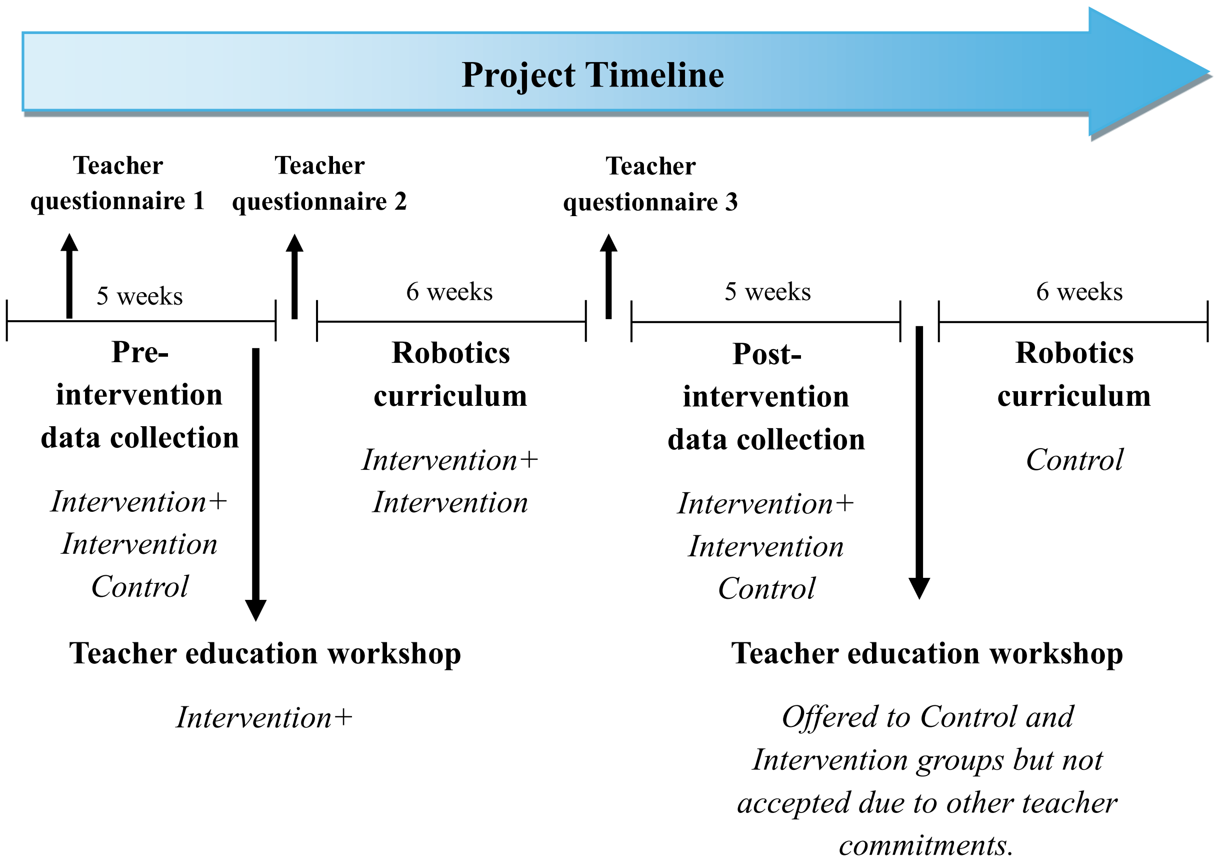
Methods

Design

As described in Chapter 5, the current study utilised a pre-test post-test design, where teachers and classrooms were divided into three different conditions: Intervention+ (pupil robotics curriculum plus teacher education workshop), Intervention (pupil robotics curriculum only) and Control (see previous chapter for more detail). Teachers and pupils in the Control condition were not provided with robotics equipment and content until after data collection had concluded (see Figure 6.1). This chapter investigates whether the teacher education workshop and/or a classroom intervention altered teacher's beliefs towards teaching programming and robotics lessons. Furthermore, it explores the effects of teacher beliefs on pupil's performance on outcome measures described in the previous chapter.

Figure 6.1

Illustration of the Project Timeline.



Participants

Nineteen foundation phase classrooms (a mix of Reception, Year 1, and Year 2) participated in this study from seven schools across South Wales (for more details, please see Chapter 5). Each school was assigned to one of the three conditions described above. Allocation was not completely randomised due to the size of the schools taking part. Two of the participating schools were particularly large (each with ~180 pupils spanning Reception to Year 2), so all classrooms within one of these large schools were randomly assigned to the Intervention+ condition and all classrooms within the other large school were assigned to the Control condition. Thus, the five remaining schools were assigned to the same condition to equate numbers in the Intervention condition. The decision was made to assign whole schools to the same condition as it would not have been possible to control the spread of information between teachers if individual classrooms within a school were assigned to varied conditions. Specifically, it would have been difficult to ensure teachers who attended the education workshop did not pass information onto teachers who did not attend the workshop.

Across these schools, 17 teachers participated (demographic information displayed in Table 6.1). Seventeen teachers signed up to participate, however two Control group teachers did not complete any of the teacher questionnaires during the intervention, thus minimal information about these teachers was acquired. Fewer teachers than classrooms participated in this study as some teachers taught multiple classes of pupils. For example, in the Intervention+ condition, two teachers delivered the robotics curriculum to two different classrooms. However, it is important to note that these teachers were known to these pupils as they would regularly deliver some lessons to these classes. Approximately 550 pupils completed the robotics curriculum and data was collected from 430 children with parental consent. For more details, please see previous Chapter 5.

Table 6.1

Teacher Demographics, Grouped by Condition.

Teacher	Gender	Age	Year group taught	Additional job role(s)	Years experience	Highest qualification level	Past programming experience
Intervention+							
1	Female	24	Reception	No	1	PGCE	No
2	Female	23	Year 1	No	1.5	BSc with QTS	No
3	Female	43	Year 1	Foundation Phase Leader	22	BSc with QTS	No
4	Female	47	Year 2	No	1		No
Intervention							
1	Male	43	Year 2	Lead for Welsh and Health and Wellbeing	17	PGCE	No
2	Female	47	Reception	Leader of Teaching and Learning, and Mathematics.	5	PGCE	No
3	Female	38	Year 1	No	18	BSc	Yes
4	Female	29	Reception	ICT Lead	6	PGCE	No

5	Male	42	Year 2	CLA Lead	4	BSc	No
6	Female	44	Year 1 and Year 2 (mixed classroom)	Science and Technology Coordinator	23	BSc with QTS	No
7	Female	-	Reception and Year 1 (mixed classroom)	No	15	PGCE	No
Control							
1	Female	41	Year 1	No	15	PGCE	No
2	Female	-	Year 1	No	-	-	-
3	Female	28	Year 2	Assessment Lead	6	BSc	Yes
4	Female	50	Year 2	Senior Student Mentor	26	PGCE	No
5	Female	-	Reception	-	-	-	-
6	Female	-	Reception	-	-	-	-

Note: Children Looked After (CLA), Postgraduate Certificate in Education (PGCE), Bachelor of Science (BSc), Qualified Teacher Status (QTS), - indicates missing information.

Procedure and Materials

Teacher Education Workshop

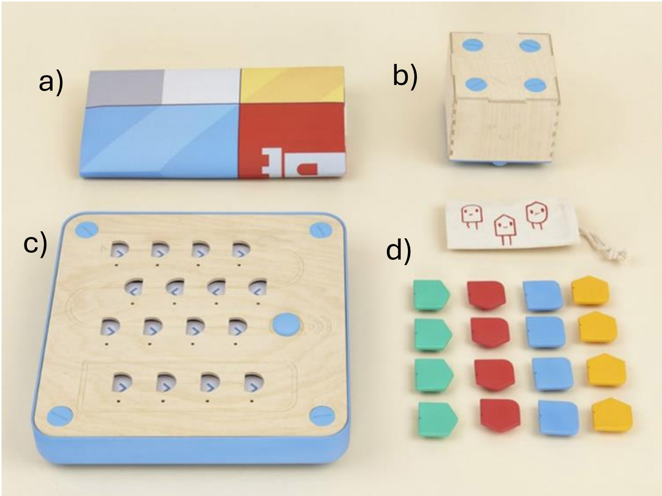
Classroom teachers allocated to the Intervention+ condition ($n = 4$) attended a robotics workshop prior to the 6-week robotics curriculum. This workshop was delivered by me (as Principal Investigator) and a secondary researcher. The content of this workshop was determined by the outcomes of a focus group and online survey (see Chapters 2 and 3), both of which employed samples of Welsh foundation phase teachers (those teaching children aged 4 to 7 years). As a result, this 3-hour session had three main components: (1) providing robotics and programming knowledge that is easily applicable within early childhood classroom contexts; (2) teaching content that focuses on interdisciplinary learning and (3) providing structured support while adopting experiential learning approaches.

Firstly, via a PowerPoint presentation (see Appendix G), teachers were introduced to key terms often used when discussing programming education (e.g., ‘debugging’, ‘algorithms’, ‘computational thinking’). To best support lower primary educators, terms were defined using jargon free, non-specialist language to ensure the content was accessible for both inexperienced teachers and their pupils. Additionally, real-life examples were given to help teachers relate programming terminology to everyday activities. For example, algorithms were presented as a series of ordered steps taken in sequence to solve a problem or achieve an end goal (i.e., brushing teeth). This presentation also introduced several methods used to teach programming to young pupils (as found in earlier focus group and survey research, see Chapters 2 and 3). These included unplugged programming, screen-based programming, and educational robotics.

Teachers were then introduced to the Cubetto robot (see Figure 6.2, www.PrimoToys.com) that would be used in this intervention. In this portion of the workshop, teachers learned about how the Cubetto robot could be used to introduce programming concepts in early childhood without the use of a screen. Teachers were also given the opportunity to gain hands on experience with the Cubetto kit. As a group, they explored how the robot worked, what the various tokens did, and how they could code the robot to move across a map. To use Cubetto, teachers wrote their own algorithms using its block-based programming language. Each coloured token placed into the board provided Cubetto with a specific instruction (i.e., move forwards, left or right). This part of the workshop also explored how to introduce the Cubetto robot to pupils. The group discussed how to set up the equipment in the classroom (i.e., whole class introductions or small group exploration), and techniques to help pupils learn the function of each movement token.

Figure 6.2

Cubetto Playset.



Note: Each set comes with a floor map (a), a Cubetto robot (b), an interface board (c) and a range of movement tokens (d).

Finally, teachers were prompted to think about how they might teach programming alongside other classroom subjects, working together to generate ideas for cross-curricular learning. In my previous studies (see Chapters 2 and 3), teachers highlighted the importance of cross-curricular learning and shared their positive thoughts on integrating programming and robotics with other subject areas. Thus, the final focus of this workshop was to demonstrate how teachers could combine robot activities with other classroom subjects (e.g., mathematics, literacy, art) to teach programming, not as an individual skill, but instead as an approach to developing broader knowledge and general computational thinking skills. Thus, teachers in this session viewed example lesson plans (designed by Primo Toys and an Early Years teacher, see below for more detail) and discussed as a group how they could adapt the plans to fit various themes and subjects, whilst ensuring the focus of the programming task (i.e., writing algorithms or debugging) remained the same.

Robotics Curriculum

Teachers involved in this study were provided with Cubetto robotics kits to use in their classrooms. To accompany this equipment, all teachers were provided with an introductory guide (see Appendix H) and a collection of predetermined lesson plans (all written by Primo Toys; see Appendix E).

Introductory Guide. At the start of this study, all teachers in the Intervention+ and Intervention conditions were given a Teaching Guide (written by Primo Toys). This document provided guidance on how to introduce Cubetto within their classroom. It introduced the Cubetto playset, how it worked and how they could begin using the toy with their pupils as a learning tool. Guidance included how to introduce Cubetto to pupils – they were encouraged to highlight that Cubetto could not think for him/herself and so needed to be programmed by the child, just like any other machine. They were also advised to introduce Cubetto's interface board as a remote control that children could use to send instructions to Cubetto. Finally, blocks could be introduced as the directions Cubetto followed when inserted into the board and sent to Cubetto by pressing the button (different blocks = different instructions). This guide also described concepts like computational thinking, algorithms (e.g., "*Sets of precise instructions that form a program. Cubetto's Blocks are a physical representation of an instruction that combine to create a program"* Primo Toys, p. 5) and debugging *("The instructions are laid on the Board. Fixing mistakes is as easy as swapping a block if Cubetto doesn't arrive where he needs to. This is called debugging"* Primo Toys, p. 5).

Lesson Plans. Teachers were provided with a collection of 53 lesson plans aimed at children in early primary education. These lesson plans were designed by Primo Toys and an Early Years teacher. Six of these lesson plans were made compulsory for this intervention study and teachers were asked to deliver at least one programming lesson a week, for six weeks. Specific lesson plans were selected by the research team to ensure that there was a baseline for what teachers were delivering in lessons, and that this was consistent across all schools participating in the intervention. Furthermore, plans were chosen to ensure that teachers covered a range of programming skills (e.g., sequencing, prediction and debugging). Teachers were allowed to deliver additional lessons to their pupils if they wished, and they were asked to keep a record of this. Records showed that no teachers reported using the additional lessons plans during the intervention and instead only delivered the six compulsory plans.

During each lesson, children completed a range of activities with a Cubetto robot (see Chapter 5 for descriptions of each lesson). These exercises included writing algorithms to move Cubetto robots around the map to collect various objects. Children were also taught how to read algorithms and think about where these instructions tell Cubetto to go. Additionally, children gained experience writing algorithms for their peers who would then attempt to debug the sequence. These activities completed by pupils were described in more detail in the previous chapter. Here, I focus on explaining the structure of the lesson plans provided to teachers.

Each lesson was designed to take between 1 to 2 hours to complete and each plan provided a clear breakdown of the content covered in each session (see Figure 6.3). For example, all resources needed for the session were listed to aid teachers' lesson preparations. Printable resources were provided to teachers in a supplementary materials document. The plans also clearly highlighted specific computing curriculum objectives, learning outcomes, targeted computational thinking skills and key vocabulary for pupils. Plans then provided step-by-step guidance for Cubetto activities, both group and independent learning exercises. There were also additional, more challenging activities for those seeking further learning. All activities were clearly linked to a cross-curricula learning area (i.e., maths, literacy, art). Plenary and assessment guidance was also included in the plans.

Figure 6.3

Example Lesson Plan (Compulsory Lesson #4).

Learning with Cubetto - 6 week robotics program PRIMO

Lesson 4: Cubetto's Quest (1 of 2)

Cross-curricula Area: Art

NC Objectives	Outcomes	Resources Needed	Prep Needed	Resources Provided	Key Vocabulary
To use logical reasoning to predict behaviour of simple programs	• I can predict what a program will do • I can draw a treasure map	• A4 paper with large squares (3x3)	• Check batteries • Choose a square for the treasure to be buried in and mark it on a laminated map (keep it secret!) • Write the algorithm needed to get there	• Primo maps	• Predicting • Algorithm • Clue • Program

Computational thinking concept

Logic

Computational thinking approach

Perseverance

Teacher-led Introduction

1. Ask the children to close their eyes and think about the most precious thing they own (it doesn't have to be an object!).
2. Tell the children that Cubetto has lost something very important to him and is very sad.
3. Explain that today the children will be trying to help Cubetto find it by predicting where an algorithm will take you.
4. Tell the children that you know where Cubetto's treasure is hidden and show the clue (the algorithm) to find it.
5. Ask: Looking at the algorithm, where do you think the treasure is? Why do you think that? How are you predicting where it is?
6. Ask for a volunteer to program Cubetto with the algorithm you showed the class.
7. Encourage the children to discuss whether they predicted correctly and why. Explain that computing often involves trying things lots of times before we get things right and this it is very important to be patient.
8. Model marking on a map where the treasure was buried with a cross.

 13

Learning with Cubetto - 6 week robotics program PRIMO

Lesson 4: Cubetto's Quest (2 of 2)

Creative Play

Make or find some treasure for Cubetto to discover.

Guided Activity

1. On squared paper, make your own map like Cubetto's.
2. Draw different pictures in each square. You might want to choose a theme such as school, sport, music or a game.
3. Decide where on the map your treasure is buried. Mark on the back of your sheet where it is (to keep it secret).
4. Write down where to start on your map.
5. Write an algorithm for where your treasure is hidden: this is your treasure hunt clue.
6. Find a partner and ask them to predict where your treasure is buried by looking at your map and working out the algorithm.

Independent Activity

1. Look at the first treasure hunt clue. Where do you predict the algorithm will take you?
2. Put a cross on the map where you think the algorithm will take you.
3. Program Cubetto using the algorithm clue and press the action button.
4. Did you predict the right place? Why? Why not?
5. If you weren't right, try to work out which part you got wrong.
6. Repeat for other algorithms.

Challenge

Can you say the coordinates of where the treasure is hidden?

Plenary and Assessment

1. Ask for a volunteer to share their treasure map with the class.
2. The class predicts where the treasure is hidden and programs Cubetto to test it out.
3. Ask: Could we make the algorithm simpler or use fewer blocks? Reinforce the importance of making things simpler.
4. Ask: Which clues did you find harder to work out? Why do you think this was?
5. Ask: What does predict mean? What is an algorithm? What does program mean?
6. Ask the children to think about today and what they did. Ask: What skills do people who work in computing need to have? Collect and display. Elicit: try again and again/perseverance; make sure it's correct/be exact/precise; make it better each time/more efficient.

 14

Teacher Beliefs Questionnaire

The teacher questionnaire administered in this study combined two validated attitude measures, the Dimensions of Attitudes toward Science (DAS) scale (van Aalderen-Smeets & Walma van der Molen, 2013) and the Preschool teacher attitudes and beliefs toward science (P-TABS) questionnaire (Maier et al., 2013). These validated measures have previously been used to assess several aspects of teachers' beliefs toward science teaching. Both investigated teacher's self-efficacy, cognitive aspects (i.e., perceived relevance of teaching the subject), affective aspects (enjoyment, anxiety, or fear regarding science teaching) and contextual aspects (perceived time and resources needed for science teaching). An advantage of combining these measures is that the P-TABS also included items which investigated how teachers intended to teach the subject and their teaching practices (behavioural aspects). This later served as a manipulation check to confirm that teachings of programming and robotics content increased during the intervention (for Intervention+ and Intervention groups).

Duplicate questions and questions too specifically related to physical sciences were removed. For example, *"I demonstrate experimental procedures (e.g., comparing objects to see if they will sink or float) in my classroom"* was removed. Both positively and negatively worded items were included to reduce the likelihood that teachers' answers would be skewed toward positive response options. All items were rated on a five-point Likert scale ranging from *"strongly disagree" to "strongly agree"*. A total of 52 items (see Appendix I) were used in the final questionnaire, compartmentalised into 8 subscales (defined below), 7 of which reflected the defined subscales used by van Aalderen-Smeets and Walma van der Molen, (2013).

Teachers completed this questionnaire at three timepoints during the study. First, it was completed at the start of the intervention (prior to data collection, the robotics curriculum, and any teacher education workshop). Next, all teachers were invited to submit a response after Intervention+ teachers had completed an educational robotics workshop (approximately 3 weeks after the first assessment). The final measure of teachers' attitudes was taken approximately 7 weeks later once teachers in the Intervention+ and Intervention conditions had completed the robotics curriculum in their classrooms.

The first subscale focused on the *relevance of teaching programming* (both in present and future contexts). These 16 items ($\alpha = .88$) measured the extent to which teachers found it important and relevant to teach programming to children in the foundation phase. This component was measured by items such as: *"I think that programming should be included in* *primary education as early as possible."* Additionally, items also assessed the extent to which teachers found it important to teach programming to primary school children for their pupils' future development. These items asked teachers to consider statements like *"I believe that programming education in the primary school is essential for students to be able to make good educational and career choices."* Scores on this subscale could range from 16 to 80, with higher scores indicating more positive beliefs about the importance of programming in early years education.

The second subscale, *difficulty of teaching programming*, included 3 items (α = .60) and investigated whether teachers thought that programming in general was more difficult to teach than other topics. This component represented teachers' general beliefs about the difficulty of teaching programming rather than their perceptions about their own ability to teach programming. van Aalderen-Smeets and Walma van der Molen (2013) noted the importance of using items in this subscale that were unambiguous with respect to this difference. As a result, items were phrased in the following manner: *"Most teachers find programming difficult to teach"* instead of *"Programming is difficult to teach."* Scores on this subscale could range from 3 to 15, with higher scores indicating higher perceptions of difficulty.

The third subscale assessed *gender-stereotypical beliefs* and included 4 items (α = .71). These gender related beliefs were assessed in two ways. Firstly, items asked teachers to consider potential differences between male and female teachers with regards to their ability to teach programming, interest in the topic and enjoyment. Secondly, items assessed perceived differences between boys and girls (pupils) in programming (in ability, interest, and enjoyment). Scores on this subscale could range from 4 to 20.

The fourth and fifth subscales investigated *teachers' enjoyment* and their *anxiety* (when teaching programming). Items in both sections measured the experiences related to teaching programming, both positive and negative. Scores on the *enjoyment* subscale contained 5 items (α = .91) and thus scores could range from 5 to 25, with higher scores indicating higher enjoyment. The *anxiety* subscale had 4 items (α = .86), with scores ranging from 4 to 20, with higher scores indicating higher anxiety.

The sixth subscale specifically measured teacher *self-efficacy*. These items questioned teachers about their perceived ability to teach programming in primary school themselves and their ability to handle problems that may arise when teaching the subject. Eight items were included in this subscale (α = .85); thus, scores could range from 9 to 45, with higher scores indicating higher self-efficacy.

The seventh subscale focused on contextual factors and assessed perceptions teachers may have about *external factors* that could hinder or advance their teaching of programming. For instance, teachers considered statements like *"For me, the availability of a ready-to-use existing package of materials (e.g., robotics kits) is an essential prerequisite for being able to teach programming in class."* Items did not solely focus on physical resources, but also asked teachers to consider the importance of colleague support (e.g., *"For me, the support of my colleagues is decisive for whether or not I will teach programming in class"*). This subscale included 5 items (α = -0.20) and scores could range from 5 to 25. Higher scores indicated more perceived barriers to programming education.

An eighth subscale emerged as a result of integrating the P-TABS questionnaire. These final items investigated teachers' *current teaching practices* (when completing the questionnaire) with items such as "*I make an effort to include some programming activities throughout the week."* Seven items were included in this scale (α = 0.74) and scores could range from 7 to 35.

Feedback Forms

Teachers delivering the robotics curriculum were asked to complete feedback forms after delivering each lesson so that we could monitor how they were implementing the robotics curriculum. These weekly feedback forms were described in detail in the previous Chapter (see Chapter 5). To summarise, teachers were asked to complete one form each week, recording how much time they spent teaching with robotics that week and whether they used any of the additional lesson plans provided (which they did not, see previous chapter for details). Teachers were also given open text boxes to reflect on how each session went, noting both positive and negative aspects. Control group teachers were given a weekly feedback form to complete that prompted them to share details of any CT activities they delivered to their class as part of their normal curriculum.

Pupil Assessments

The previous chapter described a collection of pupil assessments that measured pupil skills related to computational thinking (i.e., debugging, prediction and picture-sequencing skills), their programming skills and their personal beliefs. In that chapter, I reported that the 6-week robotics curriculum improved children's debugging and prediction skills when the curriculum was delivered by a classroom teacher who attended a teacher education workshop prior to curriculum delivery (i.e., Intervention+). For a full description of the debugging and prediction tasks, see Chapter 5. As these effects were only found when teachers received additional education prior to the intervention, this suggested that the teachers themselves may have played an important role in the success of the robot intervention. Thus, in the analyses for this chapter, I explore whether teachers' beliefs impacted changes in pupil's performance on these outcome measures.

Data Preparation

To prepare data for analysis, responses on the teacher beliefs questionnaire were translated into numerical responses on a 5-point scale. Answers of 'strongly agree' were given a score of 5, with answers of 'strongly disagree' given a score of 1. Negatively phrased items were reverse scored (i.e., 'strongly agree' was then given a score of 1). Scores for each question within the 8 subscales were then combined to produce a total score for each section of the questionnaire. Subscales were then analysed separately to investigate the effects of the intervention on different teacher beliefs.

This results section outlines the analysis and findings for 5 of the 8 subscales. These include *teacher enjoyment*, *relevance, anxiety*, and *self-efficacy* beliefs. Questions pertaining to *teaching practices* were also included as this served as a manipulation check to explore whether implementing the 6-week robotics curriculum increased teachers and pupils' engagement with programming and computational thinking tasks. The remaining subscales (*gender beliefs*, *perceptions of external barriers* and *difficulties teaching programming*) are not explored in the main text for several reasons. Firstly, the *difficulty of teaching programming* subscale required teachers to think about how "most teachers" felt about teaching programming rather than their own personal beliefs and we had no expectations that the intervention would alter their ratings of others' perceptions. Secondly, this study did not manipulate *external factors* or barriers teachers may face when trying to teach programming and robotics education, nor was it an intervention targeting teachers' *gender stereotype* beliefs. Although these data are not explored in this chapter, analysis of these subscales can be found in the Appendix J.

Multi-Level Modelling

Multilevel modelling is a statistical method commonly used when data is nested within a hierarchal structure (Hox, 1998). In this study, teachers were nested within schools, which were then nested within intervention conditions. Additionally, teacher beliefs data was collected at three timepoints (pre-workshop, post-workshop, and post-intervention), thus measurement occasion was nested within the individual. Multilevel models account for the fact that observations made on the same individual at different time points are likely to be highly correlated (Koepsell et al., 1992). As this chapter also explores the effect of teacher beliefs on pupil outcomes, it is worth noting that in this hierarchical structure, pupils were nested within teachers and pupil data was collected at two time points (pre and post intervention).

Fitting Models

All models were built and analysed using R Statistical Software (R Core Team, 2023), specifically the *lme4* package (Bates et al., 2015). When employing this package for fitting mixed-effects models, the methodology predominantly relies on Maximum Likelihood Estimation (MLE), defaulting to the use of Full Information Maximum Likelihood (FIML) (Grund et al., 2018). MLE aims to find parameter values that maximize the likelihood function, offering unbiased estimates. Integrated into *lme4*, FIML addresses missing data by incorporating all available information, including cases with missing values, during the parameter estimation process. This is particularly advantageous in scenarios where data is missing. In this study, data was missing when teachers did not complete belief questionnaires at all time points.

To achieve the aims of this study, all base models included interaction effects of Condition (Intervention+, Intervention, Control) and assessment Time-point (i.e., pre-workshop, post-workshop, and post-intervention). A random effects structure was fit with random intercepts only. Initially, with the aim of accounting for maximum variability, both School and classroom Teacher were included as random intercepts. However, this raised singularity errors in R, identical to those raised in the previous pupil intervention chapter. It is thought that these errors occur when the random effect structure is too complex. Thus, models with independent School or Teacher intercepts were compared. The Akaike information criterion (AIC; Akaike, 1974) was used to determine which variable provided the best fit for the data. Selecting Teacher as the random intercept provided the best model fit for the data (as indicated by the lower AIC value).

Results

Questionnaire Responses

The number of questionnaire responses collected at each time point is displayed in Table 6.2. Attempts to collect teacher beliefs data from the Control group at time point two were unsuccessful, however time points one and two had better response rates. Additionally, two Reception class teachers from the Control group did not complete the beliefs questionnaire at any of the three timepoints. To investigate changes in teacher beliefs across the intervention, all time points were included in the analysis models. When investigating how teacher beliefs were related to changes in pupil's learning outcomes, teacher beliefs at time point three (post-intervention) were used. This is explained further later in this section.

Table 6.2

Responses to the Teacher Beliefs Questionnaire.

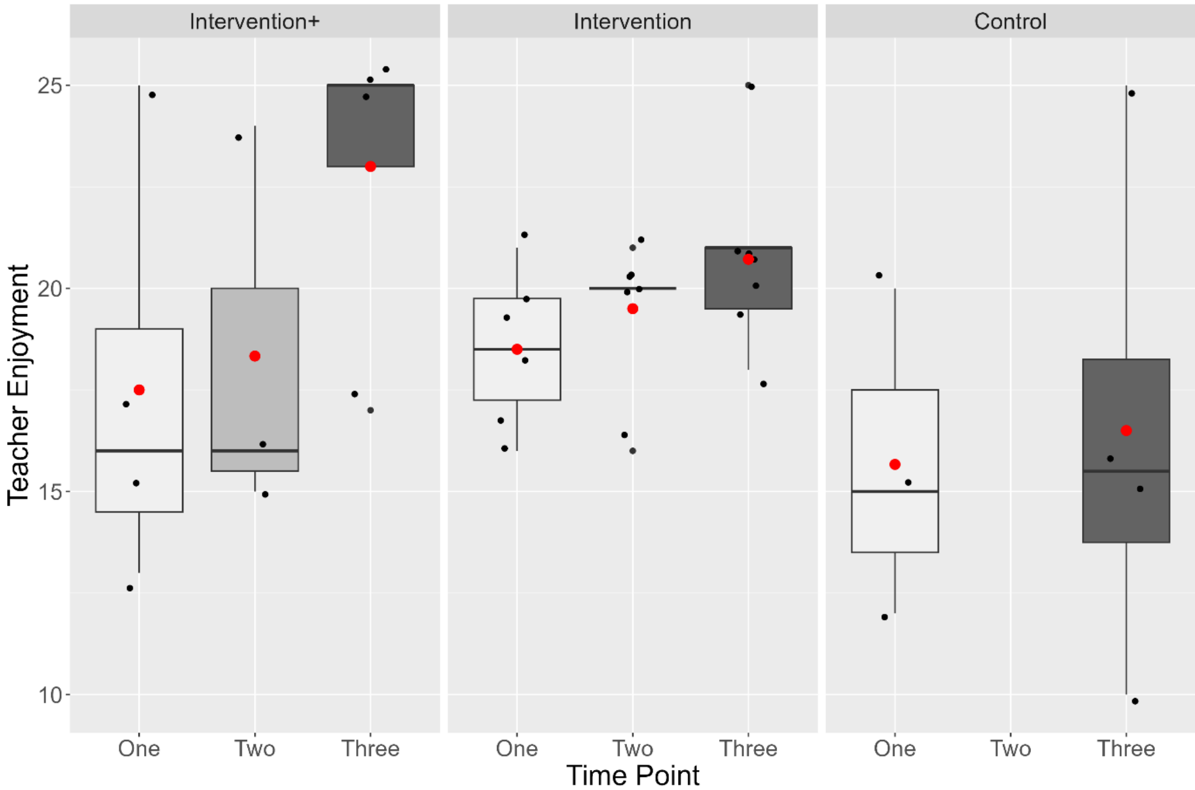
Condition	**Time-point 1**		**Time-point 2**		**Time-point 3**	
	N	Missing	*N*	Missing	*N*	Missing
Intervention+	4	0	3	1	4	0
Intervention	6	1	6	1	7	0
Control	3	3	0	6	4	2
Total	13	4	9	8	15	2

Enjoyment Scores

Figure 6.4 illustrates teacher enjoyment scores as measured by the beliefs questionnaire. Within R, plots were created with the *Tidyverse* packages (Wickham et al., 2019).

Figure 6.4

Teacher Enjoyment Scores, Grouped by Condition and Time Point.



Note: Group means at each time point are shown in red.

A Linear Mixed-effect Model (LMM) from the *lme4* package in R (Bates et al., 2015) was used to analyse group differences across time points. The current model included teacher enjoyment scores as an ordinal outcome variable. The interaction between Condition (Intervention+, Intervention and Control) and Time (pre-workshop, post-workshop, and post-intervention) was included as a fixed effect and observations were grouped by Teacher. Confidence intervals were computed with the *confint()* function. P values were obtained using the *lmeTest* package. The model that was estimated used the following structure:

Teacher Enjoyment Score ~ Condition*Time + (1|TeacherID).

Table 6.3

Teacher Enjoyment LMM Model Results: Fixed Effects.

	Model summary					
	β	*SE*	*t*	CI 95%	*p*	
Intercept	15.54	2.02	7.70	[11.83, 19.26]	< 0.001	
Intervention+ Vs Control	1.52	2.73	0.56	[-3.51, 6.57]	0.58	
Intervention Vs Control	3.52	2.48	1.42	[-1.04, 8.12]	0.17	
Time	0.48	0.79	0.60	[-1.03, 1.96]	0.55	
Intervention+ vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	2.27	1.05	2.15	[0.27, 4.29]	0.04	*
Intervention vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	0.37	0.97	0.38	[-1.45, 2.25]	0.71	

*Note: * $p < .05$.*

The fixed effects from the model results are described in Table 6.3. The model revealed a significant interaction effect of Condition x Time on teachers' enjoyment scores. On average, improvements in teacher enjoyment scores in the Intervention+ condition were significantly greater than changes in enjoyment scores in the Control group ($p = 0.04$). On the other hand, changes in teacher enjoyment scores in the Intervention condition were not significantly different from those in the Control group. When comparing both intervention groups, improvements in enjoyment scores were significantly larger for those in the Intervention+ condition than the Intervention condition, ($B = 1.90$, $SE = 0.89$, $t = 2.12$, CI 95% [0.17, 3.59], $p = 0.04$).

The Condition x Time interaction was followed up using the *pairs()* function in the *emmeans* package (Lenth, 2021). Results are displayed in Table 6.4. Results show that enjoyment scores significantly improved overall (i.e., between time points one and three) for teachers in the Intervention+ condition ($p = 0.003$). Additionally, enjoyment scores significantly increased between time points two and three ($p = 0.03$), but not between time points one and two ($p = 0.78$).

For teachers in the Intervention condition, there were no significant improvements in enjoyment scores between pre and post intervention ($p = 0.32$). Similarly, there were no significant improvements between time point one and two ($p = 0.67$) or two and three ($p = 0.82$).

In the Control condition, post hoc tests revealed no significant difference in teacher enjoyment scores pre and post intervention ($p = 0.83$). Comparisons could not be made for time point two due to missing data from teachers in this condition.

Table 6.4

Teacher Enjoyment Post Hoc Test Results.

Condition	Time	*β*	*SE*	*t*	*p*	
Intervention+	One - Two	-1.04	1.58	-0.66	0.79	
	One - Three	-5.50	1.42	-3.88	0.003	**
	Two - Three	-4.46	1.58	-2.82	0.02	*
Intervention	One – Two	-1.00	1.15	-0.86	0.67	
	One – Three	-1.70	1.15	-1.48	0.32	
	Two - Three	-0.50	1.15	-0.61	0.82	
Control	One – Two					
	One – Three	-0.95	1.62	-0.59	0.83	
	Two - Three					

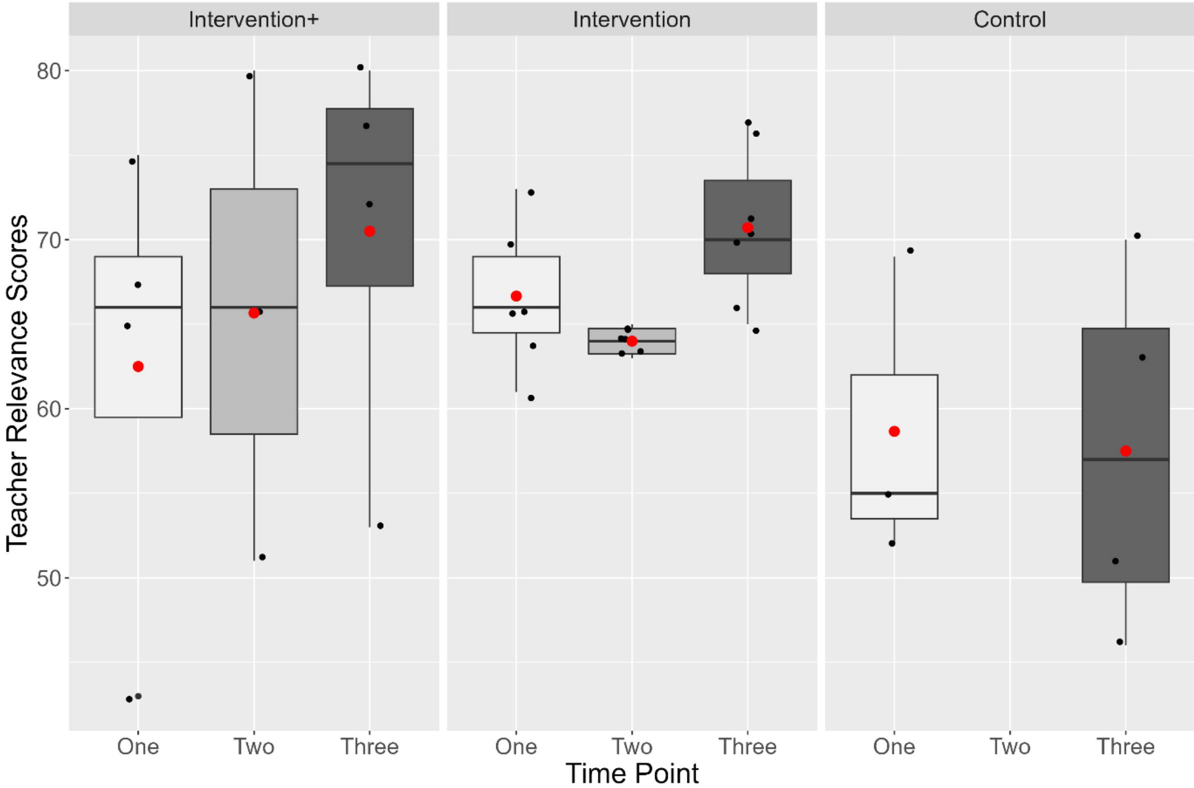
Note: * $p < .05$, ** $p < 0.01$, *** $p < 0.001$.

Relevance Scores

Figure 6.5 illustrates teacher relevance scores across as measured by the beliefs questionnaire.

Figure 6.5

Teacher Relevance Scores, Grouped by Condition and Time Point.



Note: Group means at each timepoint are shown in red.

A LMM was used to analyse group differences across time points. The model included teacher relevance scores as an ordinal outcome variable. The interaction between Condition and Time was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following model structure:

Teacher Relevance Score ~ Condition*Time + (1|TeacherID).

Table 6.5

Teacher Relevance LMM Model Results: Fixed Effects.

	Model summary					
	β	*SE*	*t*	CI 95%	*p*	
Intercept	60.22	4.48	13.43	[51.95, 68.54]	< 0.001	
Intervention+ Vs Control	2.50	6.17	0.40	[-8.98, 13.91]	0.69	
Intervention Vs Control	4.95	5.55	0.89	[-5.36, 15.21]	0.39	
Time	-1.36	1.36	-1.00	[-3.91, 1.25]	0.33	
Intervention+ vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	5.36	1.80	2.97	[1.89, 8.75]	0.01	**
Intervention vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	3.45	1.66	2.07	[0.26, 6.58]	0.06	**.**

Note: **.** $p < 0.1$, ** $p < 0.01$.

The fixed effects from the model results are described in Table 6.5. The model revealed a significant interaction effect of Condition x Time on teachers' relevance scores. On average, improvements in teacher relevance scores in the Intervention+ condition were significantly larger than in the Control group ($p = 0.007$). Improvements in teacher relevance scores in the Intervention condition were not significantly different from those in the Control condition ($p = 0.06$). When comparing both intervention groups, improvements in relevance scores in the Intervention+ condition did not differ significantly from those in the Intervention condition, ($B = 1.91$, $SE = 1.53$, $t = 1.25$, CI 95% [-1.00, 4.81], $p = 0.23$).

The Condition x Time interaction was followed up using the *pairs()* function in the *emmeans* package (Lenth, 2021). Results are displayed in Table 6.6. Results show that relevance scores significantly improved overall (i.e., between time points one and three) for teachers in the Intervention+ condition ($p = 0.002$). Additionally, relevance scores showed marginal increases between time points one and two ($p = 0.08$), but no significant differences were found between time points two and three ($p = 0.33$).

For teachers in the Intervention condition, there were significant improvements in relevance scores between time points one and three ($p = 0.04$). Additionally, there was a significant improvement in relevance scores between time points two and three ($p = 0.001$). However, these results should be interpreted with caution as the model revealed no significant effect of Time and Condition overall within the Intervention condition. In the Control condition, post hoc tests revealed no significant difference in teacher relevance scores pre and post intervention ($p = 0.42$). Comparisons could not be made for time point two due to missing data from teachers in this condition.

Table 6.6

Teacher Relevance Post Hoc Test Results.

Condition	Time	*β*	*SE*	*t*	*p*	
Intervention+	One - Two	-4.90	2.12	-2.31	0.08	.
	One - Three	-8.00	1.90	-4.22	0.002	**
	Two - Three	-3.10	2.12	-1.46	0.34	
Intervention	One – Two	2.67	1.55	1.72	0.23	
	One – Three	-4.15	1.55	-2.69	0.04	*
	Two - Three	-6.82	1.55	-4.41	0.001	**
Control	One – Two					
	One – Three	2.81	2.18	1.29	0.42	
	Two - Three					

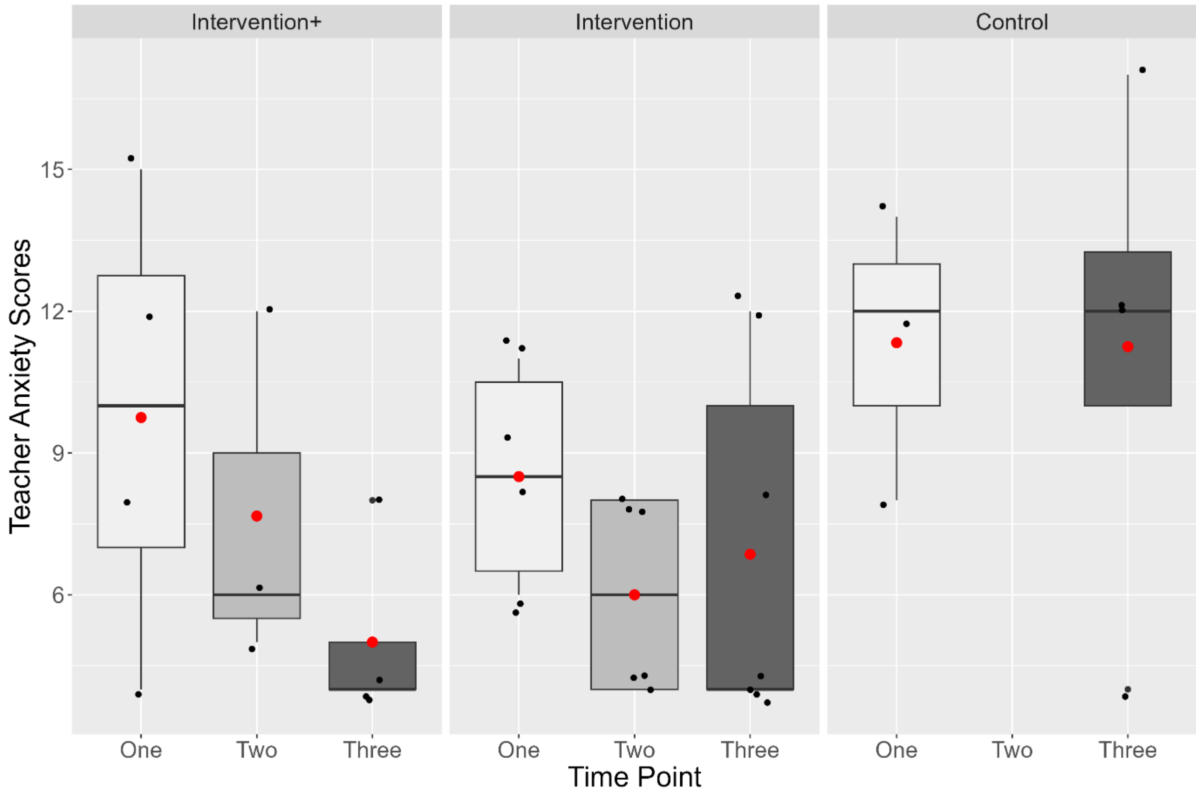
Note: . $p < 0.1$, * $p < .05$, ** $p < 0.01$.

Anxiety Scores

Figure 6.6 illustrates teacher anxiety scores as measured by the beliefs questionnaire.

Figure 6.6

Teacher Anxiety Scores, Grouped by Condition and Time Point.



Note: Group means at each timepoint are shown in red.

An LMM was used to analyse group differences across time points. The model included teacher anxiety scores as an ordinal outcome variable. The interaction between Condition and Time was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following model structure:

Teacher Anxiety Score ~ Condition*Time + (1|TeacherID).

Table 6.7

Teacher Anxiety LMM Model Results: Fixed Effects.

	Model summary				
	β	*SE*	*t*	CI 95%	*p*
Intercept	11.84	1.99	5.94	[7.99, 15.56]	<0.001
Intervention+ Vs Control	-1.86	2.59	-0.72	[-6.64, 2.95]	0.48
Intervention Vs Control	-4.03	2.40	-1.68	[-8.53, 0.57]	0.10
Time	-0.29	1.16	-0.25	[-2.43, 2.13]	0.80
Intervention+ vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	-2.08	1.57	-1.32	[-5.30, 0.84]	0.20
Intervention vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	-0.43	1.43	-0.30	[-3.43, 2.20]	0.76

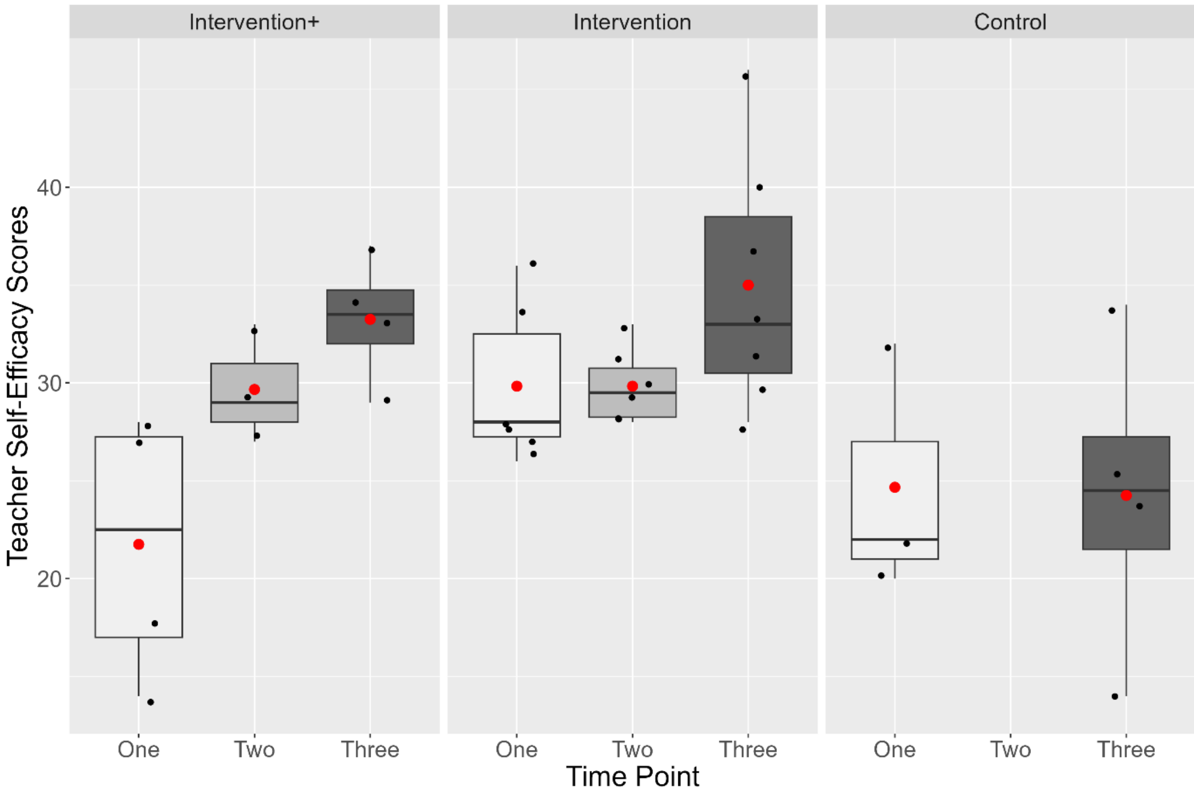
The fixed effects from the model results are described in Table 6.7. The model revealed no significant interaction effect of Condition x Time on teachers' anxiety scores. Results illustrate that there were no significant differences in reductions in anxiety scores over time, between the Intervention+ and Control groups, and the Intervention and Control groups. Similarly, there was no significant difference within the Intervention+ and Intervention conditions ($B = -1.65$, $SE = 1.35$, $t = -1.22$, CI 95% [-4.23, 1.01], $p = 0.24$).

Self-Efficacy Scores

Figure 6.7 illustrates teacher self-efficacy scores as measured by the beliefs questionnaire.

Figure 6.7

Teacher Self-Efficacy Scores, Grouped by Condition and Time Point.



Note: Group means at each timepoint are shown in red.

An LMM was used to analyse group differences across time points. The model included teacher self-efficacy scores as an ordinal outcome variable. The interaction between Condition and Time was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following model structure:

Teacher Self-efficacy Score ~ Condition*Time + (1|TeacherID).

Table 6.8

Teacher Self-Efficacy LMM Model Results: Fixed Effects.

	Model summary					
	β	*SE*	*t*	CI 95%	*p*	
Intercept	22.80	3.21	7.11	[16.75, 28.92]	< 0.001	
Intervention+ Vs Control	-0.23	4.26	-0.05	[-8.12, 7.80]	0.96	
Intervention Vs Control	7.37	3.91	1.88	[-0.41, 14.94]	0.07	.
Time	0.72	1.55	0.47	[-2.81, 3.55]	0.64	
Intervention+ vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	5.03	2.07	2.43	[1.19, 9.59]	0.02	*
Intervention vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	1.39	1.90	0.73	[-2.09, 5.98]	0.48	

Note: **.** $p < 0.1$, * $p < .05$.

The fixed effects from the model results are described in Table 6.8. The model revealed a significant interaction effect of Condition x Time on teachers' self-efficacy scores. On average, improvements in teacher self-efficacy scores in the Intervention+ condition were significantly larger than in the Control group ($p = 0.02$). Improvements in teacher self-efficacy scores in the Intervention condition were not significantly different than those in the Control condition ($p = 0.48$). When comparing both intervention groups, improvements in self-efficacy scores in the Intervention+ condition did not differ significantly from those in the Intervention condition, but this difference was marginal, ($B = 3.64$, $SE = 1.77$, $t = 2.06$, CI 95% [-0.12, 6.98], $p = 0.06$).

The Condition x Time interaction was followed up using the *pairs()* function. Results are displayed in Table 6.9. Results show that self-efficacy scores significantly improved overall (i.e., between time points one and three) for teachers in the Intervention+ condition ($p = 0.001$). Additionally, self-efficacy scores showed significant increases between time points one and two ($p = 0.02$), but no significant differences were found between time points two and three ($p = 0.68$).

For teachers in the Intervention condition, there were no significant improvements in self-efficacy scores between time points one and three ($p = 0.17$). Additionally, significant improvements were not found between time points one and two ($p = 1.00$) or two and three ($p = 0.17$).

In the Control condition, post hoc tests revealed no significant difference in teacher self-efficacy scores pre and post intervention (p = 0.87). Comparisons could not be made for time point two due to missing data from teachers in this condition.

Table 6.9

Teacher Self-Efficacy Post Hoc Results.

Condition	Time	β	*SE*	*t*	*p*	
Intervention+	One - Two	-9.00	3.00	-2.99	0.02	*
	One - Three	-11.50	2.70	-4.27	0.001	**
	Two - Three	-2.50	3.00	-0.83	0.69	
Intervention	One – Two	0.00	2.20	0.00	0.23	
	One – Three	-4.13	2.18	-1.89	0.17	
	Two - Three	-4.13	2.18	-1.89	0.17	
Control	One – Two					
	One – Three	-1.52	3.06	-0.50	0.87	
	Two - Three					

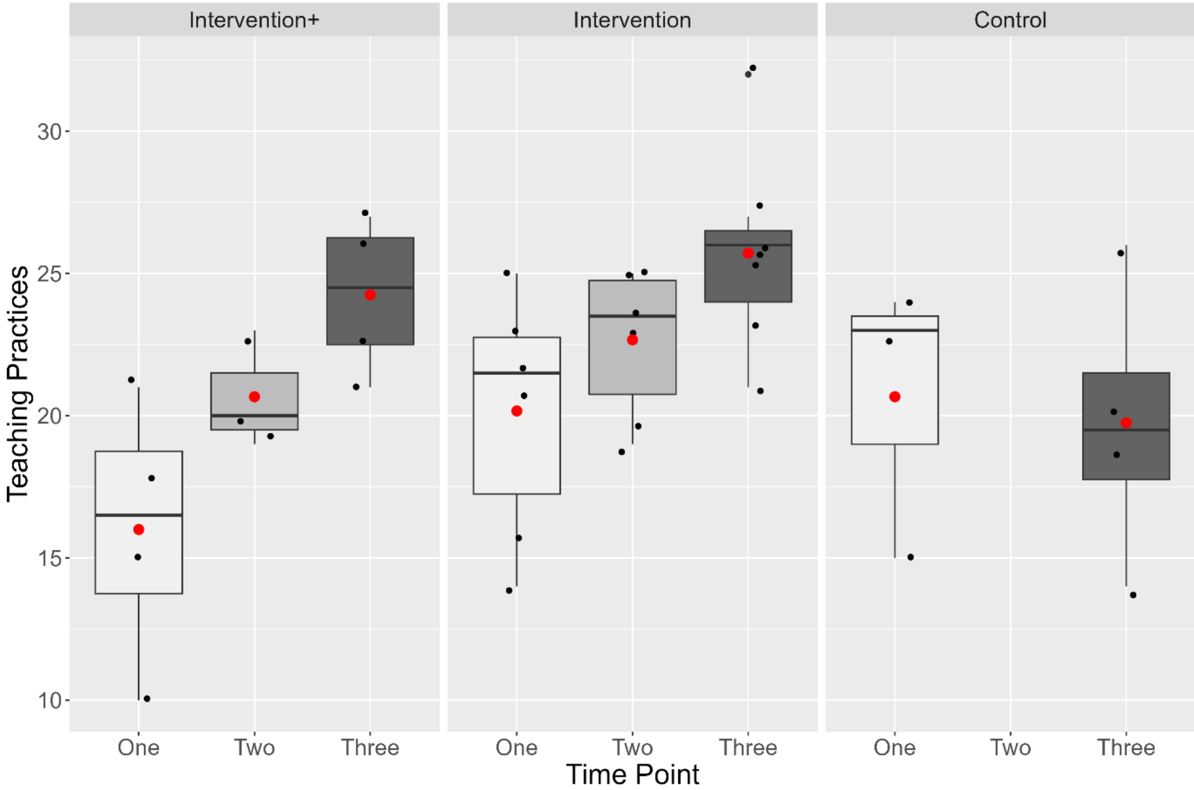
Note: * $p < .05$, ** $p < 0.01$.

Teaching Practices

Figure 6.8 illustrates teaching practice scores as measured by the beliefs questionnaire.

Figure 6.8

Teaching Practices Scores, Grouped by Condition and Time Point.



Note: Group means at each timepoint are shown in red.

An LMM was used to analyse group differences across time points. The model included teaching practice scores as an ordinal outcome variable. The interaction between Condition and Time was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following model structure:

Teacher Practices Score ~ Condition*Time + (1|TeacherID).

Table 6.10

Teaching Practices LMM model results: Fixed Effects.

	Model summary					
	β	*SE*	*t*	CI 95%	*p*	
Intercept	20.51	2.05	10.03	[16.75, 24.27]	< 0.001	
Intervention+ Vs Control	-4.51	2.73	-1.65	[-9.54, 0.52]	0.11	
Intervention Vs Control	-0.28	2.50	-0.11	[-4.86, 4.32]	0.91	
Time	-0.38	0.92	-0.41	[-2.14, 1.36]	0.68	
Intervention+ vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	4.51	1.24	3.65	[2.17, 6.86]	0.002	**
Intervention vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	3.09	1.14	2.72	[0.96, 5.26]	0.01	*

Note: * $p < .05$, ** $p < 0.01$.

The fixed effects from the model results are described in Table 6.10. The model revealed a significant interaction effect of Condition x Time on teachers' teaching practices. On average, improvements in teaching practice scores in the Intervention+ condition were significantly larger than in the Control group (p = 0.002). Improvements in teaching practice scores in the Intervention condition also significantly improved in comparison to those in the Control condition (p = 0.01). When comparing both intervention groups, improvements in teaching practice scores in the Intervention+ condition did not differ significantly from those in the Intervention condition, (B = 1.41, SE = 1.05, t = 1.34, CI 95% [-0.60, 3.41], p = 0.19).

The Condition x Time interaction was followed up using the *pairs()* function. Results are displayed in Table 6.11. Results show that teaching practice scores significantly improved overall (i.e., between time points one and three) for teachers in the Intervention+ condition (p < 0.001) and the Intervention condition (p = 0.003). Thus, teachers in these conditions increased their teachings of programming and robotics in line with the curriculum set in this study. Furthermore, in the Control condition, post hoc tests revealed no significant difference in teaching practice scores pre and post intervention (p = 0.92), thus suggesting their teachings of programming and robotics did not change over the course of the intervention.

Table 6.11

Teaching Practices Post Hoc Test Results.

Condition	Time	*β*	*SE*	*t*	*p*	
Intervention+	One - Two	-4.15	1.93	-2.15	0.11	
	One - Three	-8.25	1.73	-4.77	< 0.001	
	Two - Three	-4.10	1.93	-2.13	0.11	
Intervention	One – Two	-2.50	1.41	-1.77	0.21	
	One – Three	-5.42	1.40	-3.88	0.003	**
	Two - Three	-2.92	1.40	-2.09	0.12	
Control	One – Two					
	One – Three	0.77	1.96	0.39	0.92	
	Two - Three					

Note: ** $p < 0.01$.

Investigating the Effects of Teacher Beliefs on Pupil Outcomes

Linear mixed-effect models were used to investigate whether there was an effect of teacher beliefs on pupil's improved performance on outcome measures. The following analyses included pupil performance on debugging and prediction tasks as a continuous difference score (post-test score minus pre-test score). These analyses further explored teacher beliefs effects on these two outcome variables due to the significant differences in performance between conditions described in the previous chapter (see Chapter 5).

As time point two data (post-workshop) was missing for eight teachers (47% of the overall sample), teacher beliefs data from time point three (post-intervention) was used in this analysis. Teacher beliefs at time point three were used to capture the variability caused by the intervention conditions. Observations were grouped by classroom teacher. Due to the small number of classroom teachers that participated in this study, the potential effects of teacher beliefs on pupil outcomes were investigated across the three intervention conditions, rather than as separate groups. The following analyses included teacher beliefs data from 15 teachers and up to 413 pupils (where available). Class sizes for each teacher varied between 12 and 52 pupils (in cases where one teacher taught two classes, average classroom size = 28 pupils). Data from 40 pupils could not be analysed here as two classroom teachers did not complete the beliefs questionnaire. Models used the following structure:

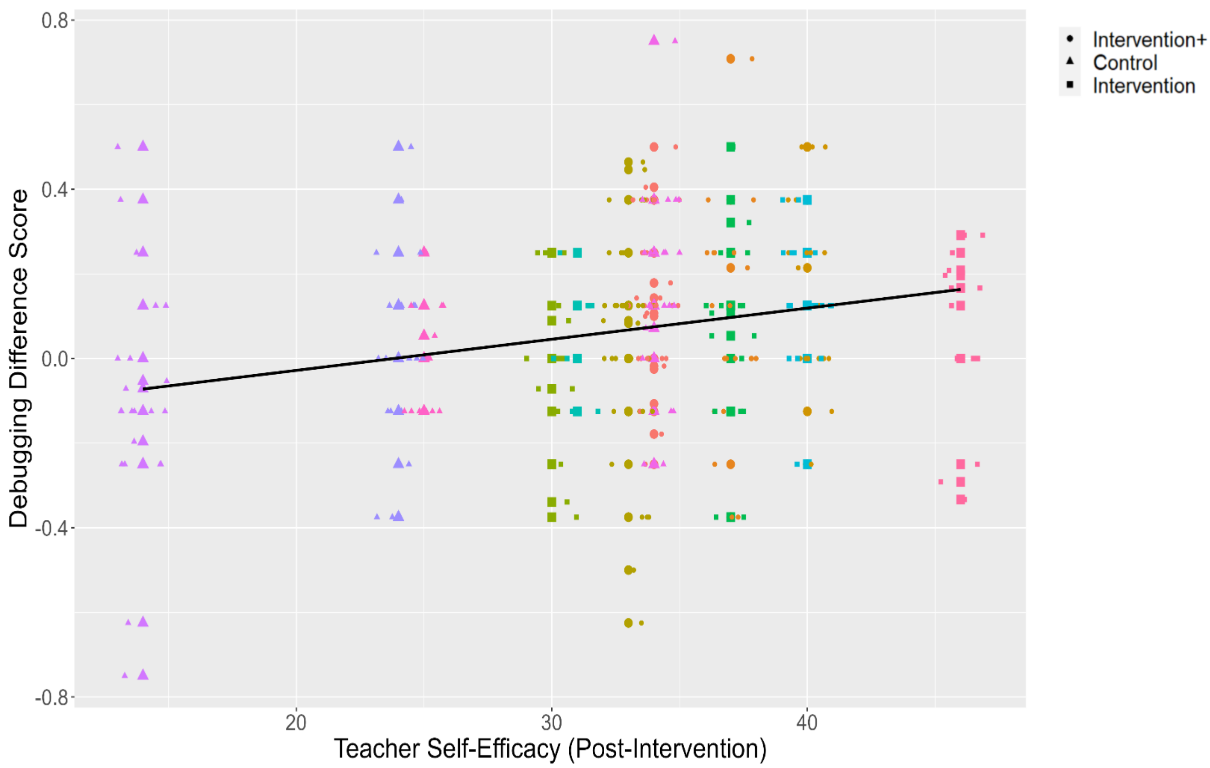
Pupil Difference Score ~ Teacher Beliefs Post-Intervention Score + (1|TeacherID).

Pupils' Debugging Performance

This analysis included beliefs data from 15 teachers and debugging performance data from 274 pupils. 120 children did not provide debugging data pre- and post-intervention; thus, a difference score could not be calculated for these participants, so they were not included in this analysis. Models exploring the effects of teacher beliefs on changes in pupil's debugging performance found a significant main effect of teacher self-efficacy (B = 0.007, SE = 0.001, t = 3.72, CI 95% [0.003, 0.01], p = 0.002; see Figure 6.9) and teacher anxiety (B = -0.01, SE = 0.003, t = -4.45, CI 95% [-0.02, -0.01], p = 0.0003; see Figure 6.10). On the other hand, there was no significant main effect of teacher relevance beliefs on pupil's debugging improvement (B = 0.001, SE = 0.002, t = 0.77, CI 95% [-0.002, 0.03], p = 0.45) or teacher enjoyment beliefs (B = 0.007, SE = 0.004, t = 1.67, CI 95 [-0.001, 0.02], p = 0.12). Thus, these results suggest that higher teacher self-efficacy and lower anxiety was associated with improved debugging performance.

Figure 6.9

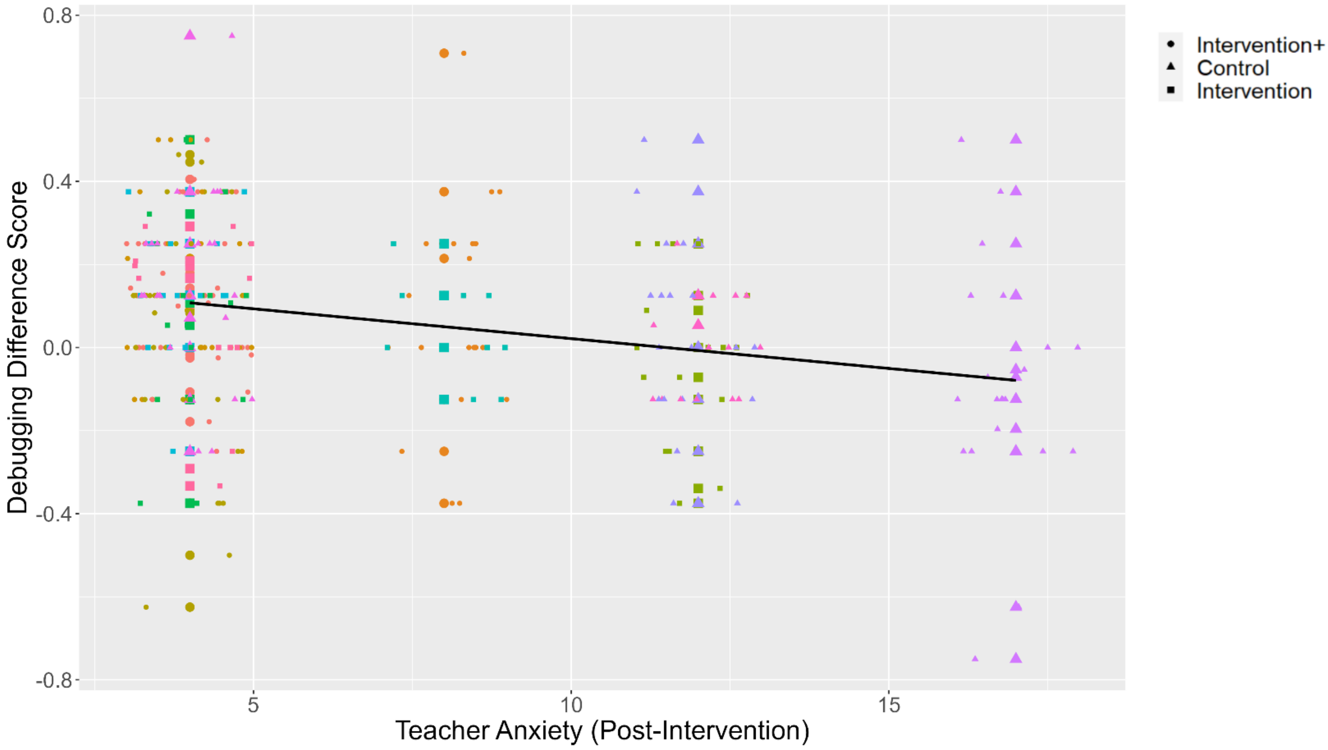
The Relation Between Teacher Enjoyment Scores and Changes in Pupils' Debugging Scores.



Note: Colours indicate individual teachers.

Figure 6.10

The Relation Between Teacher Anxiety Scores and Changes in Pupils' Debugging Scores.



Note: Colours indicate individual teachers.

Pupils' Prediction Performance (Correct Answers)

This analysis included data from 15 teachers and debugging performance data from 292 pupils. Models exploring the effects of teacher beliefs on changes in pupil's prediction performance (specifically how many trials they answered correctly) found that the effect of teacher relevance scores was marginal ($B = 0.002$, $SE = 0.001$, $t = 2.05$, CI 95% [0.0001, 0.005], $p = 0.07$). Furthermore, there was no significant effect of teacher enjoyment scores ($B = 0.005$, $SE = 0.003$, $t = 1.74$, CI 95% [-0.001, 0.01], $p = 0.11$), self-efficacy ($B = 0.003$, $SE = 0.002$, $t = 1.49$, CI 95% [-0.001, 0.007], $p = 0.16$) or anxiety scores ($B = -0.004$, $SE = 0.004$, $t = -1.05$, CI 95% [-0.01, 0.003], $p = 0.32$).

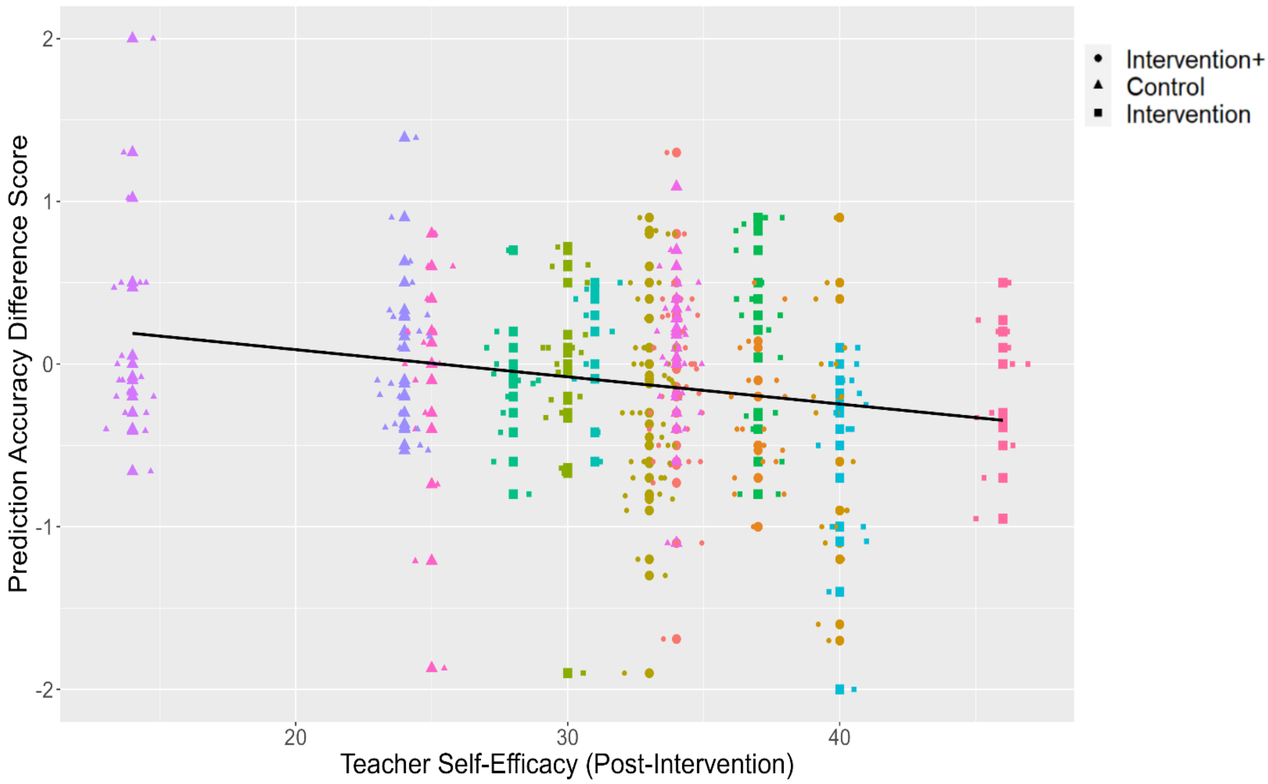
Pupils' Prediction Performance (Accuracy)

This analysis included data from 15 teachers and prediction performance data from 292 pupils. Please note that when scoring the accuracy of children's answers on the prediction task, lower scores at post-intervention indicated improved accuracy and thus improved performance (see Chapter 5 for more details). Models exploring the effects of teacher beliefs on changes in pupil's prediction accuracy found a significant main effect of teacher self-efficacy scores only

(B = -0.02, SE = 0.007, t = -2.26, CI 95% [-0.03, -0.002], p = 0.04; see Figure 6.11). Model results showed no significant effect of enjoyment scores (B = -0.02, SE = 0.01, t = -1.75, CI 95% [-0.05, 0.003], p = 0.12), relevance scores (B = -0.009, SE = 0.006, t = -1.49, CI 95% [-0.02, 0.003], p = 0.16) or anxiety scores (B = 0.02, SE = 0.01, t = 1.55, CI 95% [-0.01, 0.05], p = 0.15) on pupil's prediction accuracy scores.

Figure 6.11

The Relation Between Teacher Self-Efficacy Score and Changes in Pupils' Debugging Scores.



Note: Colours indicate individual teachers.

Discussion

This chapter continued to explore data collected during a programming and robotics intervention implemented within primary school classrooms in South Wales. Specifically, this chapter presented the teacher beliefs data collected. This school-based intervention recruited 17 early primary school teachers to implement a 6-week robotics curriculum in their classrooms. This study aimed to investigate whether this intervention improved teachers' beliefs about programming and robotics education. Furthermore, it explored whether changes in teacher beliefs differed due to the addition of a teacher education workshop that aimed to improve teachers' beliefs. In the previous chapter (Chapter 5), I found that significant improvements in pupil's learning outcomes after the robotics curriculum only occurred if their classroom teacher attended a teacher education workshop prior to delivering the curriculum. Those findings suggested that the teachers themselves may have played an important role in the success of the intervention on children's development of computational thinking skills (specifically prediction and debugging skills). As previous research has suggested that teacher beliefs can affect both classroom teaching practices (Mok & Moore, 2019) and pupil's learning outcomes (Kim & Seo, 2018; Klassen & Tze, 2014), this study also explored whether teacher beliefs impacted improvements in pupil's learning outcomes. This section will discuss the findings in relation to the two research questions outlined in the introduction and additional literature.

Can a 6-week robotics Curriculum Improve Teachers' Beliefs Regarding Programming and Robotics and How do Changes Vary as a Result of Teachers Attending an Additional Teacher Education Workshop?

In this study, two groups of teachers (Intervention+ and Intervention conditions) delivered a 6-week robotics curriculum to their pupils. This curriculum aimed to develop pupil's programming and computational thinking skills and improve teachers' beliefs about teaching these topics. One group of teachers (Intervention+) also attended a teacher education workshop prior to delivering the robotics curriculum. Thus, to investigate how both experiences from teaching the robotics curriculum and the teacher education workshop impacted teachers' beliefs, teachers were asked to complete a beliefs questionnaire at three time points: pre-workshop, post-workshop, and post-intervention. This questionnaire measured teachers' how much they enjoyed teaching programming and robotics, their relevance beliefs, their feelings of anxiety and their self-efficacy.

Enjoyment

The results of this study suggested that the combination of an education workshop and experience delivering the robotics curriculum resulted in significant improvements to teachers' enjoyment. Notably, the improvements in teacher enjoyment in the Intervention+ condition were distinctly larger than the Intervention and Control groups. Furthermore, significant improvements in teacher enjoyment were not found in the Intervention group who did not attend an additional workshop, thus indicating that the addition of the teacher education workshop was crucial for the observed positive effects on enjoyment. These findings are supported by the results of other studies that have investigated the impact of a teacher education training workshop on teachers' enjoyment beliefs. One study (Kim et al., 2015) delivered a 3-

week robotics course to 16 pre-service teachers. Through pre- and post-course surveys and interviews, it was concluded that the 3-week program improved teachers' enjoyment. It was also found that the program improved teachers' motivation and interest towards programming and robotics. Combined with the results from this study, these findings illustrate the positive impact of teacher education on practising teachers' enjoyment beliefs. Furthermore, the findings of this study illustrate that a shorter teacher education program (approximately 3 hrs) can still significantly improve teachers' feelings of enjoyment in this area.

Finally, additional analyses from this study suggested that, for those in the Intervention+ condition, significant improvements in enjoyment occurred between time points 2 and 3 (i.e., post-workshop and post-intervention). This suggests that it was not the teacher education workshop alone that improved teachers' enjoyment beliefs. Instead, it appeared that combining teacher education with real life teaching experiences (i.e., delivery of the robotics curriculum) was most effective for improving enjoyment beliefs.

Relevance

Like the findings relating to teacher enjoyment beliefs, the results of this study suggested that combining a teacher education workshop with the delivery of a classroom robotics curriculum significantly improved teachers' relevance beliefs compared to teachers in the curriculum only condition (Intervention) and the Control group. Relevance beliefs referred to teachers' beliefs about how important programming and robotics education is in early primary school. The findings of this study further supported the notion that the addition of the teacher education workshop was important for improving teachers' personal beliefs.

Further analysis showed that there were marginal improvements in the Intervention+ condition occurred between time points one (pre-workshop) and two (post-workshop). This seems suggests that the teacher education workshop alone was somewhat effective at improving teachers' relevance beliefs, however these findings should be interpreted with caution. During the workshop, teachers heard about the growing movement to integrate computational thinking, programming, and robotics within early primary school classrooms. In discussing how these concepts can be introduced to pupils, they also heard about how robotics have been used in past research studies to develop children's cognitive skills and how this study aimed to contribute to this area of research. The results of this study demonstrated that the sharing knowledge in this way may have the potential to improve teachers' relevance beliefs.

Furthermore, these results suggested that delivery of a robotics curriculum combined with a one-time 3-hour workshop could improve teachers' relevance beliefs. Findings from Chang and Peterson (2018) further support this as they found that a 2-hour educational technology course improved pre-service primary school teachers' beliefs (n = 59) about the importance of teaching programming, computational thinking, and robotics to children (aged 4 to 12). Their conclusions were based on written reflections from teachers following the course. A strength of the current study is that teacher beliefs were assessed using quantitative measures, meaning changes (i.e., improvements) could be tracked across multiple time points.

Anxiety

This study did not find significant reductions in teachers anxiety scores when comparing the three experimental conditions. However, it is worth noting that while improvements in teachers' anxiety beliefs were not significant, the descriptive trends appear to be positive. Descriptive trends suggested a reduction in Intervention+ teachers' anxiety post-workshop, and further reductions post-intervention. While firm conclusions cannot be drawn as these results were not statistically significant, the encouraging trends in the descriptive data illustrate the potential effectiveness of the combined teacher education workshop and robotics curriculum in positively influencing teachers' anxiety beliefs. To gain a more comprehensive understanding of these potential effects, future research could benefit from a larger sample size of teachers. Unfortunately, resources and tight timelines for data collection meant that additional teachers (and their classes of pupils) could not be recruited for this study.

Self-Efficacy

Previous research has illustrated that teacher education programs can improve teachers' self-efficacy beliefs regarding teaching programming and robotics to children in early primary education. One study (Bers et al., 2013) measured changes in teacher beliefs following a 3-day intensive teacher education program (n = 32). The program aimed to help primary school teachers understand how robotics technologies could be used for cross curricular learning with young children and sessions comprised of lectures, small group discussions and hands-on learning with educational robotics. Using surveys and semi-structured interviews, Bers and colleagues (2013) found that their program increased teacher's self-efficacy beliefs.

The current intervention study also investigated whether teachers' self-efficacy beliefs could be improved by a teacher education workshop and/or by teachers delivering a robotics curriculum. Like the study described above, this study found that combining a teachers education workshop with teaching experience of delivering a robotics curriculum significantly improved primary school teachers' self-efficacy beliefs. However, it is again important to note that these improvements were not found amongst teachers who did not attend the additional teacher education workshop. The workshop delivered in this study was designed with the goal of improving teacher self-efficacy in mind. For example, the session provided teachers with opportunities for experiential learning with a Cubetto robot so that they felt comfortable using the equipment before they left the session. Analyses revealed that in the Intervention+ condition, teachers' self-efficacy scores significantly improved after the teacher education workshop but prior to the robotics curriculum. This finding evidences the success of the workshop in increasing teachers' confidence in teaching programming and robotics lessons. Furthermore, an additional strength of this study is that its findings relating to teacher self-efficacy support those from previous research (Bers et al., 2013) and do so whilst utilising a control group in the experimental design.

Do Teacher Beliefs (Post-intervention) Relate to Pupil Achievement Following a 6-week Robotics Curriculum?

As well as looking at how teacher beliefs changed over the course of this intervention, the simultaneous collection of pupil and teacher outcomes data provided an opportunity to explore whether pupil outcomes were influenced by teacher beliefs. Bandura's influential theory has previously been used to argue that teachers with higher self-efficacy may be more effective at increasing pupil achievement (Klassen et al., 2021). However, research investigating possible links between teacher beliefs and pupil achievement has provided mixed results. Meta-analyses have evidenced significant main effects of teacher self-efficacy on pupil achievement (Kim & Seo, 2018; Klassen & Tze, 2014). However, more recently, Jerrim et al., (2023) carried out their own large-scale international study and found no evidence of a link between teacher self-efficacy and pupil achievement.

This intervention study has also produced mixed results concerning the effects of teacher beliefs on pupil's learning outcomes. However, while past research has focused on exploring the potential impact of teachers' self-efficacy beliefs on pupil outcomes, an advantage of this study is that it explored additional dimensions of teacher beliefs (including relevance beliefs, enjoyment, and anxiety). This study found significant main effects of teachers' self-efficacy and anxiety beliefs (post-intervention) on changes in pupil's debugging performance. Thus, results suggested that higher self-efficacy and lower anxiety beliefs were associated with improved debugging performance. Furthermore, this study found that the higher teachers' self-efficacy beliefs were during post-intervention, the more accurate children's answers were on the prediction assessment at post-intervention compared to pre-intervention. However, in contrast to pupil's debugging performance, their improved performance on prediction tasks (specifically the number of correct trials or improvements in the accuracy of their answers) were not related to teachers' self-efficacy beliefs. It is possible that a more substantial sample of teachers and their pupils may yield more definitive results in the future.

Although some of these results suggested that teacher beliefs were in some cases related to children's learning outcomes, a limitation of this study is that it cannot identify the mechanisms relating teacher beliefs and learning outcomes. For example, the literature suggests that teacher self-efficacy may impact pupil achievement in two ways: (1) via improved teaching methods or (2) via role-modelling. Firstly, it has been suggested that higher self-efficacy may lead to better teaching behaviours and classroom practices which may consequently improve pupil outcomes (Lauermann & Butler, 2021). Previous research has related teacher self-efficacy with teacher effectiveness and instructional behaviours (Mok & Moore, 2019). For example, education research has evidenced positive links between teachers' self-efficacy beliefs and their lesson planning abilities and management of classroom behaviours (Chacon, 2005; Woolfolk et al., 1990). Thus, it is possible that, in this study, more confident teachers may have felt more at ease in implementing the robotics curriculum and felt more equipped to tailor curriculum content to suit the needs of their pupils. These improvements to teaching practices may have resulted in the positive changes in pupil's performance on assessment measures.

Alternatively, it is suggested that increased teacher self-efficacy might be conveyed to pupils through role-modelling processes, where pupils observe and mirror teachers' confidence. This heightened self-efficacy in pupils is believed to contribute to increased persistence and, subsequently, positively impact pupil achievement (Lauermann & ten Hagen, 2021). However, given the lack of improvements in pupil-self efficacy in the previous chapter (see Chapter 5), current findings suggest that the first explanation, emphasising the influence of teacher self-efficacy on classroom practices may be more plausible. However, future research should delve deeper into this aspect, exploring specific teacher behaviours and instructional strategies associated with various teacher beliefs that may contribute to improved pupil outcomes. Additionally, future research should not solely concentrate on self-efficacy but should broaden its scope to comprehensively explore how teachers' enjoyment, relevance beliefs, and anxiety may impact pupil learning outcomes. A holistic investigation into these factors, using both quantitative and qualitative methodologies, could provide a more comprehensive understanding of the dynamics influencing pupil achievement in the context of robotics education. Exploring the interplay between teacher beliefs, classroom practices, and pupil outcomes would contribute valuable insights, enabling the development of more nuanced and effective interventions tailored to enhance the overall learning experience for pupils.

Successes and Limitations

There are several methodological strengths to this intervention study. Firstly, this intervention took place in naturalistic school settings, thus increasing the ecological validity and applicability of its findings (Ching & Hsu, 2023). Secondly, the robotics curriculum delivered in this intervention was solely implemented by practising classroom teachers. This is something previous research has lacked as past robotics intervention curriculums have been delivered to pupils by primary researchers or research assistants (see for example Sullivan & Bers, 2015; Sullivan & Bers, 2016; Sullivan et al., 2013; Strawhacker & Bers, 2015). Instead, in some cases (Sullivan & Bers, 2016), it was assumed that teachers' confidence would increase as they observed a research assistant delivering curriculum content. Findings from my previous focus group and survey research (see Chapters 2 and 3) have suggested that learning through observation is not a method favoured by teachers when it comes to their own development. Instead, my findings illustrated teachers' preference for hands-on, experiential learning opportunities. By recruiting teachers to deliver this robotics curriculum themselves, this study not only provided teachers with these experiences, but did so in real life classroom settings. Furthermore, the results of this study illustrated that teachers were able to deliver the entire curriculum to pupils successfully, as illustrated by the increase in reported teaching practices across both intervention conditions.

Another strength of this study is its use of an experimental design and a control group. Several review papers have highlighted that few robotics studies investigating the integration of robotics within education have used experimental designs. For example, Tselegkaridis and Sapounidis (2022) found that 67% of the 36 papers they reviewed used non-experimental designs (i.e., one shot post testing or observation only studies). Similarly, Xia and Zhong (2018) reviewed 22 studies and found that 59% of the studies reviewed used non-experimental designs. The authors of these review papers argued that by not utilising control groups in their research design, researchers cannot fully assess the effectiveness of robotics interventions

(Tselegkaridis & Sapounidis, 2022). Thus, utilising a control group in this study increased the validity of its findings.

It is important to note that recruiting schools for randomised controlled trials can present unique challenges for researchers (Harrington et al., 1997). For example, it cannot be assumed that schools will be happy to participate in research as a control group. Instead, schools may prefer to self-select their conditions and participate in the main treatment/intervention condition as they wish to receive the program that aims to benefit pupil development (Lytle et al., 2001). To minimise the chances of teachers and schools from withdrawing their participation after being placed in the control group, information and consent forms for this study explicitly stated that some teachers and their pupils would be allocated to a control group. These documents required teachers to sign and acknowledge that this was a possibility and that they still wished to participate. To further persuade teachers to continue to participate if allocated to the control group, it was made clear that the control group would receive the same robotics equipment and curriculum once data collection was completed. Furthermore, all teachers were offered a teacher education workshop post-intervention if desired. These methods were successful in ensuring that control group participants completed the project.

However, although participant attrition was not a problem in this study, engaging Control group teachers throughout the entire intervention proved to be difficult. In this study, monitoring forms were sent to Control group teachers to monitor whether they were delivering computational thinking, programming, or robotics activities to their pupils. Unfortunately, the response rate for these forms was very low. As a result, it was not possible to monitor whether teachers and pupils were completing activities that could have impacted this study's outcome variables. Furthermore, this study administered a teacher beliefs questionnaire at three time points (pre-workshop, post-workshop, and post-intervention). As Control group teachers did not complete a teacher education workshop or deliver the robotics curriculum during the intervention period, getting them to complete the beliefs questionnaire at all time points was very difficult. Future research should consider ways to keep control groups engaged at all points in the research process. Perhaps this could be achieved by providing Control group teachers with classroom activities that are unrelated to the intervention's outcomes.

A final strength of this study relates to the design of the teacher education workshop. This education workshop was designed to provide teachers with the knowledge they needed to deliver the curriculum successfully and confidently to their pupils. The design of the teacher education workshop was informed by the previous focus group and survey findings (see Chapters 2 and 3). Past research had not yet used insights from practising primary school teachers to inform intervention design, but coproduced research is likely to be more impactful (Oliver et al., 2015). As recommended by my previous research findings, this workshop had three main components: (1) providing robotics and programming knowledge that is easily applicable within early childhood classroom contexts; (2) teaching content that focuses on interdisciplinary learning and (3) providing structured support while adopting experiential learning approaches. The positive impact of this teacher education workshop on both teacher beliefs and pupil outcomes, as evidenced in this chapter and in Chapter 5, underscores the importance of incorporating these principles into future education design for optimal effectiveness.

Conclusion

In conclusion, this intervention study aimed to investigate the impact of a 6-week robotics curriculum on teachers' beliefs. This included their relevance, self-efficacy, enjoyment, and anxiety beliefs. Furthermore, in this chapter, I explored whether attending a teacher education workshop prior to delivering the robotics curriculum was particularly beneficial for changing teachers' beliefs. To do so, this study employed three experimental conditions: robotics curriculum plus teacher education (Intervention+), robotics curriculum only (Intervention) and a control group. Finally, this study aimed to explore the potential links between teacher beliefs and pupil learning outcomes. It did so by combining the teacher beliefs data explored in this chapter with a subset of the pupil outcomes data from Chapter 5.

The results of this study showed that participating in the Intervention+ condition significantly improved teachers' enjoyment, relevance, and self-efficacy beliefs. Interestingly, these results suggested that the teacher education workshop alone significantly improved teachers' self-efficacy beliefs before they engaged in implementing the intervention in their classrooms. This was likely due to the tailored content of the workshop which aimed to help teachers understand how programming and robotics could be simply integrated within early years classrooms and gave teachers hands-on learning opportunities with the Cubetto technology. On the other hand, results suggested that teacher enjoyment significantly improved once teachers had attended the workshop and had also delivered the robotics curriculum to their class. These trends implied that hands-on, real life teaching experiences with pupils is important for improving teachers' enjoyment in this area of teaching. When investigating potential links between teachers' beliefs and pupils' learning outcomes, this study produced mixed results. It found that improvements in children's debugging performance was related to teachers' self-efficacy and anxiety beliefs. Similar results were not consistently found when analysing teacher beliefs and pupil performance on the prediction task. These mixed findings appear to reflect fluctuating findings in previous literature in this area, however it is important to note that these mixed findings could be due to the small sample size of teachers used in this study.

This study had several unique and innovative elements within its experimental design. Firstly, this intervention was designed for pupils aged 4 to 7 years old and their classroom teachers. Not only was this important given the introduction of computational thinking, programming and robotics into early years education, but previous studies have typically favoured samples of pupils over the age of 8 years (Mangina et al., 2023). Secondly, the robotics curriculum in this intervention was solely delivered by classroom teachers instead of researchers. This was to give teachers hands-on, in classroom teaching experiences. Past research has emphasised the benefits of experiential learning opportunities for teachers' development (Bers et al., 2013; Burke & Hutchins, 2007), however many studies have previously used researchers to deliver robotics curriculums to pupils. This study design also utilised a control group to more accurately determine the effectiveness of the intervention which is something that studies have lacked previously (Tselegkaridis & Sapounidis, 2022). Finally, this study not only integrated a teacher education workshop but designed this workshop based on the feedback and recommendations from practising early education teachers within Wales (see Chapters 2 and 3). This tailored approach to teacher education appears to have contributed to the success of this intervention for both teachers and pupils. In the future, these methods can be used to support teachers and pupils as schools continue to integrate programming and robotics education within classrooms to fulfil goals set by Welsh curriculum guidance.

Chapter 7. General Discussion

Aims of Thesis

International initiatives have emphasised the importance of developing children's digital skills within education and thus computer science has been introduced to primary school curriculums across the world (Balanskat & Engelhardt, 2015; Bers 2020; European Schoolnet, 2015; Uzunboylu et al., 2017). Recent changes to the curriculum in Wales have mirrored this movement, highlighting *'science and technology'* as a key area of learning. Consequently, curriculum guidance emphasises computational thinking (CT) as key skills to introduce starting in early education and illustrates how these skills can be targeted through programming education. In my introductory Chapter, I explored theoretical approaches (i.e., the Montessori approach, constructivism, constructionism) and methodological approaches (i.e. screen-based programs, unplugged learning, educational robotics) to teaching programming to children in the first stage of primary education (i.e., 4 to 7 years old). Past research has shown that educational robotics can be a great tool for introducing these concepts and skills to young children (see for example, Bennie et al., 2015; Sullivan et al., 2017). However, questions remained about *how* robotics could be integrated within early primary school classrooms to optimally benefit children's learning of programming concepts and practices.

This thesis aimed to investigate how this could be achieved and did so by exploring integration from two perspectives: (1) the teacher and (2) the child. Firstly, having an in-depth understanding of teachers' beliefs regarding CT, programming and robotics was a vital step in identifying how to integrate these curriculum concepts and devices into classroom practices more effectively. As CT, programming and robotics are often unfamiliar to early education teachers, introducing these concepts to children can be daunting. Furthermore, if teachers do not feel confident teaching these concepts, or feel these concepts are not important for young children, they will likely neglect this area of learning (Hew & Brush, 2007; Larke, 2019). Chapter 2 of this thesis investigated teachers' beliefs about CT, programming, and robotics through a focus group study and aimed to uncover insights into the digital education landscape in Wales from the perspectives of practising primary school teachers. Chapter 3 then built upon these findings as it explored teacher beliefs and previous teaching methods through a mixed-methods survey. This study aimed to gather insights from a larger and more diverse sample of primary school teachers across Wales.

From a second perspective, my research investigated how children could learn with educational robotics and how these tools could aid the development of CT and other cognitive skills. In Chapter 4, I presented a laboratory study which aimed to investigate how children could learn with Cubetto, the educational robot (designed by Primo Toys), used, and referred to throughout the projects in this thesis. Specifically, this study aimed to explore whether visual perspective taking (VPT) skills were related to children's performance on a selection of programming tasks (i.e., algorithm writing, prediction and debugging) and what role executive functioning played in VPT abilities. I also investigated whether embodied learning techniques aided algorithm writing with a tangible robot. It is important to note that although the grant used to fund the research presented in this thesis was obtained with the co-operation of Primo Toys, and the company provided the Cubetto equipment used in two of these studies (Chapters 4 to 6), the company were not involved in discussions regarding the design of the studies I conducted. Additionally, they did not play a role in the collection nor analysis of the data, interpretation of the results, nor writing of this thesis.

Finally, Chapters 5 and 6 combined teacher and child perspectives and investigated the effects of a school-based robotics intervention on both children's CT skills (i.e., debugging, prediction, sequencing), their programming abilities and their value and self-efficacy beliefs. I also investigated the effects of the intervention on teachers' beliefs (i.e., enjoyment, relevance, anxiety and self-efficacy). A unique strength of this robotics study was that teachers' beliefs and insights (from Chapters 2 and 3) were used to inform the design of the intervention study.

By considering both the perspectives of teachers and children, the overarching goal of this thesis was to provide recommendations for the effective implementation of programming and robotics education, aligned with the new curriculum for Wales. Additionally, this research aimed to offer insights applicable beyond the Welsh context, given the global initiatives to integrate CT, programming and robotics into primary education more broadly. This chapter will provide a summary of the main findings of this thesis, as well as outline important implications, strengths, and limitations of the research. Considerations and future directions for research in the field will also be highlighted.

Summary of Findings

Understanding of Teacher Beliefs about Computational Thinking, Programming, and Robotics Education.

Both my focus group study (Chapter 2) and online survey (Chapter 3) explored primary school teachers' beliefs about CT, programming, and robotics education. These studies also investigated what methods teachers had used previously to teach CT and programming content in early childhood classrooms.

My first key finding was that, on both occasions, teachers highlighted the three main approaches to teaching in this area. These included screen-based methods, unplugged learning activities and educational robotics. I found that teachers liked the versatility of Scratch Jr (www.scratchjr.org), a screen-based programming application, as it could be accessed on a range of devices. On the other hand, at times unplugged methods were preferred as they did not contribute to children's screen time. With regards to educational robotics, my survey found that Bee-bot robots (www.tts-group.co.uk) were most used by teachers. Mainly, teachers liked the physicality of the device and its focus on hands-on play. Hands-on learning activities with robotics have been found to benefit children's learning experiences within programming education as they can be used to improve children's CT skills (see Chapter 5; Bennie et al., 2015; Sullivan et al., 2017; Veenman & Spaans, 2005). The popularity of the Bee-bot robot is mirrored within published research as most studies investigating how robot-mediated learning can promote the development of CT skills in young children have utilised this device (see Bakala et al., 2021 for review).

Teachers in my research also highlighted the suitability of educational robotics for children who were still developing their reading skills. Educational robotics tools designed for young children typically employ a symbolic programming language whereby children learn that different symbols or colours have a corresponding action for the robot (i.e., move forwards, turn left etc., Relkin et al., 2021). Thus, children are not required to read written instructions when using these devices. Instead, they learn the symbolic functions of different buttons.

For my second key finding relating to teachers' beliefs, I found possible links between personal beliefs, teachers' past teaching experiences, and their favoured pedagogies within programming education. Findings from my focus group study (Chapter 2) suggested that those with little experience teaching in this area tended to believe that CT and programming education automatically required the use of technology (i.e., computers or other screen-based technologies like iPads). On the other hand, accounts from those more familiar with the topic suggested that they believed technology was not essential as CT and programming concepts could be taught using tangible or unplugged programming methods. These responses suggested that teacher beliefs about how to teach CT and programming may be linked to teachers' individual experiences. For example, those with more experience teaching CT or programming appeared to be more aware of alternative teaching approaches and thus seemed less likely to hold this belief about the need for traditional, screen-based technologies. These links were supported by findings from my online survey (Chapter 3). In the survey, all teachers that agreed CT could be taught without computers were a part of a minority sub-group that stated they were very familiar with the concept. Those who disagreed or were unsure whether computers were an essential resource were those not at all or only slightly familiar with the concept of CT. These findings illustrated the importance of teacher education programs as a way of preventing teachers' beliefs from limiting their teaching practices (Ertmer, et al., 2012; Larke, 2019).

My third key finding was that overall, teachers in both studies held positive value beliefs regarding the importance and relevance of teaching CT and programming within early primary school classrooms. Thus, participating teachers in these studies appeared supportive of recent curriculum changes in this area (see Welsh Digital Competence Framework; Hwb, 2018). Furthermore, findings from my survey study not only highlighted that teachers believed CT, programming and robotics education was important for developing children's interpersonal and scientific enquiry skills in the short term, but also believed that learning in this area could benefit pupils in the long term. For example, most teachers believed that programming and robotics education could positively impact pupils' lifelong learning skills and could discourage future gender stereotypes (see Chapter 3). This finding that teachers were knowledgeable about both the short-term and long-term benefits of introducing programming and robotics education in early years classrooms was encouraging. Furthermore, my findings illustrated that teachers had a clear desire to teach these concepts to their pupils. Most teachers surveyed wanted to teach CT, programming and robotics lessons more often, and those who had not yet done so were eager to start teaching the subject. Consequently, this raised questions about why teachers were not teaching CT, programming and robotics lessons more frequently, and why some teachers had not taught these lessons at all. To investigate this further, the first two studies in this thesis also explored teachers' beliefs about barriers that may limit CT, programming, and robotics education in early years education.

Perceived Barriers to Programming Education

My first two studies identified a range of first and second-order barriers that teachers believed hindered their teaching instruction in this area of learning. First-order barriers have been described by Ertmer (1999) as factors external to the teacher. First-order barriers identified in my research included pupils' skills (i.e., technical skills), external barriers (i.e., lack of resources, time restrictions), technology barriers (i.e., program visualisation problems, reliance on screens) and teacher education factors (i.e., the content and structure of education programs).

Some of the barriers identified in my studies reflected those found in previous research. For example, teachers identified barriers including access to resources (Khanlari, 2016), cost of resources (Chang et al., 2010) and a lack of technical and content support (Greifenstein et al., 2021). However, my studies also offered further insights into other barriers to CT, programming and robotics integration. Firstly, in both the focus group and survey studies, teachers discussed the potential limitations of pupils' technology skills. For example, teachers who had approached CT and programming education using computers and laptops had found that young children struggled to use these devices and thus could not access the lesson content. This problem likely stemmed from the fact that keyboard-based devices require sufficient fine motor skills to operate the keyboard or mouse, as well as a certain level of cognitive development to understand keyboard symbols (Geist, 2014). Additionally, children must be capable of logging into these devices to access the chosen programming software. The need for these skills can render such devices developmentally inappropriate for very young children. Thus, these pupil-related barriers may prevent teachers from delivering CT and programming lessons, particularly if they only have access to these screen-based devices. These findings appeared to highlight the potential advantages of using educational robotics to teach CT and programming concepts as their tangible programming languages do not require the same level of cognitive development or fine motor skills as screen-based technologies.

My research findings have also provided additional insights into the effects of external barriers. Previous research has identified 'time' as a barrier to programming and robotics education (see Kahnlari, 2016 for example) and further explained that this was due to timetable restrictions and an inability to fit in additional subjects. My interactions with primary school teachers evidenced that they already aimed to integrate programming and robotics with other classroom subjects and thus were less concerned about finding time within lesson plans. Instead, the results of the survey illustrated that the limitation of 'time' may be related to the time needed to build teachers' knowledge in this area rather than finding the time to teach the content. Both the focus group and survey participants highlighted that there is typically an expectation for teachers to develop their knowledge and skills in their own time. This is something that teachers have struggled to find the time for in the past. This finding expanded our understanding of time as a barrier by highlighting how time barriers may not only be related to the time spent teaching inside the classroom. Furthermore, these findings emphasise the importance of providing teachers with structured time within teacher education programs to fully understand the content and consider how the content could be transferred into their classrooms. This was incorporated into the design of the teacher education workshop delivered in my intervention study (Chapter 6).

While exploring teachers' previous teaching practices, the focus group and survey studies also explored teachers' beliefs about the limitations of those teaching methods. These accounts provided insights into the potential disadvantages of screen-based approaches (like Scratch Jr) and robotics (like Bee-bot) that were specific to the designs of the technologies themselves. For example, some teachers in the focus group raised concerns about pupils' screen time and shared that their schools had made conscious efforts to employ methods of teaching that did not use screens. Thus, those teachers were seeking alternative approaches to CT and programming education. For those using Bee-bot robots to approach learning in this area, teachers proposed that the lack of program visualisation can make learning more difficult for younger pupils. They explained that to operate Bee-bot robots, children input their sequence of instructions into the robot using small buttons on top of the device. This is then stored internally with no visual presentation of the sequence. This interface can be difficult for children to operate as children are required to remember their programmed sequence, thus increasing the cognitive demands of this task. Teachers shared that, in their experiences, this can be difficult for children as they forget what they instructed the robot to do. These findings contribute to our understanding of these approaches by exploring the limits of these technologies through the perspectives of those who use them in classrooms.

My final key finding relating to teachers' beliefs about first-order barriers is that teachers believed past teacher education programs (specifically for CT, programming, and robotics) were not designed to meet the needs of those teaching pupils under the age of 8 years old. On the one hand, some teachers who had attended education courses found that the content was aimed at those teaching children in later primary years (i.e., older than 8 years old). On the other hand, some teachers had not had the opportunity to attend any form of education program to learn about teaching CT, programming, or robotics. This conclusion was concerning given the government's focus on developing children's digital skills in early schooling (Hwb, 2018; 2024a). These findings supported the previously argued notion that early childhood teachers are not given the same teacher education opportunities as those teaching older children, specifically when it comes to STEM (science, technology, engineering, and mathematics) subjects (Bers, 2010; Bers & Portsmore, 2005; Sullivan & Moriarty, 2009). Thus, my focus group and survey studies highlighted unequal teacher education opportunities that must be addressed to ensure teachers are delivering CT, programming, and robotics lessons to the best of their ability to achieve the prescribed curriculum goals.

In this thesis, I also explored teachers' beliefs about second-order barriers to CT, programming, and robotics education. Second-order barriers typically comprise of barriers specific to the internal beliefs of the teacher (Ertmer, 1999), such as their beliefs about things like curriculum priorities and teaching ability. Previous research has found that low self-efficacy (i.e., self-confidence) for programming and robotics is commonly found in samples of primary school teachers (see, for example, Khanlari, 2016; Ohashi et al., 2018; Ray et al., 2020). The findings from my focus group and survey studies mirrored this notion as most teachers reported that they did not feel confident teaching programming and robotics lessons to their pupils. However, the results of my survey explored this further. Those results showed that although some teachers did not feel confident in their teaching ability at the time of data collection, they later indicated that they were confident that they could *learn* how to teach these topics to their pupils successfully. Thus, these findings expanded upon those of previous research, suggesting that teacher self-efficacy beliefs may not be a long-term barrier to programming and robotics integration as teachers remained optimistic about their ability to introduce these topics to their teaching in the future. These findings further illustrated the importance of providing teacher education programs to boost teachers' confidence in this area. Findings from my later intervention study evidenced that teacher education workshops can improve teacher confidence.

How Can Young Children Learn Computational Thinking and Programming with Robotics?

In Chapter 4, I presented a laboratory study which explored children's learning with a Cubetto robot. As educational robotics incorporate spatial movement within a child's physical environment, I investigated whether perspective taking skills were important for successful programming of a Cubetto robot. For instance, when children write algorithms for a robot to follow, this may be more difficult when the robot's spatial orientation does not align with the programmer's as this must be corrected mentally as they pre-plan their algorithm. Thus, I investigated whether visual perspective taking (VPT) skills were related to children's performance on a selection of programming tasks (i.e., Algorithm Writing, Prediction and Debugging).

Readings of previous studies also suggested that executive functioning skills may play an important role in VPT (Diamond et al., 2002; Frick & Baumeler, 2017). For instance, when children are presented with a conflict between two perspectives in the same physical environment, children's own perspective (the conflicting information) must be ignored for them to be successful at imagining the alternative perspective. This would require inhibitory control skills. Thus, I also explored whether performance on a VPT task would be associated with performance on an executive functioning task.

Finally, this study applied embodied learning principles to learning with a Cubetto robot. Previous literature suggested that aligning the orientation of a moving object with the participant's viewpoint could improve the participant's ability to navigate the object as they no longer needed to mentally correct the incongruent perspective (Cho et al., 2017). Thus, I investigated whether embodied learning methods improved children's perception of robot movements by aligning their visual perspectives, thus helping them write accurate algorithms.

In this study, 78 children (aged 4 to 7) completed a collection of cognitive tasks. These tasks included programming games with Cubetto, a VPT task using coloured blocks and an iPad-based executive function measure. To complete an algorithm writing task with Cubetto, children were allocated to one of three conditions: (1) Incongruent programming (Cubetto's spatial orientation did not match the child's), (2) Congruent programming (the child and Cubetto both faced the same way at the start of the task) and (3) Embodied programming (the child moved through the space as they programmed Cubetto as if they were Cubetto).

This study did not find evidence for a link between children's programming performance and their performance on a perspective taking task. Furthermore, children's programming or perspective taking abilities were not found to be related to their executive functioning skills. Finally, embodied learning was not found to aid programming performance or performance on programming-related assessments (i.e., prediction or debugging measures) in this study. Following these findings, the intervention study (presented in Chapters 5 and 6) did not investigate VPT and embodied learning further. Instead, this school intervention investigated skills more traditionally associated with CT (i.e., sequencing, prediction and debugging). Although the findings from the laboratory study did not support my initial predictions, the study did evidence the suitability of the Cubetto robot for children aged 4 to 7 years old. Observations made in this study were that children enjoyed using the robot. Furthermore, results showed that most children were able to progress through several programming-related assessments, thus suggesting that children were able to understand how the device worked despite not having used it before. After helping children learn with Cubetto, I was able to advise teachers on how to introduce the robot to children in my intervention study (Chapters 5 and 6).

Successfully Implementing Educational Robotics in the Classroom

In Chapters 5 and 6, I presented an intervention study that was specifically designed for children under the age of 8 years, was delivered by teachers instead of researchers and utilised a control group. Recent literature reviews have found that studies in these areas are lacking these design qualities (Mangina et al., 2023; Tselegkaridis & Sapounidis, 2022). During the intervention, a 6-week Cubetto curriculum was implemented within 19 early years classrooms in South Wales, with data collected from 430 children (aged 4 to 7) and 15 classroom teachers at multiple time points. Schools were assigned to one of three conditions: (1) Intervention+ (pupils completed the robotics curriculum, teachers attended a teacher education workshop before intervention), (2) Intervention (pupils completed robotics curriculum, teachers did not attend an education workshop), (3) Control group (no robotics curriculum, no teacher education). The design of the teacher education workshop was informed by the previous focus group and survey findings (see Chapters 2 and 3). The workshop had three main goals: (1) to provide robotics and programming knowledge that is easily applicable within early childhood classroom contexts; (2) to teach content that focuses on interdisciplinary learning and (3) to provide structured support while adopting experiential learning approaches (i.e., proving hands on learning experiences with the Cubetto equipment). During data collection, children completed pre- and post-intervention cognitive assessments including CT assessments (for debugging and algorithm prediction skills), a picture-sequencing task, a beliefs questionnaire, an iPad-based programming task (Lightbot Jr) and an executive functioning assessment (the Minnesota Executive Function Scale). Teachers completed beliefs questionnaires at three time points: pre-workshop, post-workshop, and post-intervention. This questionnaire measured teachers' enjoyment (i.e., how much teachers enjoyed teaching programming and robotics), relevance (i.e., how important they believed programming was for their pupils), anxiety (i.e., how anxious they feel teaching CT, programming and robotics) and self-efficacy beliefs (i.e., how confident they feel teaching programming and robotics lessons).

The first key finding from this study was that combining a 6-week robotics curriculum with a teacher education workshop (Intervention+ condition) significantly improved children's debugging and algorithm prediction skills compared to the Control condition. This highlights the importance of providing effective teacher education alongside curriculum content as delivering the curriculum alone (the Intervention condition) did not significantly improve these variables in comparison to a Control group. Furthermore, teachers in the Intervention+ condition showed significantly larger improvements in enjoyment scores, relevance scores and self-efficacy scores as measured with a beliefs questionnaire. Expanding on this, analyses revealed that self-efficacy beliefs significantly improved following the teacher education workshop, even before the delivery of the robotics curriculum. Furthermore, results suggested that teacher enjoyment significantly improved once teachers had both attended the education workshop and delivered the robotics curriculum. These findings suggested that the addition of real-life teaching experiences were important for improving teachers' enjoyment, relevance and self-efficacy beliefs. When exploring the potential impact of teacher beliefs on pupil performance, I found that teachers' self-efficacy beliefs (post intervention) were related to pupil improvements on debugging and prediction tasks (between pre- and post-intervention assessments). These findings supported previous meta-analyses that have evidenced significant main effects of teacher self-efficacy on pupil achievement more broadly (Kim & Seo, 2018; Klassen & Tze, 2014).

The results of this intervention study evidenced that following my recommendations for teacher education programs (first presented in Chapters 2 and 3) can positively impact teachers' beliefs and pupils' learning outcomes. For example, teachers' improvements in self-efficacy beliefs mid intervention were likely due to the tailored content of the workshop which aimed to help teachers understand how programming and robotics could be integrated within early years classrooms specifically and gave them hands-on learning opportunities with the robotics equipment (Konen & Horton, 2000). Overall, these findings illustrate the importance of incorporating my three teacher education recommendations into the design of future education programs for optimal effectiveness.

Implications of Thesis Findings for Theory

The research presented in this thesis significantly contributes to the field of CT, programming, and robotics by extending existing knowledge and addressing critical gaps in past research. Jeannette Wing (2006) defined CT as a fundamental skill for everyone (not just computer scientists), emphasising enhancing problem solving skills for application in various aspects of life.

On the one hand, the research presented in this thesis supports the notion that CT is for all, as I have demonstrated how children as young as 4 years old can develop some CT skills through programming education. Specifically, findings from my intervention study suggest that integrating educational robotics into the curriculum can aid children's understanding of abstract CT concepts such as algorithms, debugging, and prediction (Bers, 2020). It appears that the tangible nature of these devices can make CT and programming concepts accessible to young learners. This supports Montessori (1967) and constructionist (Piaget & Cook, 1952) theories of hands-on learning. Thus, this thesis contributes to the growing body of literature that promotes the use of robotics in early years classrooms (for example, Kazakoff et al., 2013; Misirli & Komis, 2023; Pugnali et al., 2017; Strawhacker et al., 2013). Furthermore, this research advances this area of research due to the robust design of the school intervention. For example, my intervention study not only utilised a control group but was also delivered by teachers rather than researchers and was specifically designed for children under the age of 7. Previous research lacks such study designs (Mangina et al., 2023; Tselegkaridis & Sapounidis, 2022).

On the other hand, I did not find that CT skills were necessarily transferred to all learning contexts, suggesting that there may be limits to the development of CT using robotics over a 6-week period. My intervention study found evidence of "near transfer" as improvements in CT skills (i.e., debugging and prediction) were closely related to the context in which they were learnt. Near transfer requires similar contexts and the performance of similar skills and strategies (Perkins & Salomon, 1992). In the intervention study, both the debugging and algorithm prediction tasks assessed skills specifically targeted within the given Cubetto curriculum. The remaining CT assessments (i.e., picture-sequencing and algorithm writing with Lightbot Jr) provided different contexts to the Cubetto curriculum activities as they were not explicitly linked to the robot. Thus, for performance on the sequencing and algorithm writing tasks to improve, this "far" transfer of knowledge would have had to have been an automatic and spontaneous process which is unlikely to occur within this age range (Gutiérrez-Núñez et al., 2022; Hajian, 2019; Salomon & Perkins, 1989). The findings of my intervention study suggested that the generalisation of CT skills to various learning contexts may not be as easy as Wing suggested. Thus, additional scaffolding (i.e., structured support and reinforcement from teachers) may be necessary to facilitate the transfer of knowledge to broader CT skills in unfamiliar contexts. Future interventions should consider incorporating ongoing support mechanisms to help children apply their newly acquired knowledge and skills across different domains in daily life.

This research also contributed to our understanding of teachers' beliefs about CT, programming and robotics education by recruiting samples of practising primary school teachers. This is a demographic often overlooked in previous studies which have typically employed samples of pre-service teachers (i.e., those still completing their teaching qualifications; see for example Chang & Peterson, 2018; Kim et al., 2015). Conducting research with practising teachers as well as pre-service teachers is important as teaching experiences between these two groups may vary. For example, classroom dynamics, student diversity, and the evolving demands of day-to-day teaching may significantly shape teachers' beliefs and practices, and pre-service teachers may not yet have had the opportunity to fully navigate these complexities. This limited exposure to real-world teaching scenarios may bias their perspectives on incorporating robotics into their (future) teaching practices. By conducting research with practising primary school teachers, this research enriched our understanding of the challenges they encounter when attempting to integrate CT, programming, and robotics into primary education.

Implications of Thesis Findings for Educational Practice

(Inter)National Context

Although the research in this thesis was conducted within Wales and can be framed within the Welsh curriculum, the results of these studies are likely applicable more broadly as many countries share similar educational goals aimed at preparing pupils for the challenges of the 21st century, which often include fostering CT and digital literacy (European Commission, 2022). Additionally, when it comes to teaching CT in primary education, the underlying pedagogical principles, such as inquiry-based learning, hands-on experimentation, and collaborative problem-solving are widely applicable. In a recent review of CT in compulsory education (European Commission, 2022), it was noted that primary school teachers across Europe introduce pupils to basic concepts using hands-on, playful activities with educational robotics and virtual block environments. These approaches reflect those used by participating teachers in this thesis (see Chapters 2 and 3). Thus, insights into the successful implementation of these pedagogical strategies in the Welsh context can inform educators in other countries seeking to adopt similar methods.

Furthermore, previous research has illustrated that some of the challenges associated with technology integration, including access to resources, teacher education and curriculum alignment, are not unique to the Welsh context (see for example Chang et al., 2010; Greifenstein et al., 2021; Khanlari, 2014). Instead, teachers worldwide face these same barriers as they seek to introduce new technologies (i.e., educational robotics) to enhance teaching and learning experiences. This is illustrated in a recent report from the European Commission (2022) which explored the challenges posed by the integration of CT skills within compulsory education. After administering a survey across 24 European countries, it was found that most respondents (across 18 countries) mentioned a *'lack of adequately trained teachers'* as a challenge in primary education. Teachers who participated in my earlier conducted focus group and survey research also highlighted a lack of teacher education as a concern and potential barrier to teaching CT, programming, and robotics. Following these studies, I was able to identify several recommendations that could be implemented within teacher education programs to improve teacher readiness in these areas. These were then integrated into the design of my later robotics intervention which included a teacher education workshop. It is worth noting that the workshop was designed to be impactful yet concise, lasting only three hours. This intentional brevity was aimed at accommodating the busy schedules and time constraints often faced by educators. Despite its brief duration, the workshop proved highly effective in positively influencing both teacher beliefs and students' learning outcomes (when combined with a 6-week robotics curriculum).

The 2022 report from the European Commission also identified *'competition with other curriculum priorities'* as a barrier to the integration of CT in primary education. While this was also raised as a concern in my previous focus group and survey studies, the robotics curriculum used in my intervention study provides an example of how CT can be taught in a cross-curricular manner with other classroom subjects. Finally, the report indicated that the *'assessment of computational thinking/ programming'* skills is also a concern for teachers (mentioned in 10 countries). Previous research (see Bakala et al., 2021 for review) has favoured portfolio-style assessments of children's abilities, often including hands-on testing with the chosen programming equipment (i.e., educational robotics). Such assessments may not be practical for classroom teachers due to time restrictions and class size. Thus, another strength of my intervention study was its development of paper-based CT assessments. For teachers' use in the future, paper assessments are advantageous for several reasons. Firstly, they are more time effective as a large group of children can be assessed at the same time. Additionally, paper handouts can be marked easily and can be used for evidencing learning.

In summary, the findings of this thesis may provide valuable guidance for teachers and schools facing similar obstacles in other countries. Insights from the Welsh context can thus contribute to a collective understanding of effective strategies for integrating CT, programming, and robotics into primary education, enriching the global knowledge base, and fostering continuous improvement in educational practices worldwide.

Recommendations for Teacher Education Programs

By analysing the experiences and beliefs of teachers, this thesis has identified several recommendations for teacher education that future CT, programming and robotics programs should integrate. Based on the findings from Chapters 2 and 3, I proposed that teacher education in these areas should (1) provide robotics and programming knowledge that is developmentally appropriate and easily applicable within early childhood classroom contexts; (2) teach content that focuses on interdisciplinary learning and (3) provide structured support while adopting experiential learning approaches. These recommendations were then integrated into the teacher education workshop delivered during the school intervention study (Chapters 5 and 6). The findings of the intervention study found positive effects of integrating the three teacher workshop recommendations on teacher beliefs and children's programming related CT skills. These recommendations are now outlined in more detail.

Developmentally Appropriate Content. Firstly, in my focus group and survey studies (Chapters 2 and 3), I recommended that teacher education programs should aim to aid teachers' understanding of how to deliver CT, programming and robotics lessons that are age-appropriate for their pupils. Findings from these studies illustrated that past teaching programs have not achieved this and have instead provided teachers with materials and teaching content designed for older primary school teachers. Following these sessions, early years teachers have then struggled to transfer this knowledge to their classrooms and into their practices. Thus, it is important that teacher education programs are also specifically designed for those teaching children in the first phase of primary education (i.e., under the age of 8). Improving the relevance of teacher education content in this way is likely to lead to improved transfer into the classroom (Axtell et al., 1997).

To implement this in my intervention study, the teacher education workshop included activities that guided teachers to think of and create age-appropriate materials and activities for their young pupils. Additionally, content in the workshop was covered using clear and jargon free language, suitable for both teachers and their young pupils. This was done so teachers could easily access the content and could understand how to later pass on this knowledge and explain CT, programming and robotics concepts in child-friendly ways. These examples illustrate how future teacher education programs can provide learning opportunities that are relevant for those teaching younger children.

Interdisciplinary Learning. My second recommendation for teacher education programs is that they should demonstrate how to teach programming, not as an individual skill, but instead as an approach to developing knowledge of other subjects and general CT skills. To achieve this, my teacher education workshop (Chapter 6) took an interdisciplinary approach, combining programming and robotics technologies with other classroom subjects (e.g., mathematics, literacy, art). I proposed that providing formal guidance on how to integrate CT, programming and robotics within other classroom subjects would likely increase the frequency of these sessions as teachers do not need to find the time to schedule these sessions separately. Furthermore, by integrating programming education with other classroom subjects, teachers may not need to prioritise other subjects over these technology skills. Discussions in the focus group study (Chapter 2) suggested that some teachers have previously removed technology lessons from timetables to cover other core subjects. Thus, interdisciplinary approaches may aid teachers' understanding of the versatility of programming education and robotics technologies (Greifenstein et al., 2021).

Experiential Learning. Alongside the relevance of the content of teacher education programs, it is important that programs also consider the structure of the support being offered to teachers. Chapters 2 and 3 highlighted the expectations placed upon teachers to develop their knowledge of CT, programming and robotics in their own time. In some cases, even after attending teacher education sessions, teachers were expected to revisit the technologies used in their own time. To help reduce these additional demands on individual teachers, teacher education programs should provide teachers with hands-on experiences with the technologies featured. Past research has evidenced that hands-on training is advantageous (Agatolio et al., 2017; Kim et al., 2015) and has identified hands-on practice as one of the "best practices" in educational robotics teacher training courses (Schina et al., 2021).

In the intervention workshop (see Chapter 6), providing hands-on learning opportunities with the robotics equipment gave teachers space to think about what would work best for them in practice, and how they could tailor sessions to meet the needs of their pupils specifically. This was important as teaching is not a “one size fits all” exercise. Instead, teachers deliver lessons in different ways depending on the abilities of their pupils and the types of resources and technologies available to them. This personalised approach aligns with the flexible, learner-centred New Curriculum for Wales, which encourages schools to design curricula reflecting their pupils' needs and interests (Hwb, 2024c). Emphasising digital skills, the government aims for pupils to achieve "digital competence" by school completion (Hwb, 2018). Since formal guidance from government bodies is lacking, teacher education programs that offer experiential learning and lesson planning opportunities are crucial.

Teacher Education vs Training. Overall, these findings emphasise the importance of prioritising teacher education over mere teacher training. Teacher training often reduces effective teaching to the replication of predetermined tasks (Stephens et al., 2004), leading to difficulties when teachers find these activities unsuitable for their pupils. This is evident in accounts from teachers (Chapters 2 and 3) who struggled to adapt training content for younger pupils. In contrast, teacher education has a broad focus on intellectual and personal development and typically includes a combination of theoretical and practical learning to encourage reflection, analysis, and a deeper understanding of content. This may help teachers apply knowledge to a diverse range of contexts as they feel more confident in adapting instructional strategies to suit the specific needs of their classrooms. The three teacher education recommendations outlined above align with this notion of teacher education. Thus, the positive findings from the intervention study illustrate the benefits of structuring teacher workshops in a way that supports teacher education rather than providing training in replicable activities.

Dissemination of Findings

My dissemination plan is designed to effectively reach academics, teachers, and policymakers, ensuring that the research findings are practically applied. To engage the academic community, I will publish in peer-reviewed journals and present at conferences focused on education and educational technologies. Publishing research in journal articles is important for validating the research through peer review and contributing to the academic body of knowledge. However, as noted by O’Connor et al. (2021), journal articles can be challenging for teachers to access due to their length, complexity, and the barrier of paywalls.

This poses a challenge for teachers, who often have limited time, to apply these insights in their classrooms.

To better reach teachers, I will create infographics and visual summaries. O'Connor et al. (2021) highlight that visual strategies like these can help translate complex research into clear and concise information that early years teachers can easily implement. Additionally, to reach a broader audience—including academics, teachers, families, and policymakers—I will share the findings through social media and online platforms. Social media provides a wide-reaching and immediate method to disseminate research and engage with various audiences.

Finally, to further extend the research's reach, collaboration with Educational Psychologists, lecturers in education and digital learning programs like Technocamps (based in Wales, www.technocamps.com) is recommended to further apply the recommendations for teacher education that are outlined above. This will help ensure educators are equipped with the necessary skills to teach computational thinking, programming, and robotics effectively in early years education.

These approaches align with recommendations to use multiple methods to disseminate early years research findings to ensure the research reaches all relevant stakeholders effectively (O'Connor et al., 2021). By combining these strategies, this plan ensures that the research not only contributes to academic knowledge but also influences educational practices and policies, benefiting a wider community.

Strengths and Limitations of the Thesis

One of the key strengths of this thesis is the combination of qualitative and quantitative methods. I used both a focus group study and a survey study to explore teachers' beliefs about CT, programming, and robotics in early primary education. This mixed methods approach allowed for a more comprehensive understanding of teachers' beliefs by utilising the strengths of both qualitative and quantitative methods. On the one hand, conducting the focus group study allowed me to explore teachers' beliefs and experiences in depth, capturing rich, nuanced insights into their beliefs about CT, programming, and robotics. However, the generalisability of these findings was potentially limited by the small sample size. Thus, I conducted an online survey study which allowed for broader data collection from a larger sample of teachers. This improved the generalisability of my findings and validated my understanding of teacher beliefs, particularly because the findings of the original focus group were supported by the data collected in the more widely distributed survey.

When first investigating how children could learn with a Cubetto robot, I did so in a laboratory study. Conducting research in a lab environment offered several advantages, including the ability to control variables such as noise and other external distractions. Additionally, this controlled setting allowed me to manipulate embodiment conditions while ensuring all other task instructions and setups remained the same. However, with this level of control over experimental conditions, the ecological validity of the findings can be impacted. This was especially relevant within this research as children would not ordinarily learn on a 1:1 basis in a silent room with no other distractions. Rather, children typically learn in a classroom environment with their peers. Thus, engaging children with a Cubetto robot in an isolated room could not confirm that these devices would work well in a real-life classroom setting. Therefore, to address this limitation, the second key strength of this thesis was its inclusion of an intervention study which took place within a selection of primary schools. By investigating children's learning with Cubetto robots in these authentic educational settings, I could better understand how these devices could be integrated into classroom instruction and how they could benefit learning outcomes in real-world contexts.

When it comes to exploring the effectiveness of my classroom intervention, the study could be criticised for its small sample of primary school teachers. The study found positive effects of a teacher education workshop and robotics curriculum on teachers' enjoyment, relevance and self-efficacy beliefs; however, these significant improvements were only found in the four teachers allocated to the Intervention+ condition. Although it is compelling that we found positive effects of the intervention despite the small sample size, replication with a larger sample of teachers would be beneficial to further investigate these effects on teacher beliefs.

As the intervention study also collected data from primary school pupils, recruiting more teachers would have resulted in a larger sample of pupils which would have increased numbers for data collection and thus increased school visits and travelling. Therefore, unfortunately, increasing the sample of teachers was not possible for the study due to time and funding restrictions. That being said, a strength of the intervention study presented in Chapters 5 and 6 is that the study recruited a large sample of primary school children (n = 430). To my knowledge, there have been no other studies that have recruited a similar sample size for a programming and robotics intervention, particularly within early primary education. Instead, review papers have concluded that most studies in this area have previously recruited samples of fewer than 80 children (Bakala et al., 2021; Xia & Zhong, 2018). The successful organisation and execution of a large-scale robotics intervention in early primary classrooms (for pupils and their teachers) is another key strength of this thesis.

When further exploring the effectiveness of the intervention (presented in Chapters 5 and 6), it is important to note that confounding variables could have impacted both pupil outcomes and teacher beliefs. For example, the implementation of the 6-week Cubetto curriculum could have been impacted by various classroom-related variables. For example, class size could have impacted pupils' learning experiences and teachers' experiences of delivering activities. Additionally, school-level factors (i.e., school funding, resources, and organisational structure) have been found to affect pupil achievement (Hofman et al., 2002) and could also impact teacher beliefs. However, another strength of this thesis was the analysis method employed to analyse the results for the two intervention chapters.

Multilevel modelling was used as it is an appropriate statistical technique when data is nested within a hierarchal structure (i.e., pupils nested within teachers, within schools, within a condition; Hox, 1998). By incorporating this hierarchical structure into the analysis, multilevel modelling allowed for the estimation of variance at each level of the hierarchy, capturing the unique contributions of individual students, teachers, schools, and experimental conditions to the outcomes of interest. This approach helped to control for potential confounding variables that may have occurred in differences between schools, classrooms, or teachers, thereby improving the accuracy and reliability of the estimates of intervention effects. While specific covariates were not included in the analysis due to data limitations, the use of multilevel modelling still provided a rigorous method for examining the effectiveness of the intervention while accounting for the hierarchical nature of the data and potential confounding variables.

Finally, although the research in this study was conducted within Welsh primary schools with practising teachers in Wales, for reasons I have explained above, the findings of this thesis will likely have implications beyond the Welsh context. For example, not only do other countries inside and outside of Europe have similar curriculum goals (regarding the integration of CT and programming), but other countries also favour similar pedagogical approaches and teachers have shared similar experiences of barriers to CT and programming education (European Commission, 2022). Thus, the key findings of this thesis can be used by other countries to improve the integration of these topics within early primary school classrooms.

Future Directions

Future studies could extend this research in several ways. Firstly, future research should consider investigating the long-term benefits of introducing programming and robotics in early childhood education. One paper (Futschek & Moschitz, 2011) proposed activities to teach children (from age 5) algorithmic thinking skills using unplugged programming methods (i.e., tangible objects and activities). At the end of the paper, the authors concluded that the tangible activities could then help children transition their knowledge into a virtual programming environment (i.e., Scratch). However, it is important to emphasise that these conclusions were not derived from empirical data during which authors assessed children's learning but instead were based on author theories. To my knowledge, research has yet to investigate whether learning a simplified programming language in the early years of education helps children later learn more complex programming languages. For example, it would be interesting to explore whether learning with a Cubetto robot helps children progress to Bee-bot to Scratch Jr to Scratch to HTML to Python or JavaScript and so on. In my intervention study, I did not find evidence of transfer from programming with Cubetto to programming with Lightbot Jr (an iPad application), however, it would be interesting to investigate whether this kind of transfer could be developed over a longer period, and what additional support may assist with this transfer.

My second recommendation for future research focuses on the role of teacher education programs when combined with real-life classroom experiences. My studies have highlighted key recommendations for teacher education programs including (1) developmentally appropriate content; (2) interdisciplinary learning and (3) experiential learning approaches. My intervention research then evidenced the benefits of using these recommendations to guide the design of a workshop relating to CT, programming and robotics in primary education. It also illustrated the benefits of combining teacher education with supported, in classroom teaching experiences. Future research should explore whether applying these recommendations to teacher education in other subject areas and combining these experiences with classroom teaching experiences, continues to benefit pupils' learning and teachers' beliefs in the chosen topic area.

Finally, future research should also consider investigating how educational robotics can be used as a tool to engage children with additional learning needs. In this thesis, I was interested in exploring how robotics could be used to improve learning outcomes in typically developing children, however anecdotal feedback from some intervention teachers suggested that the Cubetto robots positively engaged children who struggled to engage with learning or were learning English as an additional language. One survey (Di Battista et al., 2020) study found that primary school teachers (n = 323) generally believed that educational robotics could be a powerful tool for engaging children with a range of additional needs, especially for those with Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorder (ASD), and Dyspraxia. Kindergarten teachers specifically identified robotics as particularly helpful for ASD, ADHD, Down Syndrome (DS), psychological or emotional distress, and the needs of foreign students. Additionally, primary school teachers noted the effectiveness of robotics primarily for pupils with ADHD, Dyspraxia, and ASD (Di Battista et al., 2020). These findings and the feedback collected from teachers in my intervention study both illustrate that teachers believe educational robotics could be useful learning tools for children with additional learning needs. However, to my knowledge, research is yet to investigate how robotics can be used to improve learning outcomes in these pupils using an intervention design like the one used in this thesis. Intervention research focusing on children with additional learning needs would ensure that educational interventions meet the diverse needs of all learners, thus contributing to more inclusive educational practices.

As of January 2024, 11.2% of pupils in local authority maintained schools in Wales were identified as having additional learning needs (ALN) or special educational needs (SEN; Welsh Government 2024b). The principles of the Additional Learning Needs (ALN) Code for Wales align closely with the aims of the research in this thesis, particularly in promoting inclusivity and tailored support within mainstream classrooms (Welsh Government, 2021). Teachers suggesting that educational robotics can engage children with ALN supports the Code's objective of integrating these learners into mainstream educational settings. Thus, the intervention (presented in Chapters 5 and 6) could serve as a valuable approach for inclusive education, enabling teachers to better meet the diverse needs of their students.

However, while teacher feedback suggests that educational robotics are suitable tools for engaging ALN children, further research is needed to explore which robot designs are appropriate for children with varying learning needs. Additionally, researchers and teachers may need to consider adapting the assessment measures used in the intervention study to better suit pupils' individual needs, such as incorporating differentiated instruction or providing alternative formats. Future research should focus on tailoring robotics-based interventions for children with ALN, addressing potential barriers to accessibility, and ensuring that all learners can benefit from such programs. By aligning with the ALN Code's goals, further research would not only advance our understanding of educational robotics but would also contribute to the development of more inclusive educational practices in Wales.

Final Conclusions

The research in this thesis has explored the integration of educational robotics in early years classrooms and has shed light on the crucial role of teacher perspectives in shaping the effectiveness of such educational initiatives. My findings have highlighted the recognition amongst Welsh primary school teachers of the value and importance of CT programming and robotics education in early education, however, many felt unsure about their ability to teach these topics effectively. Nonetheless, teachers exhibited a strong willingness and confidence in their ability to learn and enhance their skills in this domain. A standout finding in my earlier studies was the clear need for effective education programs tailored to support teachers in unlocking their potential in delivering programming and robotics content. Recommendations emerged from my research, emphasising the importance of developmentally appropriate content, hands-on learning opportunities and interdisciplinary approaches within teacher education programs. I found that the integration of these recommendations into an intervention study yielded promising outcomes, with teacher education workshops combined with real-life teaching experiences significantly enhancing teachers' relevance, enjoyment and self-efficacy beliefs in teaching programming and robotics education. Furthermore, my intervention research also explored the impact of educational robotics on pupils' learning outcomes in primary schools across South Wales. My findings indicated that a six-week robotics curriculum improved children's CT-related skills (i.e., debugging and prediction skills) after teachers attended an education workshop.

This thesis not only contributes valuable insights into the integration of educational robotics within primary school settings but also highlights the importance of using teachers' insights to inform research direction and study design. By incorporating both teacher and child perspectives, a comprehensive understanding of the challenges and opportunities in integrating robotics education was achieved. Looking ahead, future research should aim to investigate longitudinal effects on learning outcomes (i.e., children's understanding of more complex programming languages), further examine the impact of teacher education programs on pupil learning and teacher beliefs and explore the benefits of educational robotics among children with diverse learning needs.

References

Aartun, I., Walseth, K., Standal, Ø. F., & Kirk, D. (2022). Pedagogies of embodiment in physical education–a literature review. *Sport, Education and Society*, *27*(1), 1-13.

Agatolio, F., Pivetti, M., Di Battista, S., Menegatti, E., & Moro, M. (2017). A training course in educational robotics for learning support teachers. In *Educational Robotics in the Makers Era 1* (pp. 43-57). Springer International Publishing.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, *19*(6), 716-723.

Anderson, S. E., & Maninger, R. M. (2007). Preservice teachers' abilities, beliefs, and intentions regarding technology integration. *Journal of educational computing research*, *37*(2), 151-172.

Anderson, M. L. (2003). Embodied cognition: A field guide. *Artificial Intelligence*, 149(1), 91–130.

Apperly, I. A., & Butterfill, S. A. (2009). Do humans have two systems to track beliefs and belief-like states?. *Psychological review*, *116*(4), 953.

Australian Curriculum Assessment and Reporting Authority. (2015). *Digital technologies: Sequence.* Retrieved from https://docs.acara.edu.au/resources/Digital_Technologies_-_Sequence_of_content.pdf

Axtell, C. M., Maitlis, S., & Yearta, S. K. (1997). Predicting immediate and longer-term transfer of training. *Personnel review*, *26*(3), 201-213.

Bakala, E., Gerosa, A., Hourcade, J. P., & Tejera, G. (2021). Preschool children, robots, and computational thinking: A systematic review. *International Journal of Child-Computer Interaction*, *29*, 100337.

Balanskat, A., & Engelhardt, K. (2015). Computing our future. *Computer programming and coding. Priorities, school curricula and initiatives across Europe.* European Schoolnet, Brussels.

Barefoot Computing. (2024). *What is computational thinking and why is it importance for your children to learn?* Retrieved from https://www.barefootcomputing.org/docs/default-source/at-home/quick--guide-to-computational-thinking.pdf

Barnett, S. M., & Ceci, S. J. (2002). When and where do we apply what we learn?: A taxonomy for far transfer. *Psychological bulletin*, *128*(4), 612.

Baron-Cohen, S., Leslie, A. M., & Frith, U. (1986). Mechanical, behavioural and intentional understanding of picture stories in autistic children. *British Journal of developmental psychology*, *4*(2), 113-125.

Barr, D. J. (2013). Random effects structure for testing interactions in linear mixed-effects models. *Frontiers in psychology*, *4*, 54057.

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology*, *38*(6), 20-23.

Barr, R. (2013). Memory constraints on infant learning from picture books, television, and touchscreens. *Child Development Perspectives*, *7*(4), 205-210.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?. *ACM Inroads*, *2*(1), 48-54.

Bates, D., Maechler, M., Bolker, B., Walker, S., Christensen, R. H. B., Singmann, H., ... & Bolker, M. B. (2015). Package 'lme4'. *convergence*, *12*(1), 2.

Beck, D. M., Schaefer, C., Pang, K., & Carlson, S. M. (2011). Executive function in preschool children: Test–retest reliability. *Journal of Cognition and Development, 12*(2), 169-193.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, *13*(1), 20-29.

Bennie, F., Corbett, C., & Palo, A. (2015). Building Bridges, Robots, and High Expectations. *Odyssey: New Directions in Deaf Education*, *16*, 14-19.

Bers, M. U. (2010). The TangibleK robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice*, *12*(2), n2.

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

Bers, M. U. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.

Bers, M. U., & Portsmore, M. (2005). Teaching partnerships: Early childhood and engineering students teaching math and science through robotics. *Journal of Science Education and Technology*, *14*, 59-73.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & education*, *72*, 145-157.

Bers, M. U., González-González, C., & Armas–Torres, M. B. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, *138*, 130-145.

Bers, M., Seddighin, S., & Sullivan, A. (2013). Ready for robotics: Bringing together the T and E of STEM in early childhood teacher education. *Journal of Technology and Teacher Education*, *21*(3), 355-377.

Blanca, M. J., Alarcón, R., Arnau, J., Bono, R., & Bendayan, R. (2017). Non-normal data: Is ANOVA still a valid option?. *Psicothema, 2017, vol. 29, num. 4, p. 552-557.*

Boud, D., Keogh, R., & Walker, D. (Eds.). (2013). *Reflection: Turning experience into learning*. Routledge.

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the 12th workshop on primary and secondary computing education* (pp. 65-72).

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).

Brown, A. L., & Murphy, M. D. (1975). Reconstruction of arbitrary versus logical sequences by preschool children. *Journal of Experimental Child Psychology*, *20*(2), 307-326.

Burke, L. A., & Hutchins, H. M. (2007). Training transfer: An integrative literature review. *Human resource development review*, *6*(3), 263-296.

Caeli, E. N., & Bundsgaard, J. (2020). Computational thinking in compulsory education: A survey study on initiatives and conceptions. *Educational technology research and development*, *68*(1), 551-573.

Carlson, S. M. (2005). Developmentally sensitive measures of executive function in preschool children. *Developmental neuropsychology*, *28*(2), 595-616.

Carlson, S. M., Schaefer, C. (2012). *Executive function scale for preschoolers–test manual*.

Carver, R. (1996). Theory for practice: A framework for thinking about experiential education. *Journal of Experiential Education*, *19*(1), 8-13.

Cassar, A., & Jang, E. (2010). Investigating the effects of a game-based approach in teaching word recognition and spelling to students with reading disabilities and attention deficits. *Australian Journal of Learning Difficulties*, 15(2), 193–211.

Castro, E., Cecchi, F., Salvini, P., Valente, M., Buselli, E., Menichetti, L., ... & Dario, P. (2018). Design and impact of a teacher training course, and attitude change concerning educational robotics. *International Journal of Social Robotics*, *10*, 669-685.

Chacón, C. T. (2005). Teachers' perceived efficacy among English as a foreign language teachers in middle schools in Venezuela. *Teaching and teacher education*, *21*(3), 257-272.

Chang, C. W., Lee, J. H., Chao, P. Y., Wang, C. Y., & Chen, G. D. (2010). Exploring the possibility of using humanoid robots as instructional tools for teaching a second language in primary school. *Journal of Educational Technology & Society*, *13*(2), 13-24.

Chang, Y. H., & Peterson, L. (2018). Pre-service teachers' perceptions of computational thinking. *Journal of Technology and Teacher Education*, *26*(3), 353-374.

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & education*, *109*, 162-175.

Cheung, A. C., & Slavin, R. E. (2013). The effectiveness of educational technology applications for enhancing mathematics achievement in K-12 classrooms: A meta-analysis. *Educational research review*, *9*, 88-113.

Ching, Y. H., & Hsu, Y. C. (2023). Educational robotics for developing computational thinking in young learners: A systematic review. *TechTrends*, 1-12.

Cho, K., Cho, M., & Jeon, J. (2017). Fly a drone safely: Evaluation of an embodied egocentric drone controller interface. *Interacting with computers*, *29*(3), 345-354.

Città, G., Gentile, M., Allegra, M., Arrigo, M., Conti, D., Ottaviano, S., Reale, F., & Sciortino, M. (2019). The effects of mental rotation on computational thinking. *Computers & Education*, *141*, 103613.

Cohen, J. (1988). *Statistical power analysis for the behavioural sciences* (2nd ed.). Hillsdale, NJ: Erlbaum.

Computer Science Teachers Association. (2020). *Standards for Computer Science Teachers.* Retrieved from https://csteachers.org/teacherstandards/

Cook, S. W., Friedman, H. S., Duggan, K. A., Cui, J., & Popescu, V. (2017). Hand gesture and mathematics learning: lessons from an Avatar. *Cognitive science*, *41*(2), 518-535.

Corradini, I., Lodi, M., & Nardelli, E. (2017, August). Conceptions and misconceptions about computational thinking among Italian primary school teachers. In *Proceedings of the 2017 ACM conference on international computing education research* (pp. 136-144).

Crawley, A. M., Anderson, D. R., Santomero, A., Wilder, A., Williams, M., Evans, M. K., & Bryant, J. (2002). *Do children learn how to watch television? The impact of extensive experience with Blue's Clues on preschool children's television viewing behavior. Journal of communication, 52*(2), 264-280.

Dagienė, V., & Sentance, S. (2016). It's computational thinking! Bebras tasks in the curriculum. In *Informatics in Schools: Improvement of Informatics Knowledge and Perception: 9th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2016, Münster, Germany, October 13-15, 2016, Proceedings 9* (pp. 28-39). Springer International Publishing.

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, *150*, 103832.

Dempster, F. N. (1992). The rise and fall of the inhibitory mechanism: Toward a unified theory of cognitive development and aging. *Developmental Review, 12*, 45–75.

Denning, P. J. (2009). The profession of IT Beyond computational thinking. *Communications of the ACM*, *52*(6), 28-30.

Department for Education. (2012, January 11). *"Harmful" ICT curriculum set to be dropped to make way for rigorous computer science* [Press Release]. Retrieved from https://www.gov.uk/government/news/harmful-ict-curriculum-set-to-be-dropped-to-make-way-for-rigorous-computer-science

Desimone, L. M. (2009). Improving impact studies of teachers' professional development: Toward better conceptualizations and measures. *Educational researcher*, *38*(3), 181-199.

Di Battista, S., Pivetti, M., Moro, M., & Menegatti, E. (2020). Teachers' opinions towards educational robotics for special needs students: An exploratory Italian study. *Robotics*, *9*(3), 72.

Diago, P. D., González-Calero, J. A., & Yáñez, D. F. (2022). Exploring the development of mental rotation and computational skills in elementary students through educational robotics. *International Journal of Child-Computer Interaction*, *32*, 100388.

Diamond, A., Kirkham, N., & Amso, D. (2002). Conditions under which young children can hold two rules in mind and inhibit a prepotent response. *Developmental psychology*, *38*(3), 352.

Digital News Asia. (2015). *IDA launches S$1.5m pilot to roll out tech toys for preschoolers.* Retrieved from https://www.digitalnewsasia.com/digital-economy/ida-launches-pilot-to-roll-out-tech-toys-for-preschoolers?page=2

Dralle-Moreano, J. (2021). *Educators' Perceptions of Benefits and Barriers of the Inclusion of Coding in K-8 Curriculum: A Qualitative Study*. St. John's University (New York).

Dumontheil, I., Apperly, I. A., & Blakemore, S. J. (2010). Online usage of theory of mind continues to develop in late adolescence. *Developmental science*, *13*(2), 331-338.

Eccles, J. S., & Wigfield, A. (2002). Motivational beliefs, values, and goals. *Annual review of psychology*, *53*(1), 109-132.

Elkin, M., Sullivan, A., & Bers, M. U. (2014). Implementing a robotics curriculum in an early childhood Montessori classroom. *Journal of Information Technology Education. Innovations in Practice*, *13*, 153.

Ensign, T. I. (2017). *Elementary educators' attitudes about the utility of educational robotics and their ability and intent to use it with students*. West Virginia University.

Ertmer, P. A. (1999). Addressing first-and second-order barriers to change: Strategies for technology integration. *Educational technology research and development*, *47*(4), 47-61.

Ertmer, P. A. (2005). Teacher pedagogical beliefs: The final frontier in our quest for technology integration? *Educational technology research and development*, *53*(4), 25-39.

Ertmer, P. A., Ottenbreit-Leftwich, A. T., Sadik, O., Sendurur, E., & Sendurur, P. (2012). Teacher beliefs and technology integration practices: A critical relationship. *Computers & education*, *59*(2), 423-435.

Ertmer, P. A., Paul, A., Molly, L., Eva, R., & Denise, W. (1999). Examining teachers' beliefs about the role of technology in the elementary classroom. *Journal of research on Computing in Education*, *32*(1), 54-72.

European Commission. (2022). *Reviewing Computational Thinking in Compulsory Education. State of play and practices from computing education.* Retrieved from: https://digital.dge.mec.pt/sites/default/files/documents/2022/174-36d642bd2aa5f00c90ad53e3bccfa907.pdf

European Schoolnet. (2015). *Computing our future. Computer programming and coding. Priorities, school curricula and initiatives across Europe.* http://www.eun.org/documents/411753/817341/Computing+our+future_final_2015.pdf

Faul, F., Erdfelder, E., Lang, A. G., & Buchner, A. (2007). G* Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior research methods*, *39*(2), 175-191.

Fessakis, G., & Prantsoudi, S. (2019). Computer Science Teachers' Perceptions, Beliefs and Attitudes on Computational Thinking in Greece. *Informatics in Education*, *18*(2), 227-258.

Fisch, S. M., Kirkorian, H. L., & Anderson, D. R. (2005). Transfer of learning in informal education. *Transfer of learning from a modern multidisciplinary perspective*, 371-393.

Flavell, J. H. (1977). *The development of knowledge about visual perception. In Nebraska symposium on motivation.* University of Nebraska Press.

Flavell, J. H., Flavell, E. F., Green, F. L., and Wilcox, S. A. (1980). Young children's knowledge about visual perception: effect of observer's distance from target on perceptual clarity of target. *Dev. Psychol. 16*, 10–12. doi: 10.1037/0012-1649.16.1.10

Foster, A., Urquhart, C., & Turner, J. (2008). Validating coding for a theoretical model of information behaviour. *Information Research*, *13*(4), 13-4.

Fox, F. E., Morris, M., & Rumsey, N. (2007). Doing synchronous online focus groups with young people: Methodological reflections. *Qualitative health research*, *17*(4), 539-547.

Frick, A., & Baumeler, D. (2017). The relation between spatial perspective taking and inhibitory control in 6-year-old children. *Psychological research*, *81*(4), 730-739.

Frick, A., & Pichelmann, S. (2023). Measuring spatial abilities in children: a comparison of mental-rotation and perspective-taking tasks. *Journal of Intelligence*, *11*(8), 165.

Fuchs, L. S., Fuchs, D., & Phillips, N. (1994). The relation between teachers' beliefs about the importance of good student work habits, teacher planning, and student achievement. *The Elementary School Journal*, *94*(3), 331-345.

Fugate, J. M., Macrine, S. L., & Cipriano, C. (2019). The role of embodied cognition for transforming learning. *International Journal of School & Educational Psychology*, *7*(4), 274-288.

Futschek, G., & Moschitz, J. (2011). Learning algorithmic thinking with tangible objects eases transition to computer programming. In *Informatics in Schools. Contributing to 21st Century Education: 5th International Conference on Informatics in Schools: Situation, Evolution and Perspectives, ISSEP 2011, Bratislava, Slovakia, October 26-29, 2011. Proceedings 5* (pp. 155-164). Springer Berlin Heidelberg.

Garvin, M., Killen, H., Plane, J., & Weintrop, D. (2019, February). Primary school teachers' conceptions of computational thinking. In *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 899-905).

Garvis, S., & Keane, T. (2023). A Literature Review of Educational Robotics and Early Childhood Education. *Technological Innovations in Education: Applications in Education and Teaching*, 71-83.

Gavrilas, L., Kotsis, K. T., & Papanikolaou, M. S. (2024). Assessing teacher readiness for educational robotics integration in primary and preschool education. *Education 3-13*, 1-17.

Geist, E. (2014). Using tablet computers with toddlers and young preschoolers. *YC Young children*, *69*(1), 58.

Georgiou, Y., & Ioannou, A. (2019). Embodied learning in a digital world: A systematic review of empirical research in K-12 education. *Learning in a digital world: Perspective on interactive technologies for formal and informal education*, 155-177.

Gerson, S. A., Morey, R. D., & van Schaik, J. E. (2022). Coding in the cot? Factors influencing 0–17s' experiences with technology and coding in the United Kingdom. *Computers & Education*, *178*, 104400.

Gilligan, K. A., Flouri, E., & Farran, E. K. (2017). The contribution of spatial ability to mathematics achievement in middle childhood. *Journal of experimental child psychology*, *163*, 107-125.

Gilligan, K. A., Hodgkiss, A., Thomas, M. S., & Farran, E. K. (2019). The developmental relations between spatial cognition and mathematics in primary school children. *Developmental science*, *22*(4), e12786.

Girvan, C., Tangney, B., & Savage, T. (2013). SLurtles: supporting constructionist learning in second life. *Computers & Education*, *61*, 115-132.

Glaser, B. G. (1965). The constant comparative method of qualitative analysis. *Social problems*, *12*(4), 436-445.

Goldstein, T. R. (2018). Developing a Dramatic Pretend Play Game Intervention. *American Journal of Play*, *10*(3), 290-308.

González-González, C. S., Herrera-González, E., Moreno-Ruiz, L., Reyes-Alonso, N., Hernández-Morales, S., Guzmán-Franco, M. D., & Infante-Moro, A. (2019). Computational thinking and down syndrome: An exploratory study using the KIBO robot. *Informatics 6*(2), 25.

Greeno, J. G. (1998). The situativity of knowing, learning, and research. *American psychologist*, *53*(1), 5.

Greifenstein, L., Graßl, I., & Fraser, G. (2021, November). Challenging but full of opportunities: Teachers' perspectives on programming in primary schools. In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research* (pp. 1-10).

Grossman, P. (2009). Research on pedagogical approaches in teacher education. In *Studying teacher education* (pp. 437-488). Routledge.

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. *Computer science education: Perspectives on teaching and learning in school*, *19*(1), 19-38.

Grund, S., Lüdtke, O., & Robitzsch, A. (2018). Multiple imputation of missing data for multilevel models: Simulations and recommendations. *Organizational Research Methods*, *21*(1), 111-149.

Guskey, T. R. (1981). Measurement of the responsibility teachers assume for academic successes and failures in the classroom. *Journal of teacher Education*, *32*(3), 44-51.

Gustafsson, J. E. (2003). What do we know about effects of school resources on educational results?. *Swedish economic policy review*, *10*(2), 77-77.

Gutiérrez-Núñez, S. E., Cordero-Hidalgo, A., & Tarango, J. (2022). Implications of computational thinking knowledge transfer for developing educational interventions. *Contemporary Educational Technology*, *14*(3), ep367.

Hajian, S. (2019). Transfer of learning and teaching: A review of transfer theories and effective instructional practices. *IAFOR Journal of education*, *7*(1), 93-111.

Harel, I. E., & Papert, S. E. (1991). *Constructionism*. Ablex Publishing.

Harrington, R. C., Cartwright-Hatton, S., & Stein, A. (2002). Annotation: randomised trials. *Journal of Child Psychology and Psychiatry*, *43*(6), 695-704.

Harris, D. N., & Sass, T. R. (2011). Teacher training, teacher quality and student achievement. *Journal of public economics*, *95*(7-8), 798-812.

Hattie, J. (2003). Teachers make a difference: What is the research evidence? *Paper presented at the Australian Council for Educational Research Annual Conference on Building Teacher Quality*. Melbourne, Australia.

Hegna, H. M., & Ørbæk, T. (2021). Traces of embodied teaching and learning: a review of empirical studies in higher education. *Teaching in Higher Education*, 1-22.

Hew, K. F., & Brush, T. (2007). Integrating technology into K-12 teaching and learning: Current knowledge gaps and recommendations for future research. *Educational technology research and development*, *55*, 223-252.

Hintzman, D. L. (1976). *Repetition and memory.* In Psychology of learning and motivation (Vol. 10, pp. 47- 91). Academic Press.

Hodgkiss, A., Gilligan, K. A., Tolmie, A. K., Thomas, M. S., & Farran, E. K. (2018). Spatial cognition and science achievement: The contribution of intrinsic and extrinsic spatial skills from 7 to 11 years. *British Journal of Educational Psychology*, *88*(4), 675-697.

Hofman, R. H., Hofman, W. A., & Guldemond, H. (2002). School governance, culture, and student achievement. *International Journal of leadership in Education*, *5*(3), 249-272.

Hoppe, H. U., & Werneburg, S. (2019). Computational thinking—More than a variant of scientific inquiry. *Computational thinking education*, 13-30.

Horn, M. S., Crouser, R. J., & Bers, M. U. (2012). Tangible interaction and learning: the case for a hybrid approach. *Personal and Ubiquitous Computing*, *16*, 379-389.

House of Commons. (2016). *Digital Skills Crisis.* Retrieved from https://publications.parliament.uk/pa/cm201617/cmselect/cmsctech/270/270.pdf

Hox, J., Moerbeek, M., & Van de Schoot, R. (2017). *Multilevel analysis: Techniques and applications.* Routledge.

Hsieh, H. F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative health research*, *15*(9), 1277-1288.

Hussain, S., Lindh, J., & Shukur, G. (2006). The effect of LEGO training on pupils' school performance in mathematics, problem solving ability and attitude: Swedish data. *Journal of Educational Technology & Society*, *9*(3), 182-194.

Huttenlocher, J., & Presson, C. C. (1973). Mental rotation and the perspective problem. *Cognitive Psychology, 4*(2), 277-299.

Huttenlocher, J., & Presson, C. C. (1973). Mental rotation and the perspective problem. *Cognitive Psychology*, *4*(2), 277-299.

Hwb. (2018). *Digital Competence Framework guidance*. Retrieved from https://hwb.gov.wales/api/storage/337437b8-cfe3-4305-ae32-f47ad82f3e76/digital-competence-framework-guidance-2018.pdf

Hwb. (2024a). *Curriculum for Wales.* Retrieved from https://hwb.gov.wales/curriculum-for-wales/

Hwb. (2024b). *Descriptions of learning. Science and Technology.* Retrieved from https://hwb.gov.wales/curriculum-for-wales/science-and-technology/descriptions-of-learning/

Hwb. (2024c). *Introduction to curriculum for Wales guidance.* Retrieved from https://hwb.gov.wales/curriculum-for-wales/introduction-to-curriculum-for-wales-guidance/

Jaipal-Jamani, K., & Angeli, C. (2017). Effect of robotics on elementary preservice teachers' self-efficacy, science learning, and computational thinking. *Journal of science education and technology*, *26*, 175-192.

Jara, I., Hepp, P., & Rodriguez, J. (2018). Policies and practices for teaching computer science in Latin America. *Microsoft.*

Jerrim, J., Sims, S., & Oliver, M. (2023). Teacher self-efficacy and pupil achievement: much ado about nothing? International evidence from TIMSS. *Teachers and Teaching*, *29*(2), 220-240.

Jones, S., & Burnett, G. (2008). Spatial ability and learning to program. *Human Technology: An Interdisciplinary Journal on Humans in ICT Environments*.

Kallia, M., & Cutts, Q. (2023). Conceptual development in early-years computing education: a grounded cognition and action based conceptual framework. *Computer Science Education*, *33*(4), 485-511.

Karpicke, J. D., & Roediger, H. L. (2008). The critical importance of retrieval for learning. *science, 319*(5865), 966-968.

Kay, J. S., & Moss, J. G. (2012, October). Using robots to teach programming to K-12 teachers. In *2012 Frontiers in Education Conference Proceedings* (pp. 1-6). IEEE.

Kaya, E., Yesilyurt, E., Newley, A., & Deniz, H. (2019). Examining the impact of a computational thinking intervention on pre-service elementary science teachers' computational thinking teaching efficacy beliefs, interest and confidence. *Journal of Computers in Mathematics and Science Teaching*, *38*(4), 385-392.

Kazakoff, E. R., Sullivan, A., & Bers, M. U. (2013). The effect of a classroom-based intensive robotics and programming workshop on sequencing ability in early childhood. *Early Childhood Education Journal*, *41*, 245-255.

Kenny, A. J. (2005). Interaction in cyberspace: An online focus group. *Journal of advanced nursing*, *49*(4), 414-422.

Khanlari, A. (2014). *Teachers' perceptions of using robotics in primary/elementary schools in Newfoundland and Labrador* (Doctoral dissertation, Memorial University of Newfoundland).

Khanlari, A. (2016). Teachers’ perceptions of the benefits and the challenges of integrating educational robots into primary/elementary curricula. *European Journal of Engineering Education*, *41*(3), 320-330.

Khanlari, A., & Mansourkiaie, F. (2015, July). Using robotics for STEM education in primary/elementary schools: Teachers' perceptions. In *2015 10th International Conference on Computer Science & Education (ICCSE)* (pp. 3-7). IEEE.

Kim, C., Kim, D., Yuan, J., Hill, R. B., Doshi, P., & Thai, C. N. (2015). Robotics to promote elementary education pre-service teachers' STEM engagement, learning, and teaching. *Computers & Education*, *91*, 14-31.

Kim, K. R., & Seo, E. H. (2018). The relationship between teacher efficacy and students' academic achievement: A meta-analysis. *Social Behavior and Personality: an international journal*, *46*(4), 529-540.

Kipster, E. (2018). *“What digital skills do adults need to succeed in the workplace now and in the next 10 years?”* Digital Skills and Inclusion Research Working Group Evidence Brief. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/807831/What_digital_skills_do_adults_need_to_succeed_in_the_workplace_now_and_in_the_next_10_years_.pdf

Kıran, I., Macun, B., Argın, Y., & Ulutaş, İ. (2021). Montessori method in early childhood education: A systematic review. *Cukurova University Faculty of Education Journal*, *50*(2), 1154-1183.

Klassen, R. M., & Tze, V. M. (2014). Teachers’ self-efficacy, personality, and teaching effectiveness: A meta-analysis. *Educational research review*, *12*, 59-76.

Koepsell, T. D., Wagner, E. H., Cheadle, A. C., Patrick, D. L., Martin, D. C., Diehr, P. H., ... & Dey, L. J. (1992). Selected methodological issues in evaluating community-based health promotion and disease prevention programs. *Annual review of public health*, *13*(1), 31-57.

Konen, J., & Horton, R. (2000). Beneficial science teacher training. *Journal of Extension*, *38*(2), 28-34.

Kosmas, P., Ioannou, A., & Zaphiris, P. (2019). Implementing embodied learning in the classroom: Effects on children's memory and language skills. *Educational Media International*, *56*(1), 59-74.

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2017). lmerTest package: tests in linear mixed effects models. *Journal of statistical software*, *82*(13).

Kwon, K., Jeon, M., Zhou, C., Kim, K., & Brush, T. A. (2022). Embodied learning for computational thinking in early primary education. *Journal of Research on Technology in Education*, 1-21.

Laker, A., Laker, J. C., & Lea, S. (2008). Sources of support for pre-service teachers during school experience. *Mentoring & Tutoring: Partnership in Learning*, *16*(2), 125-140.

Landis, J. R., & Koch, G. G. (1977). An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, 363-374.

Larke, L. R. (2019). Agentic neglect: Teachers as gatekeepers of England's national computing curriculum. *British Journal of Educational Technology*, *50*(3), 1137-1150.

Lauermann, F., & ten Hagen, I. (2021). Do teachers' perceived teaching competence and self-efficacy affect students' academic outcomes? A closer look at student-reported classroom processes and outcomes. *Educational psychologist*, *56*(4), 265-282.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32-37.

Lenth, R. (2021). emmeans: Estimated Marginal Means, aka Least-Squares Means. *R package version 1.5.4*. Available at https://CRAN.R-project.org/ package=emmeans.

Leonard, J., Buss, A., Gamboa, R., Mitchell, M., Fashola, O. S., Hubert, T., & Almughyirah, S. (2016). Using robotics and game design to enhance children's self-efficacy, STEM attitudes, and computational thinking skills. *Journal of Science Education and Technology*, *25*, 860-876.

Liao, Y. C., Ottenbreit-Leftwich, A., Karlin, M., Glazewski, K., & Brush, T. (2017). Supporting change in teacher practice: Examining shifts of teachers' professional development preferences and needs for technology integration. *Contemporary Issues in Technology and Teacher Education*, *17*(4), 522-548.

Lindh, J., & Holgersson, T. (2007). Does lego training stimulate pupils' ability to solve logical problems?. *Computers & education*, *49*(4), 1097-1111.

Linebarger, D. L., & Piotrowski, J. T. (2009). TV as storyteller: How exposure to television narratives impacts at-risk preschoolers' story knowledge and narrative skills. *British journal of developmental psychology*, *27*(1), 47-69.

Linn, M. C., & Petersen, A. C. (1985). Emergence and characterization of sex differences in spatial ability: A meta-analysis. *Child development*, 1479-1498.

Lu, C. M., Kang, S., Huang, S. C., & Black, J. B. (2011, June). Building student understanding and interest in science through embodied experiences with LEGO robotics. In *EdMedia+ Innovate Learning* (pp. 2225-2232). Association for the Advancement of Computing in Education (AACE).

Lytle, L. A., Gerlach, S., & Weinstein, A. B. (2001). Conducting nutrition education research in junior high schools: approaches and challenges. *Journal of Nutrition Education*, *33*(1), 49-54.

Maier, M. F., Greenfield, D. B., & Bulotsky-Shearer, R. J. (2013). Development and validation of a preschool teachers' attitudes and beliefs toward science teaching questionnaire. *Early Childhood Research Quarterly*, *28*(2), 366-378.

Mangina, E., Psyrra, G., Screpanti, L., & Scaradozzi, D. (2023). Robotics in the Context of Primary and Pre-School Education: A Scoping Review. *IEEE Transactions on Learning Technologies*.

Mantzicopoulos, P., Patrick, H., & Samarapungavan, A. (2008). Young children's motivational beliefs about learning science. *Early Childhood Research Quarterly*, *23*(3), 378-394.

Mares, M. L. (2006). Repetition increases children's comprehension of television content—up to a point. *Communication Monographs, 73*(2), 216-241.

Marshall, C. (2017). Montessori education: a review of the evidence base. *npj Science of Learning*, *2*(1), 11.

Masangkay, Z. S., Mccluskey, K. A., Mcintyre, C. W., Sims-Knight, J., Vaughn, B. E., and Flavell, J. H. (1974). The early development of inferences about the visual percepts of others. *Child Dev. 45*, 357–366. doi: 10.2307/1127956

Mason, S. L., & Rich, P. J. (2019). Preparing elementary school teachers to teach computing, coding, and computational thinking. *Contemporary Issues in Technology and Teacher Education*, *19*(4), 790-824.

Master, A., Cheryan, S., Moscatelli, A., & Meltzoff, A. N. (2017). Programming experience promotes higher STEM motivation among first-grade girls. *Journal of experimental child psychology*, *160*, 92-106.

McClelland, M. M., Cameron, C. E., Duncan, R., Bowles, R. P., Acock, A. C., Miao, A., & Pratt, M. E. (2014). Predictors of early growth in academic achievement: The head-toes-knees-shoulders task. *Frontiers in psychology*, *5*, 599.

Merriam, S. B. (1998). *Qualitative Research and Case Study Applications in Education. Revised and Expanded from" Case Study Research in Education."*. Jossey-Bass Publishers, 350 Sansome St, San Francisco, CA 94104.

Midgley, C., Feldlaufer, H., & Eccles, J. S. (1989). Change in teacher efficacy and student self- and task-related beliefs in mathematics during the transition to junior high school. *Journal of educational Psychology*, *81*(2), 247.

Migliaresi, G. (2016, September 6). *"A very special Montessori moment…" Cubetto passes the Montessori test!* Retrieved from https://www.primotoys.com/cubetto-featured-montessori-international/

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*, 139-158.

Mok, M. M. C., & Moore, P. J. (2019). Teachers & self-efficacy. *Educational Psychology*, *39*(1), 1-3.

Moll, H., & Tomasello, M. (2006). Level 1 perspective-taking at 24 months of age. *British Journal of Developmental Psychology*, *24*(3), 603-613.

Montessori, M. (1967). *The discovery of the child.* Toronto, Canada: Fides Publishing Inc.

Morgado, L., Cruz, M., & Kahn, K. (2006). Radia Perlman—A pioneer of young children computer programming. *Current developments in technology-assisted education. Proceedings of m-ICTE,* 1903-1908.

Moser, A., Zimmermann, L., Dickerson, K., Grenell, A., Barr, R., & Gerhardstein, P. (2015). They can interact, but can they learn? Toddlers' transfer learning from touchscreens and television. *Journal of experimental child psychology, 137*, 137-155.

Muñoz-Repiso, AGV, & González, YAC (2019). Robotics to develop computational thinking in Early Childhood Education. *Comunicar: Ibero-American scientific journal of communication and education* , (59), 63-72.

Nam, K. W., Kim, H. J., & Lee, S. (2019). Connecting plans to action: The effects of a card-coded robotics curriculum and activities on Korean kindergartners. *The Asia-Pacific Education Researcher*, *28*, 387-397.

Namey, E., Guest, G., O'Regan, A., Godwin, C. L., Taylor, J., & Martinez, A. (2020). How does mode of qualitative data collection affect data and cost? Findings from a quasi-experimental study. *Field methods*, *32*(1), 58-74.

Neuendorf, K. A. (2010). Reliability for content analysis. In *Media messages and public health* (pp. 85-105). Routledge.

Newcombe, N. S., & Shipley, T. F. (2014). Thinking about spatial thinking: New typology, new assessments. In *Studying visual and spatial reasoning for design creativity* (pp. 179-192). Dordrecht: Springer Netherlands.

Newhouse, D. C. P. (2001). A follow-up study of students using portable computers at a secondary school. *British journal of educational technology*, *32*(2), 209-219.

Noonan, R. (2017). STEM Jobs: 2017 Update. ESA Issue Brief# 02-17. *US Department of Commerce*.

O'Connor, S., McGilloway, S., Hickey, G., & Barwick, M. (2021). Disseminating early years research: an illustrative case study. *Journal of Children's Services, 16*(1), 56-73.

Ohashi, Y., Kumeno, F., Yamachi, H., & Tsujimura, Y. (2018, December). Readiness of Japanese elementary school teachers to begin computer-programming education. In *2018 IEEE*

International Conference on Teaching, Assessment, and Learning for Engineering (TALE) (pp. 807-810). IEEE.

Oppermann, E., Brunner, M., & Anders, Y. (2019). The interplay between preschool teachers' science self-efficacy beliefs, their teaching practices, and girls' and boys' early science motivation. *Learning and Individual Differences*, *70*, 86-99.

Owen, K. L., Watkins, R. C., & Hughes, J. C. (2022). From evidence-informed to evidence-based: An evidence building framework for education. *Review of Education, 10*(1), e3342.

Parkinson, J., & Cutts, Q. (2018, August). Investigating the relationship between spatial skills and computer science. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (pp. 106-114).

Perkins, D. N., & Salomon, G. (1992). Transfer of learning. *International encyclopedia of education, 2*, 6452-6457.

Perlman, R. (1974). TORTIS – Toddler's Own Recursive Turtle Interpreter System. *MIT AI Memo 311, Logo Memo 9.* Cambridge, MA, USA: Massachusetts Institute of Technology.

Piaget, J., & Cook, M. (1952). *The origins of intelligence in children* (Vol. 8, No. 5, pp. 18-1952). New York: International Universities Press.

Piaget, J., & Inhelder, B. (1956). *The child's conception of space*. London: Routledge & Kegan Paul

Pillow, B. H., and Flavell, J. H. (1986). Young children's knowledge about visual perception: projective size and shape. *Child Dev. 57*, 125–135. doi: 10.2307/1130644

Ponitz, C. E. C., McClelland, M. M., Jewkes, A. M., Connor, C. M., Farris, C. L., & Morrison, F. J. (2008). Touch your toes! Developing a direct measure of behavioral regulation in early childhood. *Early Childhood Research Quarterly*, *23*(2), 141-158.

Primo Toys. (2024). *Teaching Guide: Introducing Cubetto in the classroom.* Retrieved from: https://cspmoraviabrno-my.sharepoint.com/personal/giorgia_migliaresi_moravia-consulting_com/Documents/Primo%20Toys%20Website%20-%20files%20and%20materials/Teacher%20Guide/Teacher%20Guide%20(improved).pdf?CT=1716302680156&OR=ItemsView

Pugnali, A., Sullivan, A., & Bers, M. U. (2017). The impact of user interface on young children's computational thinking. *Journal of Information Technology Education. Innovations in Practice*, *16*, 171.

PwC. (2021). *The Potential Impact of Artificial Intelligence on UK Employment and the Demand for Skills.* Retrieved from https://assets.publishing.service.gov.uk/media/615d9a1ad3bf7f55fa92694a/impact-of-ai-on-jobs.pdf

Ray, B. B., Rogers, R. R., & Hocutt, M. M. (2020). Perceptions of non-STEM discipline teachers on coding as a teaching and learning tool: What are the possibilities?. *Journal of Digital Learning in Teacher Education*, *36*(1), 19-31.

Relkin, E., de Ruiter, L. E., & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & education*, *169*, 104222.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, K., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60-67.

Rich, K. M., Yadav, A., & Schwarz, C. V. (2019). Computational thinking, mathematics, and science: Elementary teachers' perspectives on integration. *Journal of Technology and Teacher Education*, *27*(2), 165-205.

Rigal, R. (1994). Right-left orientation: Development of correct use of right and left terms. *Perceptual and motor skills*, *79*(3), 1259-1278.

Rogers, T., Andler, C., O'Brien, B., & van Schaik, S. (2019). Self-reported emotions in simulation-based learning: active participants vs. observers. *Simulation in Healthcare*, *14*(3), 140-145.

Roussou, E., & Rangoussi, M. (2020). On the use of robotics for the development of computational thinking in kindergarten: Educational intervention and evaluation. In *Robotics in Education: Current Research and Innovations 10* (pp. 31-44). Springer International Publishing.

Salomon, G., & Perkins, D. N. (1989). Rocky roads to transfer: Rethinking mechanism of a neglected phenomenon. *Educational psychologist*, *24*(2), 113-142.

Salway, R., Walker, R., Sansum, K., House, D., Emm-Collison, L., Reid, T., Breheny, K., Williams, J. G., de Vocht, F., Hollingworth, W., Foster, C., & Jago, R. (2023). Screen-viewing behaviours of children before and after the 2020–21 COVID-19 lockdowns in the UK: a mixed methods study. *BMC Public Health*, *23*(1), 116.

Sands, P., Yadav, A., & Good, J. (2018). Computational thinking in K-12: In-service teacher perceptions of computational thinking. *Computational thinking in the STEM disciplines: Foundations and research highlights*, 151-164.

Santer, J., Griffiths, C., & Goodall, D. (2007). *Free play in early childhood: A literature review. National Children's Bureau.* [PDF file] Retrieved from: http://www.playengland.net/wpcontent/uploads/2015/09/free-play-in-early-childhood.pdf

Sapounidis, T., Tselegkaridis, S., & Stamovlasis, D. (2022). Educational robotics and STEM in primary education: a review and a meta-analysis. *Journal of Research on Technology in Education*, 1-15.

Schina, D., Esteve-González, V., & Usart, M. (2021). An overview of teacher training programs in educational robotics: characteristics, best practices and recommendations. *Education and Information Technologies*, *26*(3), 2831-2852.

Schneider, B. (2017). *Tangibles and tangible learning*. In Peppler, K. (Ed.). (2017). The SAGE encyclopedia of out-of-school learning. Sage Publications.

Schunk, D. H. (2012). *Learning theories an educational perspective*. Pearson Education, Inc.

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher’s perspective. *Education and information technologies*, *22*, 469-495.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational research review*, *22*, 142-158.

Simpkins, S. D., Davis-Kean, P. E., & Eccles, J. S. (2006). Math and science motivation: A longitudinal examination of the links between choices and beliefs. *Developmental psychology*, *42*(1), 70.

Sisman, B., Kucuk, S., & Yaman, Y. (2021). The effects of robotics training on children’s spatial ability and attitude toward STEM. *International Journal of Social Robotics*, *13*(2), 379-389.

Siu, K. W. M., Lam, M. S., & Renck, M. (2003). Technology education in Hong Kong: International implications for implementing the “eight Cs” in the early childhood curriculum. *Early Childhood Education Journal*, *31*(2).

Slangen, L., van Keulen, H., & Gravemeijer, K. (2011). What pupils can learn from working with robotic direct manipulation environments. *International Journal of Technology and Design Education*, *21*, 449-469.

Smith, M. (2016). Computer science for all. *The White House blog.* Retrieved from https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all

Sodian, B., Thoermer, C., & Metz, U. (2007). Now I see it but you don't: 14-month-olds can represent another person's visual perspective. *Developmental science*, *10*(2), 199-204.

Stephens, P., Egil tønnessen, F., & Kyriacou, C. (2004). Teacher training and teacher education in England and Norway: a comparative study of policy goals. *Comparative education*, *40*(1), 109-130.

Stipek, D. J., Givvin, K. B., Salmon, J. M., & MacGyvers, V. L. (2001). Teachers’ beliefs and practices related to mathematics instruction. *Teaching and teacher education*, *17*(2), 213-226.

Strauss, A., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory* (2nd ed.). Sage Publications, Inc.

Strawhacker, A., & Bers, M. U. (2015). “I want my robot to look for food”: Comparing Kindergartner’s programming comprehension using tangible, graphic, and hybrid user interfaces. *International Journal of Technology and Design Education*, *25*(3), 293-319.

Strawhacker, A., & Bers, M. U. (2019). What they learn when they learn coding: Investigating cognitive domains and computer programming knowledge in young children. *Educational technology research and development*, *67*, 541-575.

Strawhacker, A., Lee, M., & Bers, M. U. (2018). Teaching tools, teachers’ rules: exploring the impact of teaching styles on young children’s programming knowledge in ScratchJr. *International Journal of Technology and Design Education*, *28*, 347-376.’

Strawhacker, A., Lee, M., Caine, C., & Bers, M. (2015). ScratchJr Demo: A Coding Language for Kindergarten. *Proceedings of the 14th International Conference on Interaction Design and Children, 414–417.* https://doi.org/10.1145/2771839.2771867

Sullivan, A. A., Bers, M. U., & Mihm, C. (2017). Imagining, playing, and coding with KIBO: using robotics to foster computational thinking in young children. *Siu-cheung KONG The Education University of Hong Kong, Hong Kong*, *110*.

Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education*, *26*, 3-20.

Sullivan, A., & Bers, M. U. (2019). Investigating the use of robotics to increase girls' interest in engineering during early elementary school. *International Journal of Technology and Design Education*, *29*, 1033-1051.

Sullivan, A., Elkin, M., & Bers, M. U. (2015, June). KIBO robot demo: engaging young children in programming and engineering. In *Proceedings of the 14th international conference on interaction design and children* (pp. 418-421).

Sullivan, F. R., & Moriarty, M. A. (2009). Robotics and discovery learning: Pedagogical beliefs, teacher practice, and technology integration. *Journal of Technology and Teacher Education*, *17*(1), 109-142.

Tang, K. Y., Chou, T. L., & Tsai, C. C. (2020). A content analysis of computational thinking research: An international publication trends and research typology. *The Asia-Pacific Education Researcher*, *29*(1), 9-19.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. Computers & Education, 148, 103798. https://doi.org/10.1016/j.compedu.2019.103798

Tschannen-Moran, M., & Hoy, A. W. (2001). Teacher efficacy: Capturing an elusive construct. *Teaching and teacher education*, *17*(7), 783-805.

Tversky, B., & Hard, B. M. (2009). Embodied and disembodied cognition: Spatial perspective-taking. *Cognition*, *110*(1), 124-129.

Tzagaraki, E., Papadakis, S., & Kalogiannakis, M. (2022). Teachers' attitudes on the use of educational robotics in primary school. In *STEM, Robotics, Mobile Apps in Early Childhood and Primary Education: Technology to Promote Teaching and Learning* (pp. 257-283). Singapore: Springer Nature Singapore.

UK Parliament. (2023, September). *Education Committee's new inquiry into effects of screen time on education and wellbeing.* https://committees.parliament.uk/work/7912/screen-time-impacts-on-education-and-wellbeing/

Uttal, D. H., Miller, D. I., & Newcombe, N. S. (2013). Exploring and enhancing spatial thinking: Links to achievement in science, technology, engineering, and mathematics?. *Current Directions in Psychological Science*, *22*(5), 367-373.

Uzunboylu, H., Kınık, E., & Kanbul, S. (2017). An analysis of countries which have integrated coding into their curricula and the content analysis of academic studies on coding training in Turkey. *Tem Journal*, *6*(4), 783.

van Aalderen-Smeets, S., & Walma van der Molen, J. (2013). Measuring primary teachers' attitudes toward teaching science: Development of the dimensions of attitude toward science (DAS) instrument. *International journal of science education*, *35*(4), 577-600.

Veenman, M. V., & Spaans, M. A. (2005). Relation between intellectual and metacognitive skills: Age and task differences. *Learning and individual differences*, *15*(2), 159-176.

von Wangenheim, A., von Wangenheim, C. G., Pacheco, F. S., Hauck, J. C., & Ferreira, M. N. F. (2017). Motivating Teachers to Teach Computing in Middle School â€“A Case Study of a Physical Computing Taster Workshop for K-12 Teachers. *International Journal of Computer Science Education in Schools*, *1*(4), 35-49.

Wang, K., Sang, G. Y., Huang, L. Z., Li, S. H., & Guo, J. W. (2023). The effectiveness of educational robots in improving learning outcomes: A meta-analysis. *Sustainability*, *15*(5), 4637.

Welsh Government. (2017). *Cracking the code: A plan to expand code clubs in every part of Wales.* Retrieved from: https://www.gov.wales/sites/default/files/publications/2018-11/cracking-the-code-a-plan-to-expand-code-clubs-in-every-part-of-wales.pdf

Welsh Government (2021). *The Additional Learning Needs Code for Wales 2021.* Retrieved from https://www.gov.wales/sites/default/files/publications/2024-04/220622-the-additional-learning-needs-code-for-wales-2021%20%282%29.pdf

Welsh Government. (2022). *School Workforce Census results: as at November 2021.* Retrieved from: https://www.gov.wales/school-workforce-census-results-november-2021-html#99631

Welsh Government. (2024a). *Hwb Resources.* Retrieved from https://hwb.gov.wales/resources

Welsh Government (2024b). *Schools' census results: January 2024.* https://www.gov.wales/sites/default/files/pdf-versions/2024/8/4/1722499003/schools-census-results-january-2024.pdf

Welsh, M. C., Pennington, B. F., & Groisser, D. B. (1991). A normative-developmental study of executive function: A window on prefrontal function in children. *Developmental Neuropsychology, 7,* 131–149

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D. A., François, R., ... & Yutani, H. (2019). Welcome to the Tidyverse. *Journal of open source software*, *4*(43), 1686.

Willingham, D. T., & Daniel, D. B. (2021). Making Education Research Relevant: How researchers can give teachers more choices. *Education Next*, *21*(2), 28-34.

Wing, J. (2011). Research notebook: Computational thinking—What and why. *The link magazine*, *6*, 20-23.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Woolfolk, A. E., Rosoff, B., & Hoy, W. K. (1990). Teachers' sense of efficacy and their beliefs about managing students. *Teaching and teacher Education*, *6*(2), 137-148.

WorldSkills UK. (2021). *Disconnected: Exploring the Digital Skills Gap.* https://www.worldskillsuk.org/wp-content/uploads/2021/03/Disconnected-Report-final.pdf

Xia, L., & Zhong, B. (2018). A systematic review on teaching and learning robotics content knowledge in K-12. *Computers & Education*, *127*, 267-282.

Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, *60*(4), 55-62.

Yadollahi, E., Couto, M., Johal, W., Dillenbourg, P., & Paiva, A. (2020). Exploring the role of perspective taking in educational child-robot interaction. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21* (pp. 346-351). Springer International Publishing.

Zack, E., Barr, R., Gerhardstein, P., Dickerson, K., & Meltzoff, A. N. (2009). Infant imitation from television using novel touch screen technology. *British Journal of Developmental Psychology*, *27*(1), 13-26.

Zack, E., Gerhardstein, P., Meltzoff, A. N., & Barr, R. (2013). 15-month-olds' transfer of learning between touch screen and real-world displays: language cues and cognitive loads. *Scandinavian Journal of Psychology, 54*(1), 20-25.

Zelazo, P. D. (2006). The Dimensional Change Card Sort (DCCS): A method of assessing executive function in children. *Nature protocols*, *1*(1), 297-301.

Zelazo, P. D., Carter, A., Reznick, J. S., & Frye, D. (1997). Early development of executive function: A problem-solving framework. *Review of General Psychology, 1*, 1–29.

Zviel-Girshin, R., Luria, A., & Shaham, C. (2020). Robotics as a tool to enhance technological thinking in early childhood. *Journal of Science Education and Technology*, *29*, 294-302.

Appendix A

Focus Group Recruitment Advert (Chapter 2)

We are looking for a group of primary school teachers to take part in our focus group research! To be eligible for this study, you must teach Key Stage 1 year groups (Reception, Year 1 or Year 2).

Our research aims to explore whether you have any experience with computing and your opinions about using robotics within the classroom. Previous experience with these concepts is not necessary for you to take part in our discussions! Any information provided by teachers who feel inexperienced in this area will be just as valuable as the information provided by those who have taught these concepts before. In addition to your views about computing, we'd love to hear about your general teaching practices and experiences.

The focus group discussion will take place virtually via Zoom on **Wednesday 19th May 2021** and will run from **12:45 – 15:30**. The session is structured as follows:

- **Session 1 (12:45 - 13:00):** An introduction to our research – the role of the teacher.
- **Session 2 (13:00 – 13:30):** Scratching the surface: Teaching computer science in primary school classrooms.
- **Session 3 (13:30 – 14:00):** Robotics: A resource you'd love to have in your classroom… or is it?
- **Break (14:00 – 14:15)**
- **Session 4 (14:15 – 14:45):** How can we facilitate student learning?
- **Session 5 (14:45 – 15:15):** What makes a good classroom curriculum?
- **Closing remarks (15:15 – 15:30)**

Your participation will be instrumental in moving our project forward and so, as a thank you, you will receive a copy of the book "*Ada Twist, Scientist*" for your classroom after our session! To accompany the book, we'd be happy to arrange a time for your students to talk to us (as scientists) and for them to ask us any questions they have.

Additionally, if you are interested, some teachers will get the opportunity to partake in our robotics classroom intervention, provisionally scheduled to take place in 2022! In this future study, we will be exploring how programmable robotics can be used in the classroom to help children aged 4 to 7 years old develop a host of computational thinking and STEM skills. Those that participate in our intervention study will receive a robotics playset as a "thank you"!

If you are interested in participating in our focus group discussion and would like to hear more about the study, you can email Amy Hughes (Ph.D. student) at hughesaa1@cardiff.ac.uk.

Appendix B

Survey Recruitment Advert (Chapter 3)

Foundation Phase teachers… can you help?

We are inviting **Reception**, **Year 1** and **Year 2** teachers to take part in our online survey! At Cardiff University, we are investigating how we can integrate robotics into foundation phase classrooms.

A large part of our research is centred around exploring your experiences with computer programming and your opinions about using robotics within the classroom. We are also interested in hearing about whether your school provides teaching support for these concepts.

The findings of this survey will be used to design a classroom intervention with the aim of making programming more accessible for children aged 4 to 7. We hope that by working with teachers like you, we can create a robotics program that benefits both staff and pupils! You can learn more about our future research at the end of the survey.

Please note: Previous experience with programming and/or robotics is not required for you to take part in this research.

Enter our prize draw! As a "thank you" to those who fill out our survey, we are giving away some robotics kits. Enter your details at the end of the survey and you will be in with a chance of winning one of our kits for your classroom!

Survey link: https://cardiffunipsych.eu.qualtrics.com/jfe/form/SV_beGxDmzA3HNHDym

If you have any questions, you can contact Amy Hughes (PhD student) at hughesaa1@cardiff.ac.uk.

This survey will close on Monday, 4th October 2021.

Athrawon Cyfnod Sylfaen... allwch chi helpu?

Rydyn ni'n gwahodd athrawon dosbarthiadau **Derbyn**, **Blwyddyn 1** a **Blwyddyn 2** i gymryd rhan yn ein harolwg ar-lein! Ym Mhrifysgol Caerdydd, rydyn ni'n ymchwilio i sut gallwn ni integreiddio roboteg i ystafelloedd dosbarth y cyfnod sylfaen.

Mae rhan fawr o'n hymchwil yn canolbwyntio ar archwilio eich profiadau gyda rhaglennu cyfrifiadurol, a'ch safbwyntiau am ddefnyddio roboteg yn yr ystafell ddosbarth. Rydyn ni hefyd yn awyddus i glywed a yw eich ysgol yn rhoi cymorth addysgu mewn perthynas â'r cysyniadau yma.

Bydd canfyddiadau'r arolwg yma'n cael eu defnyddio i gynllunio ymyrraeth ystafell ddosbarth gyda'r nod o wneud rhaglennu'n fwy hygyrch i blant rhwng 4 a 7 oed. Gobeithio, drwy weithio gydag athrawon fel chi, y gallwn ni greu rhaglen roboteg sydd o fudd i staff a disgyblion! Gallwch ddysgu mwy am yr ymchwil sydd i ddod ar ddiwedd yr arolwg.

Dylech nodi: Does dim angen profiad blaenorol gyda rhaglennu a/neu roboteg er mwyn cymryd rhan yn yr ymchwil yma.

Ymunwch â'n raffl! Fel "diolch" i'r rhai sy'n ateb ein harolwg, rydyn ni'n rhoi pecynnau roboteg yn wobr. Nodwch eich manylion ar ddiwedd yr arolwg, a bydd gennych gyfle i ennill un o'r pecynnau ar gyfer eich ystafell ddosbarth!

Dolen i'r holiadur: https://cardiffunipsych.eu.qualtrics.com/jfe/form/SV_24dGag4k9Hve3aK

Os oes gennych unrhyw gwestiynau am yr holiadur yma, mae croeso i chi gysylltu ag Amy Hughes (y Prif Ymchwilydd) drwy hughesaa1@caerdydd.ac.uk.

Appendix C

Online Survey Completed by Teachers (Chapter 3).

Introduction

Dear Teachers,

We are inviting **Reception**, **Year 1** and **Year 2** teachers who **work in Wales** to participate in this online survey!

Through this survey we hope to find out what experiences primary school teachers may have with teaching computer programming and robotics. Additionally, we are interested in hearing about whether your school provides teaching support for these concepts.
Please note: previous experience with these concepts is <u>not</u> required for you to take part in this research.

We hope that your contribution will help us understand how teachers perceive robots and computer programming, so that we can successfully integrate these concepts within everyday classrooms.

The data you provide will be anonymous, stored securely and will only be associated with your unique Participant ID Number and not your personal details.

At the end of the survey, you will be given the opportunity to enter a **<u>prize draw to win a robotics kit for your classroom!</u>** These will be distributed following the conclusion of our 2022 study.

If you have any issues or questions about this questionnaire, please don't hesitate to contact Amy Hughes (Primary Researcher) at hughesaa1@cardiff.ac.uk.

If you would like to continue with the survey, please click the button below.

Consent

I understand that my participation in this study will involve answering questions regarding my own experiences with computer programming. This will require no more than 15 minutes of my time.

I understand that participation in this survey is entirely voluntary and that I can withdraw at any time, without giving a reason. I understand that I am able to contact the researchers Amy Hughes (Hughesaa1@cardiff.ac.uk) or Dr Sarah Gerson (GersonS@cardiff.ac.uk) to ask any questions, or discuss any concerns I might have.

I understand that completing this survey does not involve any significant risks.

I understand that the information provided by me will be held anonymously, so that it is impossible to trace this information back to me individually. I understand that this information may be retained indefinitely.

I also understand that at the end of the survey I will be provided with additional information about the purpose of the study.

My consent below, using the checkboxes provided, indicates that I have read this form, had the opportunity to ask questions about my participation and voluntarily consent to participate. I am aware that I can print this as a copy for my own records.

- ○ I consent to participate in this survey.
- ○ I do NOT consent to participate in this survey.

Participant Information

Gender

- ○ Male
- ○ Female
- ○ Non-binary / third gender
- ○ Other ________
- ○ Prefer not to say

In what county of Wales is your school located?

- ○ Isle of Anglesey (Ynys Môn)
- ○ Blaenau Gwent
- ○ Bridgend (Pen-y-bont ar Ogwr)
- ○ Caerphilly (Caerffili)
- ○ Cardiff (Caerdydd)
- ○ Carmarthenshire
- ○ Ceredigion
- ○ Conwy
- ○ Denbighshire (Sir Ddinbych)
- ○ Flintshire (Sir y Fflint)
- ○ Gwynedd
- ○ Merthyr Tydfil
- ○ Monmouthshire (Sir Fynwy)
- ○ Neath Port Talbot (Castell-nedd Port Talbot)
- ○ Newport
- ○ Pembrokeshire
- ○ Powys
- ○ Rhondda Cynon Taf
- ○ Swansea (Abertawe)
- ○ Torfean
- ○ Vale of Glamorgan (Bro Morgannwg)
- ○ Wrexham (Wrecsam)
- ○ Other ____________

Current job role

- ○ Pre-Service Teacher
- ○ Full Time Teacher
- ○ Part Time Teacher
- ○ Teaching Assistant
- ○ Higher Level Teaching Assistant (HLTA)

○ Other ______

What year group(s) do you **currently** teach? *(This question refers to the 2021/22 academic year. The rest of the questions in this survey will ask you to think about these pupils specifically).*

☐ Reception
☐ Year 1
☐ Year 2
☐ Other ______

Years teaching experience

______

School type

○ Government Funded
○ Private
○ Special Educational Needs
○ Other ______

How did you hear about this survey?

○ Email
○ Social media
○ From another teacher
○ Other ______

What teaching qualification do you have?

☐ BEd
☐ PGCE

- [ ] PGDE
- [ ] HLTA Certificate
- [ ] Other [   ]

What is your primary degree?

[   ]

Do you have any postgraduate certificates or degrees that are specific to computing?

- ( ) Yes
- ( ) No
- ( ) I am unsure

What postgraduate qualifications do you have?

[   ]

Thinking about access to technology resources in your classroom, please indicate which of the following statements are relevant.

- [ ] My pupils have infrequent access to devices.
- [ ] I wish we had access to devices more often.
- [ ] I wish we had access to more devices.
- [ ] Our pupil : device ratio is high.
- [ ] I'm happy with how often we can use our devices.
- [ ] My pupils have frequent access to devices.
- [ ] I'm happy with the number of devices we have access to.
- [ ] Our pupil : device ratio is low.

What technological devices do your pupils use?

- [ ] Laptops
- [ ] Computers

- ☐ iPads
- ☐ Robotics
- ☐ Interactive whiteboard
- ☐ Chromebooks
- ☐ Digital camera
- ☐ Other ______

How many hours a week do the majority of your pupils use laptops?

______

How many hours a week do the majority of your pupils use computers?

______

How many hours a week do the majority of your pupils use iPads?

______

How many hours a week do the majority of your pupils use robotics?

______

How many hours a week do the majority of your pupils use an interactive whiteboard?

______

How many hours a week do the majority of your pupils use Chromebooks?

______

How many hours a week do the majority of your pupils use digital cameras?

______

How many hours a week do the majority of your pupils use "other" devices?

[ ]

Computational thinking

To what extent are you familiar with the term "computational thinking"?

- ○ Not at all familiar.
- ○ Slightly familiar.
- ○ Very familiar.

Please read the statements below and let us know whether you think they fit a definition of computational thinking (CT). You can do this by dragging each statement to either of the boxes.

Items

CT involves imagination and creativity.

13 / 13

Does fit my definition of "computational thinking".

Does not fit my definition of "computational thinking".

I'm unsure about this statement.

Computer programming

Programming (also called coding) can be defined in a variety of ways, but it is often thought of as creating directions or instructions for a computer or robot that direct behaviour (i.e., events and sequences of events).

Do you teach programming to your pupils?

- ◯ Yes
- ◯ No
- ◯ I am unsure

How often do you teach programming to your pupils?

- ◯ Weekly.
- ◯ Fortnightly
- ◯ Monthly.
- ◯ Less than monthly.

Would you like to teach more programming in your lessons?

- ◯ Yes.
- ◯ No, I feel like I teach enough.
- ◯ I am unsure.

Why do you want to teach more programming?

Are there things that make it difficult for you to teach more programming? What's stopping you from doing so?

Are you interested in teaching programming to your pupils?

- ◯ Yes
- ◯ No
- ◯ I am unsure

Why do you want to teach your pupils programming?

Are there things that make it difficult for you to teach programming to your pupils? What's stopping you from doing so?

Why aren't you interested in teaching programming?

What tools do you use to teach programming to your pupils?

In the table below you can list the tools you use in the first column. Tell us about your experiences with these tools using columns two and three.

	Tool used to teach programming:	Advantages and/or positive experiences:	Disadvantages and/or challenging experiences:
1			
2			
3			
4			
5			

Do you use classroom resources provided by external organisations when teaching programming? *e.g., Barefoot computing, Code.org etc.*

- Yes
- No
- I am unsure

Please tell us a bit about the resources you used in the past! What do you like about these resources? Is there anything you dislike or would change?

Robotics

This next section is about robotics!

Educational robotics combine accessible and age-appropriate materials to provide children with knowledge of programming through hands-on, practical experiences.

Some examples of robotics are Bee-Bot, LEGO Mindstorms and the LOGO floor turtle.

How much experience do you have with programmable robotics?

- ○ I have never used robotics before.
- ○ I have used robotics before but not with my pupils.
- ○ I have used robotics with my pupils before, but I don't use them often.
- ○ I use robotics frequently with my pupils.

Would you like to use robotics with your pupils?

- ○ Yes
- ○ No
- ○ I am unsure

Are there things that make it difficult for you to use robotics in the classroom? What's stopping you from using them with your pupils?

Please tell us about how you have integrated robotics into your teaching activities.

Would you like to use robotics in your lessons more often?

- ○ Yes
- ○ No, I feel I use them enough
- ○ I am unsure

Are there things that make it difficult for you to use robotics in the classroom? What's stopping you from using them more often?

The following statements will get you to think about how confident you may or may not feel about using robotics. Please indicate whether you agree or disagree with each statement.

	Agree	Disagree	I am unsure
I would feel at ease using robotics in my lessons.	○	○	○
I feel confident I can incorporate robotics	○	○	○

	Agree	Disagree	I am unsure
with other subjects.			
I am confident that I could learn how to use robotics with my pupils.	○	○	○

Below are a series of statements about robotics. You can indicate whether you agree or disagree with each statement by dragging it to the relevant box.

Using programmable robotics...

Items

Gives teachers the opportunity to be learning facilitators instead of information providers.

Mathematics).

15 / 15

I agree with this statement

I disagree with this statement

I am unsure about this statement

Below are a series of statements which may be considered obstacles that prevent teachers from using robotics with their pupils. We'd like for you to indicate how big of an **obstacle** you perceive each item to be.

The scale points range from 0 - "Minor obstacle" to 100 - "Major obstacle".

You should give the most points to the item you consider to be the biggest obstacle, fewer points to the statements you consider to be a smaller obstacle. If you believe something is not an obstacle, please check the box to the right of the scale.

	Minor obstacle — Major obstacle 0 10 20 30 40 50 60 70 80 90 100	Not an obstacle
Teachers do not have the time to learn how to integrate robotics into their lessons.		☐
Schools do not have the space to store and use multiple robot devices in the classroom.		☐
Teachers do not feel confident enough to use robotics in their classes.		☐
My pupils are too young to be able to understand and work with robotics.		☐
Teachers do not have adequate instructional support.		☐
There is too much course content to teach to find time for robotics.		☐
There are not enough technical resources available in school.		☐
Class sizes are too large to plan lessons using robotics.		☐
Teachers are unsure how to make		☐

	Minor obstacle						Major obstacle				Not an obstacle
	0	10	20	30	40	50	60	70	80	90	100
robotics relevant across multiple subjects.											
Teachers do not have adequate technical support.											☐
My pupils wouldn't be motivated to learn with robotics.											☐

Below are some statements about robotics and future development. Please indicate whether you agree or disagree with each one.

	I agree	I disagree	I am unsure
Teaching my pupils basic programming languages now will help them progress to more advanced programming languages in the future.	○	○	○
Using robotics will encourage my pupils to pursue a STEM career.	○	○	○
In 20 years time, knowledge of programming will be needed in many careers.	○	○	○
Using robotics with my pupils will discourage gender stereotypes in STEM subjects.	○	○	○
Using robotics will help my pupils to become lifelong learners.	○	○	○

Training/ Internal support

Did your teacher education program (i.e., BEd, PGCE etc.) include programming, computational thinking or robotics?

- ◯ Yes
- ◯ No
- ◯ I have not yet completed my training
- ◯ I am unsure

Has your school ever provided you with training on how to teach programming to your pupils?

- ◯ Yes
- ◯ No
- ◯ I am unsure

Did this training include advice on how to use robotics in the classroom?

- ◯ Yes
- ◯ No
- ◯ I am unsure

What did you enjoy about the training you attended? Was there anything missing from the training that you would have changed?

Do you feel you need training on how to use robotics in your classroom?

- ◯ Yes
- ◯ No
- ◯ I am unsure

Does your school currently offer internal support for the teaching of programming and other computing concepts?

- ○ Yes
- ○ No
- ○ I am unsure

Please tell us a bit about the support you have access to. What are the benefits and/or limitations of the help you have available?

Has the COVID-19 pandemic impacted the amount of support you have access to?

- ○ Yes
- ○ No
- ○ I am unsure

Please tell us a bit about this.

Do you think you would benefit from having extra support with teaching programming?

- ○ Yes
- ○ No
- ○ I am unsure

Please describe how you could be better supported in this area.

Do other members of staff usually turn to you for computing advice or technical support?

- ○ Yes
- ○ No
- ○ I am unsure

Offering this support is...

- ○ Officially part of my job role
- ○ Not officially part of my job role (i.e., is something extra I do to help my colleagues)
- ○ I am unsure how to answer

Please tell us a bit about how you support your colleagues!

Future Research Interest

<u>Introducing Cubetto!</u>
Cubetto is the programmable robot at the centre of our upcoming research!

Provisionally scheduled to take place at the start of 2022, our study will combine a classroom intervention for pupils (aged 4 to 7) with professional development training for teachers.

Our main goals are:
1. To create a robotics curriculum that can be used in your classroom.
2. To deliver some training that focuses on learning how to teach computational thinking, programming and robotics to foundation phase pupils.

We will be assessing the impact such a program has on children's learning and skill development. We'll also be measuring the effects of teacher training.

Those that take part will be provided with all the necessary Cubetto equipment and will also get to keep a playset after we have concluded our study! **There is no cost for you or your school to participate in this study.**

Watch the video below to find out how Cubetto works!

If you have any questions or if you're interested in receiving more details as things progress with our research, please leave your contact details below and we will be in touch! *(The details you provide here will be stored separately from your survey responses to maintain your anonymity.)*

Name

Email address

Debrief

Thank you for taking the time to complete this survey!

Your responses will be used to help us better understand how teachers perceive robots and computer programming so that we can integrate these concepts within everyday classrooms. As outlined in the information prior to taking part in the survey, your data will be stored anonymously and will not be traceable back to you.

If you are interested in the overall findings from this survey (reporting only average responses and general findings) or have any concerns, then please feel free to contact the researchers Amy Hughes (Hughesaa1@cardiff.ac.uk) or Dr Sarah Gerson (GersonS@cardiff.ac.uk).

If you'd like to be entered into a **prize draw** to win a Cubetto Playset for your classroom, please leave your contact details below! *(The details you provide here will be stored separately from your survey responses to maintain your anonymity.)*

Name	
Email address	

Powered by Qualtrics

Appendix D

School Information and Recruitment Document (Chapter 5).

WHO ARE WE?

AMY HUGHES

PhD Student

Cardiff University Centre for Human Developmental Science
Cardiff University
Email: Hughesaa1@Cardiff.ac.uk

DR SARAH GERSON

Primary Supervisor

School of Psychology
Cardiff University
Email: GersonS@Cardiff.ac.uk

DR JOHANNA VAN SCHAIK

Secondary Supervisor

Radboud University

WHAT IS THIS PROJECT ABOUT?

A brief summary.

In the modern world, gaining technological skills is important for advancing education and innovation. Computer programming was originally thought to be a domain only accessible to individuals in higher-education institutions, but it has recently been discovered that children can begin to learn the basics of programming in fun and novel ways. We are examining how and when children begin to learn to program.

This push to teach computer programming in early childhood education is a fairly new movement, and so it's important to acknowledge that such a task can appear daunting to teachers with little experience in this area.

Research has shown that low teacher confidence can negatively impact pupil achievement. Thus, we also want to explore the impact our professional development training has on children's learning experiences and skill development.

This research is being conducted in collaboration with Techniquest, a science centre for children based in Cardiff. So far, the Techniquest team have been instrumental in helping us recruit teachers for our previous online studies. Additionally, we are collaborating with Primo Toys, a UK based technology company (find out more on page 5).

OUR RESEARCH

What are our aims? What have we done so far?

We are investigating how children first learn computer programming and whether programmable robotics can be successfully integrated within the Foundation Phase. We want to target integration from both the teachers' perspective as well as the students

In the last year, we have been investigating how primary school teachers perceive educational robotics and how they feel about using robotics to encourage computational thinking and the development of other STEM (Science, Technology, Engineering and Mathematics) skills within the classroom. Back in May, 13 teachers from across Wales met with us virtually to participate in a focus group session. The discussions that took place during this session then help us create our online survey which was distributed at the start of the school year. The information gathered from both these studies has been used to inform the development of this classroom intervention.

Over the summer we have also been back in our research lab! We have been assessing the impact visual perspective taking skills may have on a child's ability to program a robot within their physical environment. We invited children aged 4 to 7 years into the lab to play some games with a robot.

We observed whether techniques designed to support perspective taking (i.e., standing up and walking through the program as they planned it) aided programming performance as children explored how to move the robot from location to location on a floor map. We also trialled a range of games that will be used in our new study to assess things like working memory, visual perspective taking and prediction skills.


Now we get to take what we have learned during the 3 studies mentioned above and utilise the findings in our new project! The current study will combine a classroom intervention for pupils (aged 4 – 7) with professional development training for teachers.

Our main goals are to 1) create an engaging, multi-disciplinary robotics curriculum that can be used in the classroom and 2) to deliver training that focuses on learning how to teach computational thinking, programming, and robotics to FP pupils.

We will be assessing the impact our program has on children's learning and development. Additionally, we'll be measuring the effects of teacher training.

Participating schools will be provided with all the necessary learning materials and robotics equipment. **There is no cost for your school to participate in this study.**

We are inviting Reception, Year 1 and Year 2 classes to join our project, and we encourage you to volunteer as many classes as possible! Participating schools will receive enough Cubetto robots for each class for the duration of the study.

As a "thank you" for helping us with our research, your school will receive (at least) one robotics kit to keep!

NEXT STEPS ↑

INTRODUCING...

PRIMO TOYS AND CUBETTO!

For this project, we are working in collaboration with UK tech company **Primo Toys.** They are a team of technologists, designers and educators who make toys that help children learn with technology. They support the notion that coding is the new literacy and therefore, should be prioritised at an early age. They think that for girls and boys, all over the world, learning to program will be as important as their ABCs and 123s in helping them understand the world around them. As a result, they have developed Cubetto, a wooden cuboid robot designed to teach computer programming to young children.

Cubetto's design avoids textual and numerical language thus making it suitable for pre-literate children. Moreover, children can use the playset to get to grips with coding concepts through hands-on learning and without the use of a screen. This is what sets Cubetto apart from other educational robotics on the market.

As you can see in the image below, Cubetto comes equipped with an interface board, a range of function tokens (i.e., forward, right, and left turn functions) and a colourful floor map. Children can navigate the robot around the map by placing the desired tokens in the interface board and pressing the 'Go' button. Through play, Cubetto introduces and provides the foundations for a host of programming concepts including 'algorithms' and 'debugging'. Moreover, it is thought that Cubetto can also support the development of broader STEM skills beyond computer programming, and this is what our project aims to empirically investigate.

Please note: Primo Toys are not directly funding this research, however, they are kindly providing all of the Cubetto equipment. Although the company will be free to use the broader findings from this project for marketing purposes, they will not have access to the data that we collect or the details of the schools, teachers or pupils taking part.

Follow the links below to see Cubetto in action!

- Meet Cubetto!
- How does Cubetto work?
- Cubetto in the classroom 1
- Cubetto in the classroom 2

SO, WHAT'S INVOLVED IN THIS PROJECT?

Teacher Training

Teachers participating in our project will be invited to attend a training workshop with us. Our workshops will run in the New Year and in the Summer Term (your school will be randomly allocated to a session). We are prepared to run our sessions in person and/or virtually, however, this will depend on Government Covid-19 guidelines at the time.

The workshops will be open to those responsible for delivering the Cubetto program to the nominated class. This workshop will focus on the following:

- Helping teachers understand how they can integrate robotics with other classroom subjects.
- Building teacher confidence.
- Providing demonstrations with Cubetto.
- Ensuring teachers have opportunities for hands-on learning with Cubetto devices.

Cubetto Lesson Plans

Teachers will have access to a range of Cubetto lesson plans. These plans have been designed to integrate programming with other classroom subjects (e.g., maths, science, personal and social health, expressive arts and design, and literacy) in a fun and interactive way. We hope that the materials we provide are flexible enough that teachers can adapt the plans to best suit the children. For example, teachers will be able to pick and choose which aspects of the program they teach over a 6 week period. These plans will also come with fun printable resources that support the activities outlined in the program.

Pupil Assessments

For us to assess the impact of the Cubetto program, we'd like to come into the classroom to work with children and see how they perform on several programming related tasks. In these sessions, the children will play a range of fun games designed to assess their programming, debugging, prediction, working memory and visual perspective-taking abilities. The majority of these games are paper-based and children will be asked to do some drawing. The other games will be completed using an iPad.

We have trialled all these games with children in our lab to ensure that they are developmentally appropriate and enjoyable. For this part of the project, parental consent will need to be obtained for each individual pupil. We will not assess any child that does not have written parental consent.

IMPORTANT INFORMATION

All the children that participate in our project will be assessed at the same points in the school year (after the February half term and after the Easter break). Some schools will receive our Cubetto lesson plans prior to this testing, while others will receive them afterwards. Similarly, some teachers will receive training early in the year, while others will get it later. This will allow us to assess the effectiveness of our intervention while ensuring that all participating schools get to experience our Cubetto program and teacher training.

Each school will be randomly assigned to one of our three study groups (**please note:** you will find out which group your school has been assigned to prior to the study beginning). The table below illustrates the project timeline for each of the three groups.

Group no.	January - February half term	After February half term - Easter Holiday	After Easter Holiday - Summer Half term	After Summer half term - Summer holiday
1	Pupil assessments. Teacher Training.	Cubetto curriculum	Pupil assessments.	
2	Pupil assessments.	Cubetto curriculum	Pupil assessments.	Teacher training
3	Pupil assessments.		Pupil assessments.	Cubetto curriculum Teacher training

THAT'S ALL FOR NOW!

Thank you for taking the time to read this information booklet.

ANY QUESTIONS?

Feel free to get in touch! We'd be happy to answer any questions you have.

Amy Hughes - Hughesaa1@cardiff.ac.uk
Dr Sarah Gerson - GersonS@cardiff.ac.uk

INTERESTED IN PARTICIPATING?

If you'd like your school to participate or would like to hear more information, please complete this form.
Your school will then be invited to attend a virtual "drop-in" session to discuss the project in more detail. We'll also answer any questions that you have!

Important - Schools must have consent from the Head of School before a member of staff completes this form.

Please be aware that spaces on this project are limited and so we may not be able to accept every school that expresses interest.

Appendix E

Six Compulsory Lesson Plans (Chapters 5 and 6)

Introduction

The Cubetto Playset is a Montessori inspired coding toy that allows children ages 3 to 7 to program a friendly wooden robot without screens and is powered by a programming language you can touch.

New technology can sometimes be overwhelming to understand and adopt. The activities contained in this guide were created by educators for educators.

We want to make it simple for you to integrate the Cubetto Playset and its tangible programming language into your teaching.

Development and learning in other key areas

Beyond coding

The collaborative nature of Cubetto makes it an extremely versatile tool for the classroom. Cubetto fosters learning in key development areas that go beyond programming.

Communication

Children practice listening through a range of stories and narratives in relation to Cubetto, accurately anticipating key events and responding with comments, questions or actions. They also develop their own narratives and explanations.

Dexterity

Children develop coordination in large and small movements around the playset. They negotiate the placement of obstacles around the world map and place blocks on our tangible interface.

Social-Emotional

Children become confident by trying new, open-ended activities that remove "wrong" outcomes, and easily encourage group work. The open nature of the maps allows them to choose the resources they need for their play session.

Mathematics

Children add and subtract blocks to a sequence. They solve problems, including doubling and halving to get Cubetto from A to B. They discuss size, shapes and patterns, distance, position, and time to solve problems.

Logical reasoning

The blocks allow children to create and debug simple programs with their hands. They use technology purposefully to create, organise, store, manipulate and retrieve meaningful sequences.

Introducing the Playset

Introducing Cubetto

Introduce Cubetto as a friendly robot that children can program. Children should be told that Cubetto cannot think for himself, and can only move as programmed by the child, just like any other machine. If in a group setting, sit children in a circle, and allow them to pass Cubetto around to one another, saying hello or acknowledging the presence of the object.

Doing so forms a bond with Cubetto, in the same way they would with a stuffed animal, or a toy, and solving problems through narratives later on is more engaging.

Introducing the Board

Introduce the Board as a remote control that children can use to send instructions to Cubetto.

Without the Board, there is no way of sending Cubetto his instructions.

It is important for children to understand Cubetto is only able to move with a human's command. This is not only empowering, but also key to understanding computing.

Encourage children to also explain what other objects in their homes and lives function within a similar paradigm. A television needs a human to change its channels for example, or a washing machine needs a human to select its settings.

These examples, like Cubetto, are machines that need human programming to do their job.

Introducing the Blocks

Introduce the Instruction Blocks as the directions Cubetto follows when inserted in the Board and sent by pressing the action button.

Different Blocks represent different instructions, and an unambiguous, distinct command. These Blocks are what make up Cubetto's hands on coding language, and are key in the learning of computational thinking.

When each block is inserted in the Board, a child should be encouraged to predict what Cubetto will execute before pressing the "Go" button.

This is key in understanding concepts like program design, and it helps develop abstraction.

Table of contents

	Lesson 1	Lesson 2	Lesson 3
NC Computing Objectives	To understand how algorithms are implemented on devices	To explore a digital device	To use logical reasoning to predict behaviour of simple programs
Outcomes	• I can put instructions in order • I can follow an algorithm	• I can make Cubetto move • I can talk about the features of different 2D shapes	• I can predict what an algorithm will do • I can make new parts for a game
Maths Focus	ELG 12 (Shape, space and measure)	Maths - space & shape (ELG 12)	Expressive Arts & Design (ELG 16)
Computational Thinking	Logic, Tinkering	Algorithms, Tinkering	Logic, Persevering
Main Activities	Cubetto's cake 1. Write a recipe for Cubetto's birthday cake together. 2. Program Cubetto to pick up the instructions on the map in order. 3. Design the decoration for Cubetto's birthday cake. 4. Program Cubetto to pick up the cake slices. 5. Share the cake pieces between your friends.	Cubetto the 2D shape detective 1. Guess which 2D shape is missing in Cubetto's memory game. 2. Make a Shape Robot, moving Cubetto around the map collecting 2D shapes. 3. Take a 2D shape from a feely bag using a puppet and talk about its features. 4. Write a letter to Cubetto explaining what you would do together if you took him home for the weekend.	Cubetto's Quest 1. Make new parts to turn the map into a Snakes and Ladders game. 2. Predict where Cubetto will move and test out your prediction. 3. Work with a partner to play Snakes and Ladders. 4. Make your own models to add to the map!
Challenge	Does your algorithm still work with a block missing?	Can you talk about what happens when Cubetto goes wrong?	Snakes and Ladders templates, Dice numbered 1-4, Grid on paper
Resources	Pictures of cakes (and disasters), cake recipe algorithm, play cake slices.	2D shape stickers, 2D plastic shapes, hand puppets, blank letter template.	
Assessment	Ordered recipe, Cake, Verbal statements , Photos, Observation of sharing, Algorithm picking up cake slices	Algorithm painted prints,observation of role play and mirror activities, verbal statements and photos	Algorithm predictions, Models and new game parts created, Photos, Verbal statements, Observation

	Lesson 4	Lesson 5	Lesson 6
NC Computing Objectives	To use logical reasoning to predict behaviour of simple programs	To debug a simple program	To use logical reasoning to predict behaviour of simple programs
Outcomes	• I can predict what a program will do • I can draw a treasure map	• I can debug a simple algorithm • I can identify 2D shapes	• I can predict what a simple algorithm will do • I can create the boundaries of a maze
EYFS Focus	Art	Maths	Maths: Position & Movement (Direction)
Computational Thinking	Logic, Perseverance	Algorithms, Debugging	Logic, Collaborating
Main Activities	Cubetto's Quest 1. Draw your own treasure map for Cubetto to explore. Mark on where the treasure is. Write an algorithm to get there. 2. Work out where the teacher hid the treasure by predicting where the algorithms will take Cubetto.	Cubetto's Dance 1. Debug a series of algorithms in a group to reach a chosen square containing a shape. 2. In pairs one writes an algorithm to make Cubetto 'dance' to a square with a 2D shape. Hiding one step, the other debugs it.	**Cubetto's Maze Maddness!** 1. Create your own maze on the map with an entrance and exit. Write algorithm to get Cubetto to the end of the maze. 2. Solve the maze and write instructions to help someone find the exit. Add a monster as a trap!
Challenge	Can you say the coordinates of where the treasure is hidden?	Can you write an algorithm to make Cubetto dance in a circle?	Can two Cubettos use your maze at the same time?
Resources	Primo maps, A4 squared paper	Primo board template, Algorithms to debug, 2D shapes , Music (optional)	Maze template, Maze images, Blank 4x4 grid, Coloured card Map grid, Verbal statements,
Assessment	Treasure maps, Photos, Verbal statements, Observation	Photos, Verbal statements, Observation	Observation

Lesson 1: Cubetto's Cake (1 of 2)

EYFS Focus: maths (ELG 12)

NC Objectives
To understand how algorithms are implemented on devices

Outcomes
- I can put instructions in order
- I can follow an algorithm

Resources Needed
- Candles
- Pictures of cakes (and disasters)
- Sticky tack
- Cake recipe algorithm
- Play cake slices

Prep Needed
- Check batteries.
- Copy and cut up cake algorithm

Resources Provided
- N/A

Key Vocabulary
- Programming
- Recipe
- Order
- Algorithm

Computational thinking concept
Logic

Computational thinking approach
Tinkering

Teacher-led Introduction
1. Show images of cake disasters (e.g. https://www.boredpanda.com/funny-cake-fails-expectations-reality/?utm_source=google&utm_medium=organic&utm_campaign=organic) and ask: Have you ever made something that went wrong?
2. Explain that often things go wrong because we forget to do something, or we do things in the wrong order.
3. Explain that making and decorating a cake is like programming Cubetto - we have to do the steps in the right order.
4. Ask: Does anyone know how to make a cake? Collect and note down children's ideas for a recipe.
5. Ask: Can we do these steps in any order? For example, should we put the flour in the oven first?
6. Emphasise that a recipe needs to be followed carefully and in order. If a step is missed, the cake won't taste or look nice!
7. Show this cake making video from 40 seconds in: https://goo.gl/0gFbDI and ask: What was the first thing the chef did?
8. Compile a list in numbered steps (e.g. mix butter and sugar, mix in eggs, fold in flour, add vanilla and pour milk, put in oven for 30 minutes).
9. Explain that a recipe needs to be in order, just like when programming a computer.
10. Introduce the word 'algorithm' as a set of ordered instructions for a computer.
11. Watch sequencing video: https://www.youtube.com/watch?v=zW3YZdPmCnM&ab_channel=ScratchGarden

Lesson 1: Cubetto's Cake (2 of 2)

Activity 1: Recipe algorithm

1. Read and put the cake recipe algorithm in order.
2. Place the steps next to each other on squares of the map.
3. Program Cubetto to move along the steps to make a cake.

Activity 2: No disasters!

1. Print out cake slice templates.
2. Decorate cake slices for Cubetto.
3. What might Cubetto like on his cake? What colours could you use?

Activity 3: Whole cakes

1. Place the cake slices around the map.
2. Program Cubetto to pick up as many cake slices as you can!

Activity 4: Sharing slices

1. Work in pairs and lay out the cake slices on the table.
2. How many slices do you have in total?
3. Share the slices equally between you and your partner, then count them.

Challenge

Does your algorithm still work with a block missing?

Plenary and Assessment

1. Ask: Why is it important to do things in the right order? What would happen if we didn't? What else do we do every day that needs to be in order?
2. Ask pupils to tell the person next to them how to brush your teeth, starting with putting toothpaste on the brush.
3. Ask pupils to share how they programmed Cubetto to collect the recipe steps. Ask: What did you find difficult?
4. Show the recipe on the board with one step missing and ask: Would this work? Why/why not?
5. Place steps on consecutive squares on the map, with Cubetto starting on the first step. Ask pupils to write an algorithm to make Cubetto follow the steps.

Lesson 2: Cubetto the 2D Shape Detective (1 of 2)

EYFS Focus: Maths (ELG 1)

NC Objectives

To explore a digital device

Outcomes

- I can make Cubetto move
- I can talk about the features of different 2D shapes

Resources Needed

- 2D shape stickers and plastic shapes
- Hand puppets
- Blank letter template

Prep Needed

- Check batteries.
- Hide 2D shapes around the room.
- Create stamps in shape of Cubetto blocks.

Resources Provided

- N/A

Key Vocabulary

- 2D shape names
- Flat
- Properties

Computational thinking concept

Algorithms

Computational thinking approach

Tinkering

Teacher-led introduction

1. Ask pupils to look around the classroom and see if they can spot 2D shapes that Cubetto has hidden. Emphasise that 2D means flat.
2. Allow time for children to find and collect the 2D shapes. When brought back to the carpet, stick them on the board.
3. Discuss each shape: What is this shape called? What is special about this shape? Can you describe what it looks like?
4. Choose four shapes and explain that they are going to play a memory game. Ask: Can you remember the names of these shapes?
5. Ask pupils to close their eyes. Remove one of the shapes from the board and tell the children that Cubetto has taken one of the shapes.
6. Ask: Which shape is missing? Can you describe that shape?
7. Repeat for other shapes, introducing the word 'properties' for the way we describe what a shape looks like.
8. Explain that today the pupils will be helping Cubetto hunt for all the 2D shapes around the map!

Lesson 2: Cubetto the 2D Shape Detective (2 of 2)

Activity 1: Shape Robot

1. Choose a shape sticker placed on the map.
2. Move Cubetto to land on a 2D sticker on the map.
3. What shape have you collected? Can you describe it?
4. Attach the shape sticker to Cubetto and repeat.

Activity 2: Puppets

1. Work in pairs, each with a puppet.
2. The puppets choose their favourite shapes from the bag.
3. Each puppet tells the other all about the shape's properties.

Activity 3: Cubetto's pen pal

1. Where on the map would you like to travel to on an adventure?
2. What would you take with you? What would you do when you got there?
3. Write a letter to Cubetto explaining your adventure.
4. Program Cubetto to go on your adventure! Where will they go first?
5.

Challenge

Can you talk about what happens when Cubetto goes wrong?

Plenary and Assessment

1. Show a 2D shape and ask: Cubetto thinks this is a rectangle, is he right? No. Ask: How do you know it is not a rectangle?
2. Ask a volunteer to read one of the letters they have written to Cubetto, detailing the adventure they would like to have together.

Lesson 3: Cubetto's Games (1 of 2)

EYFS Focus: Expressive Arts & Design (ELG 16)

NC Objectives

To use logical reasoning to predict behaviour of simple programs

Outcomes

- I can predict what an algorithm will do
- I can make new parts for a game

Resources Needed

- Sticky tape
- Scissors
- Pens and paint
- Snake/ladder templates
- Dice num'd 1-4

Prep Needed

- Check batteries
- Print snake and ladder templates onto card

Resources Provided

- N/A

Key Vocabulary

- Algorithm
- Predicting
- Recycled

Computational thinking concept

Logic

Computational thinking approach

Persevering

Teacher-led Introduction

1. Sit the children in a circle with Cubetto on the map in the middle and the prepared algorithm on the Board.
2. Show the board to the children and ask: What do you think will happen if I press the Action button? How and where will Cubetto move? Collect pupils' ideas.
3. Ask for a volunteer to press Action and tell the class to watch Cubetto move.
4. Ask: Were you right? Did Cubetto move how you thought it would?
5. Explain that trying to work out what will happen is called predicting.
6. Move two of the Blocks around on the Board and ask: How do you think Cubetto will move now?
7. Ask for another volunteer to press Action and discuss their predictions.

Lesson 3: Cubetto's Games (2 of 2)

Activity 1: Snakes and Ladders

1. Choose a snake or ladder and colour it in.
2. Cut out the shape.
3. Stick your shape onto the map using sticky tape.

Activity 2: Play the game (teacher-led)

1. Find a partner who also has a Cubetto and a Board.
2. Put the snakes and ladders on the map.
3. The first person rolls the dice and writes an algorithm to move Cubetto that number of squares. Repeat, taking it in turns.
4. If you land on a snake, go down one. If a ladder, go up one.

Activity 2a: Making predictions

1. Roll the dice.
2. Once children have made their algorithms, encourage them to think about how it will make Cubetto move. Predict carefully.
3. Where will Cubetto move to? If they are going to land on a snake/ ladder, where will they finish then?
4. Discuss if their prediction was right.

Plenary and Assessment

1. In a circle, ask: What does predicting mean? What did we predict today?
2. Ask pupils to share how they worked out the treasure hunt clues and whether their prediction was correct.
3. Ask for pupils to share how they played Snakes and Ladders and to share the shapes they made for the map.
4. Ask for a volunteer to throw the dice in the middle of the circle. Ask: What algorithm would I need to write to make Cubetto move that number? Discuss and try out different Blocks until successful.
5. Explain that often, people who work with computers have to be very patient and keep trying to get things right before it works.

Lesson 4: Cubetto's Quest (1 of 2)

Cross-curricula Area: Art

NC Objectives	Outcomes	Resources Needed	Prep Needed	Resources Provided	Key Vocabulary
To use logical reasoning to predict behaviour of simple programs	• I can predict what a program will do • I can draw a treasure map	• A4 paper with large squares (3x3)	• Check batteries • Choose a square for the treasure to be buried in and mark it on a laminated map (keep it secret!) • Write the algorithm needed to get there	• Primo maps	• Predicting • Algorithm • Clue • Program

Computational thinking concept

Logic

Computational thinking approach

Perseverance

Teacher-led Introduction

1. Ask the children to close their eyes and think about the most precious thing they own (it doesn't have to be an object!).
2. Tell the children that Cubetto has lost something very important to him and is very sad.
3. Explain that today the children will be trying to help Cubetto find it by predicting where an algorithm will take you.
4. Tell the children that you know where Cubetto's treasure is hidden and show the clue (the algorithm) to find it.
5. Ask: Looking at the algorithm, where do you think the treasure is? Why do you think that? How are you predicting where it is?
6. Ask for a volunteer to program Cubetto with the algorithm you showed the class.
7. Encourage the children to discuss whether they predicted correctly and why. Explain that computing often involves trying things lots of times before we get things right and this it is very important to be patient.
8. Model marking on a map where the treasure was buried with a cross.

Lesson 4: Cubetto's Quest (2 of 2)

Creative Play

Make or find some treasure for Cubetto to discover.

Guided Activity

1. On squared paper, make your own map like Cubetto's.
2. Draw different pictures in each square. You might want to choose a theme such as school, sport, music or a game.
3. Decide where on the map your treasure is buried. Mark on the back of your sheet where it is (to keep it secret).
4. Write down where to start on your map.
5. Write an algorithm for where your treasure is hidden: this is your treasure hunt clue.
6. Find a partner and ask them to predict where your treasure is buried by looking at your map and working out the algorithm.

Independent Activity

1. Look at the first treasure hunt clue. Where do you predict the algorithm will take you?
2. Put a cross on the map where you think the algorithm will take you.
3. Program Cubetto using the algorithm clue and press the action button.
4. Did you predict the right place? Why? Why not?
5. If you weren't right, try to work out which part you got wrong.
6. Repeat for other algorithms.

Challenge

Can you say the coordinates of where the treasure is hidden?

Plenary and Assessment

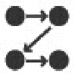
1. Ask for a volunteer to share their treasure map with the class.
2. The class predicts where the treasure is hidden and programs Cubetto to test it out.
3. Ask: Could we make the algorithm simpler or use fewer blocks? Reinforce the importance of making things simpler.
4. Ask: Which clues did you find harder to work out? Why do you think this was?
5. Ask: What does predict mean? What is an algorithm? What does program mean?
6. Ask the children to think about today and what they did. Ask: What skills do people who work in computing need to have? Collect and display. Elicit: try again and again/perseverance; make sure it's correct/be exact/precise; make it better each time/ more efficient.

Lesson 5: Cubetto's Dance (1 of 2)

Cross-curricula Area: Maths

NC Objectives	Outcomes	Resources Needed	Prep Needed	Resources Provided	Key Vocabulary
To debug a simple program	• I can debug a simple algorithm • I can identify 2D shapes	• Coloured whiteboard pens • 2D shapes • Music (optional)	• Check batteries • Set up algorithm example on Primo Board	• Primo Board template	• 2D shape names • Algorithm • Debugging • Prediction

Computational thinking concept



Algorithms

Computational thinking approach

Debugging

Teacher-led Introduction

1. Ask for volunteers to make different 2D shapes using their bodies (e.g. puff up face and make a big circle with their arms.)
2. Show the map and ask pupils where they can see a circle, triangle, square and rectangle. Ask: Can you see other shapes?
3. Explain that today the pupils will write algorithms to make Cubetto dance to places on the map that have different shapes in them. BUT there are problems with the algorithms!
4. Show the first algorithm example on the Primo Board and ask: Will this work? Why/why not? If not, how can we fix it?
5. Model pressing the action button (won't work). Model working out what is wrong and how you can fix it. Model trying again.
6. Explain that when we try to fix an algorithm that doesn't work, this is called debugging.
7. Ask: What is it called when we use what has happened before to tell us what will happen in the future? Recap prediction.

Lesson 5: Cubetto's Dance (2 of 2)

Creative Play

Draw a castle or boat using 2D shapes.

Guided Activity

1. Choose a place on the map that has circles in it: this is your starting point.
2. Choose another place on the map that has squares or rectangles in it (not too far away): this is your end point.
3. On the Primo Board and with coloured pens, write an algorithm to make Cubetto dance from the start to the end.
4. Check your algorithm works by testing it on Cubetto.
5. When you have made sure your algorithm works, rub out one of your blocks. It now needs debugging!
6. Find a partner who has finished their algorithm too.
7. Swap Boards and debug the algorithm.
8. Discuss with your partner what was missing and how you worked it out.

Independent Activity

1. Look at the first algorithm that needs fixing (e.g. Start on the boat, end on a triangle.) Ask: Where on the map can you see triangles? Where on the map does Cubetto want to get to? (In this example: start on the boat, end on the mountains).
2. Ask: Can you predict if this algorithm will work? If it won't work, what's wrong with it?
3. Ask: How can we debug this algorithm? Discuss until the group agrees on what to do.
4. Test out the algorithm with Cubetto.
5. Repeat for the other algorithm examples.

Challenge

Can you write an algorithm to make Cubetto dance in a circle?

Plenary and Assessment

1. Ask: What does debugging mean? What kinds of problems did we find today? How did we debug them?
2. Children share the algorithms they fixed (or created).
3. Ask: What was wrong? How did you work out what was wrong? How did you fix it?
4. Ask: What 2D shapes did we find on the map? What shapes can't we see?

Lesson 6: Cubetto's Maze Madness (1 of 2)

Cross-curricular Area: Maths: Position & Movement

NC Objectives	Outcomes	Resources Needed	Prep Needed	Resources Provided	Key Vocabulary
To debug simple programs	• I can predict what a simple algorithm will do • I can create the boundaries of a maze	• Pictures of simple mazes from above • Blank 4x4 grid on board • Coloured card	• Check batteries • Copy maze • Cut coloured card into strips. • Prepared algorithm for children to edit.	• Maze template	• Maze • Entrance/exit • Algorithm • Quarter turn • Bug/Debug • Direction • Route

Computational thinking concept

Logic

Computational thinking approach

Collaborating

Teacher-led Introduction

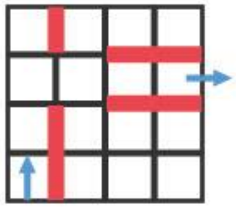
1. Show video of person walking through a maze: https://www.youtube.com/watch?v=tL3S-kzW_iU and ask: What is a maze? Have you ever been to a maze? Children to talk in groups and feedback.
2. Show pictures of mazes from above and explain that today pupils are going to create a maze for Cubetto!
3. Explain that a maze must have an entrance and exit.
4. Model creating a maze on a 4x4 grid on board using four lines, identify entrance and exit.
5. Ask: What do you think Cubetto is scared of? Draw a monster on the maze for Cubetto to avoid.
6. Model creating maze on real map using coloured card strips.
7. Ask: Can you help me write an algorithm to get to the end? Collect children's ideas and encourage pupils to come to front to create it.
8. Model using language of forward, backwards and ¼ turn left and right to describe Cubetto's route through the maze.

Lesson 6: Cubetto's Maze Madness (2 of 2)

Creative Play

Design a scary monster for your maze.

Guided Activity



1. Ask: Can you be sneaky and change my algorithm so Cubetto gets eaten by the monster? Cover your eyes while children change algorithm!
2. Run algorithm and tell children where you think the bug is (what they changed).
3. Recap the bugs they have identified: missing blocks, incorrect blocks, random buttons, incorrect functions and now someone changed the algorithm!
4. Tell children to work in pairs to use the coloured card and create their own maze.
5. When their maze is complete, ask children to create and test algorithm to get Cubetto through the maze.
6. Once tested, ask a partner to 'bug' the algorithm to send Cubetto into a monster then their partner debugs.

Independent Activity

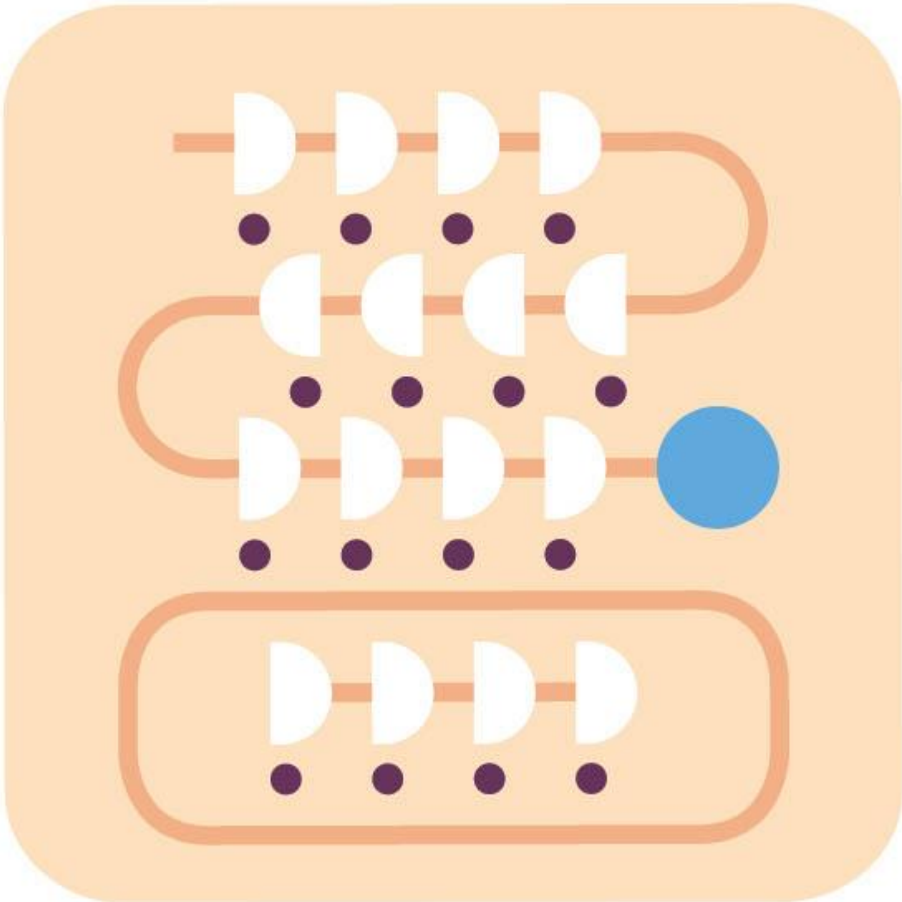
1. Look at the maze and find the start and the end.
2. Use your finger to trace a route through the maze. If you get stuck, go backwards and try another route.
3. When you have found a route, draw a line from start to finish.
4. You could add monsters to your maze!
5. Can you write instructions to tell someone how to complete the maze without looking?

Challenge

Can two Cubettos use your maze at the same time?

Plenary and Assessment

1. Ask volunteers to share their maze solutions with the class.
2. Ask pupils to come to front and show how they used the card to create the maze on the map.
3. Ask: How did your partner bug your algorithm? Why might someone add a bug?
4. Ask: Where else could you add a trap? What can you do if Cubetto gets stuck?



Appendix F

Preliminary Analyses Exploring Year (Chapter 5).

There were main effects of age overall (i.e., when not also exploring effects of Condition x Time), however these results will not be discussed further here as they were not relevant to the aims of this study (instead, see Appendix E).

Year Group

For each of the outcome variables, the effect of Year Group was tested as a main effect to investigate whether differences in task performance or pupil attitudes varied across year groups. Additionally, models including a three-way interaction of Condition x Time x Year group were tested. Results showed main effects of year group for some of the outcome measures. Post-hoc test results showed that the odds of a child performing better on the debugging task at post-test than at pre-test were significantly higher for children in Reception than those in Year 1 (odds ratio = 0.74, $z = -2.30$, $p = 0.05$) and children in Year 2 were more likely to perform better than those in Year 1 (odds ratio = 0.61, $z = -3.85$, $p = 0.0004$). Additionally, the odds of a child answering more trials correctly on the prediction task at post-test was significantly higher for children in Reception than those in Year 1 (odds ratio = 0.56, $z = -6.47$, $p < 0.0001$) and children in Year 2 were more likely to perform better than those in Year 1 (odds ratio = 0.75, $z = -4.18$, $p = 0.0001$). Finally, the odds of a child completing more trials correctly on the Lightbot Jr programming task at post-test was only significantly higher for children in children in Year 2 compared to those in Reception (odds ratio = 2.41, $z = -3.56$, $p = 0.001$).

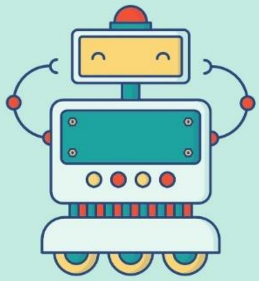
There was a marginal effect of year group on how fun children believed robots were at the end of the intervention ($B = -0.25$, $SE = 0.13$, $t = -1.93$, $p = 0.06$), but there was no main effect of year group on pupil's prediction accuracy scores, sequencing scores or their attitudes towards programming and their self-efficacy about using robotics. However, as year group and its interactions with Condition and Time were not statistically significant for any of the pupil measures (both CT related and attitudes), data was pooled across year groups. Furthermore, I had no theoretical reason to believe that intervention outcome would differ based on year group as this study was investigating individual pupil changes between pre- and post-intervention time points. Thus, analyses involving Year Group were not reported.

Appendix G

Workshop Slides Presented to Teachers in the Teacher Education Workshop (Intervention+ Condition, Chapter 6).

Our aims

- To investigate how children first learn computer programming and whether programmable robotics can be successfully integrated within the Foundation Phase.
    - To approach this from a teacher perspective as well as a child's perspective.



The journey so far...

- Exploring how teachers perceive educational robotics, and how they feel about using robotics inside the classroom.

- Focus group discussions – May 2021.
    - 13 foundation phase teachers.

- Online teacher attitudes survey – September 2021.
    - Approximately 60 foundation phase teachers.

The journey so far...

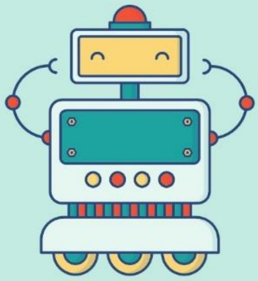
Key findings from our focus group and online survey:

- Lack of formal training for technological concepts like programming.
- Teachers want to know how they can integrate robotics with other classroom subjects.
- Training should focus on building teacher confidence with programming .
- Training should include both demonstrations and hands-on learning.

Programming in primary school: A growing movement... but why?

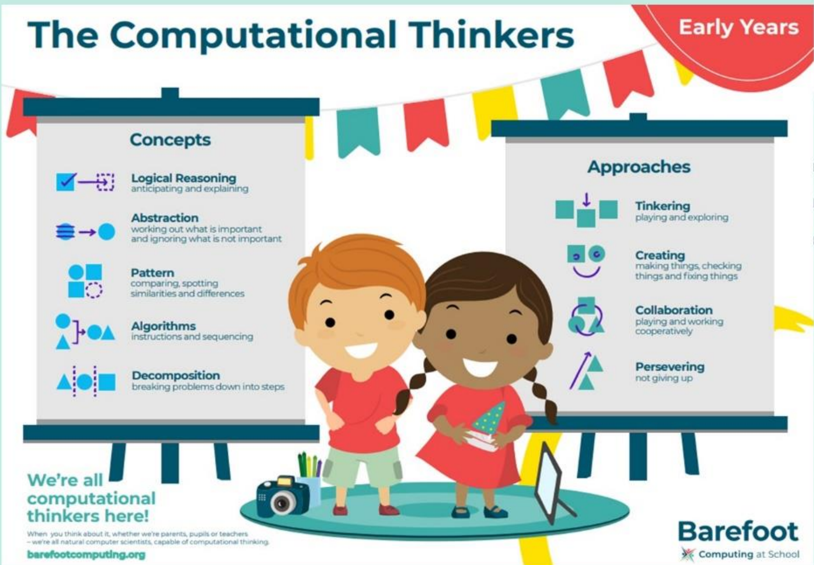
- Economic development – There has been a rise in demand for STEM skilled workers with demand now outweighing supply.
- Believed that learning programming at a young age benefits children's creative expression and skills like problem-solving.
- Encouraging children to become producers of technology, not just consumers.

- These are just a few reasons why UK governments have focused on preparing students with 21st century skills through STEM-related teaching, starting as early as primary school.



Computational thinking

"A set of problem-solving skills that we use in everyday life"



(Barefootcomputing.org)

Computer programming...

Often partnered with the term computational thinking.

When people hear the term "programming", they often think of some complex process that looks something like this...

Actually, programming can be made very simple, and research has shown that children as young as 4 years old can engage with and understand programming concepts.

You just need to use the correct tools!
Usually, tools that promote learning through play!

Computational thinking... something children only do on a computer?

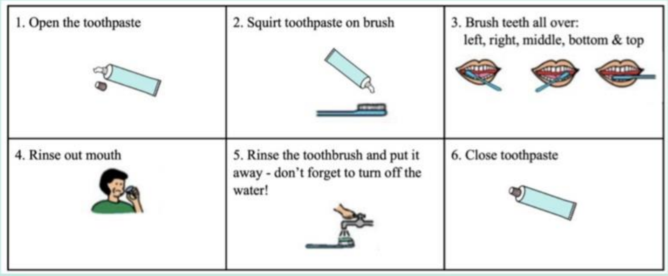
No.

We might use online activities now and then to practise some aspects of Computational Thinking skills, but in Early Years we can learn Computational Thinking (and aspects of programming) without computers.

This is called an 'unplugged' approach.

"Unplugged" examples

Algorithms – a series of ordered steps taken in sequence to solve a problem or achieve an end goal.



Activities like this reinforce the idea that order matters to make sure a process works as intended and that things go to plan.

"Unplugged" examples

Debugging – the two part act of (1) exploring a system for an issue for a "bug" that is causing a process to run incorrectly, and (2) working to resolve the bug.

It's Ava's first day of school and she's excited to dress up for the occasion. Ava picks out her favourite outfit for school and looks in the mirror. "Oh no, my socks don't match!" When she opens her sock drawer, she can't seem to find a pair that are the same. Every pair of socks in the drawer is mismatched! She takes all of the socks out of the drawer and places them on the bed and starts to sort them.

First, she puts all the white socks together, then all the red, blue, etc. She then notices that they are still not matched. She looks at the white socks and sees some have patterns and some are plain white, so she separates those. She does the same for the other colours. She pairs up all of the socks by colour and pattern only to find that two of the socks don't match each other or any others.

She calls her father over to ask what to do. Her father suggests going downstairs to check in the laundry. She goes downstairs, finds the socks and pairs them up. The problem is solved, and she remembers to change into a matched pair before she goes to school! They also agree to keep the socks balled together in the future to avoid this problem.

Methods of teaching programming: Computer based programs

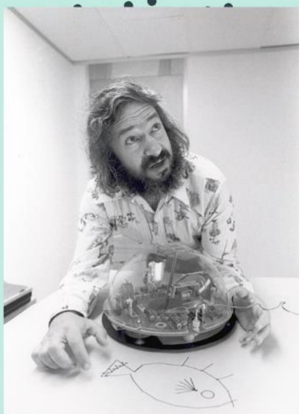
- Digital programs like Scratch allow children to 'snap' programming bricks together in different combinations to create their own animations.
- As you can see here, the original Scratch program assumes children have a basic knowledge of reading and writing.

Methods of teaching programming: Computer based programs

- Digital programs like Scratch allow children to 'snap' programming bricks together in different combinations to create their own animations.
- As you can see here, the original Scratch program assumes children have a basic knowledge of reading and writing.
- To make it more accessible for pre-literate children, Scratch Jr was developed which uses simplified programming commands.

Methods of teaching programming: Computer based programs

- Robotics combine accessible and age-appropriate materials to provide children with knowledge of programming through hands-on, practical experiences.
- They combine outcome visualisation and tangible learning in a digital world, whilst providing opportunities to integrate STEM with other disciplines, including music, literacy and art.



Coding as a playground

Programming languages (i.e., Scratch and other robotics programs) are "coding playgrounds".

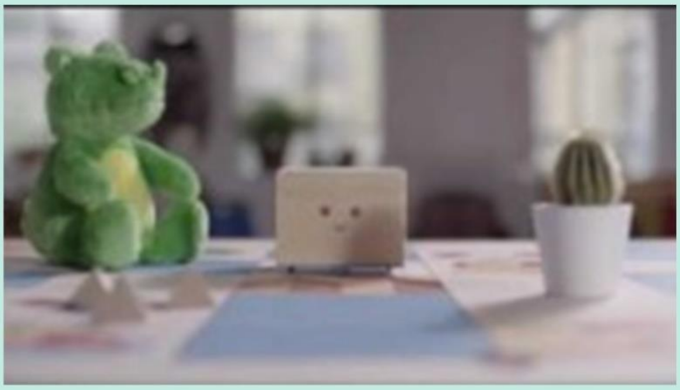
They promote:

- Problem-solving
- Imagination
- Cognitive challenges
- Social interactions
- Motor skills development
- Emotional exploration
- Decision making

THE CURRENT PROJECT

- Features Cubetto, a small wooden robot designed to teach programming to young children.
- A classroom-based robotics intervention for children, which includes support for participating teachers.

Introducing Cubetto



Primo Toys

- Coding = new literacy and should be prioritised from an early age.
- It's important to expose young children to basic principles of programming, even if they don't necessarily learn how to code.
- Using new technologies in the classroom can be overwhelming – so Primo aim to make it straightforward and easy!

How does Cubetto work?

Over to you!

Too much screen time?

Children can learn about technology without using a screen.

By removing written language and distractions such as screens and keyboards, Cubetto allows children to write and execute their first programs using the tangible block-based coding language.

Screen free learning can also support the development of socio-emotional skills – i.e., collaboration and teamwork.

Cubetto in the classroom

1. Introducing Cubetto
- Cubetto can't think on their own... we need to tell them what to do!

2. Introducing the board
- Cubetto can only move with a human's command.
- Like a remote control – "Have you used a remote control before?"

3. Introducing the blocks
- Different blocks = different commands.
- Can children guess what each token does?

Other tips:
- Repetition is key!
- Reflection – can children explain and vocalise why their instructions didn't work?

Cubetto lesson plans

- Created by Primo Toys, with support from early years teachers.
- Aims to make programming lessons fun and interactive
- Aims to make it easier for teachers to integrate the programming with other classroom subjects (i.e., maths and PHSE).
- Flexible and adaptable.

Appendix H

Introductory Cubetto Guide Given to Teachers (Chapter 6).

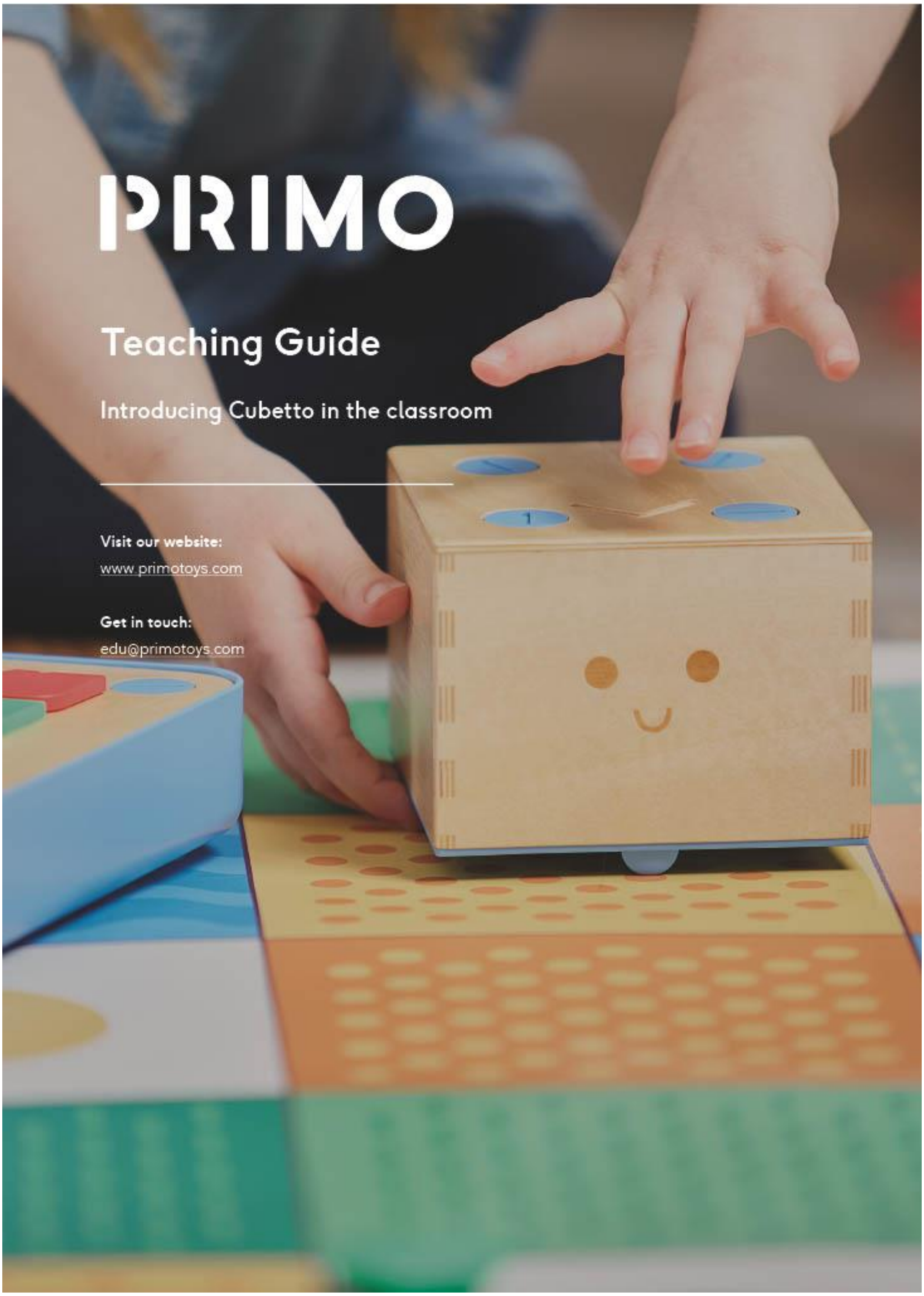


Table of content

How do I get started?

Hello!

The Cubetto Playset is a Montessori inspired coding toy that allows children ages 3 to 6 to program a friendly wooden robot without screens or literacy, and it is powered by a programming language you can touch.

What teachers love most is its versatility in cross curricular applications. It fosters student learning in key areas like Spatial Reasoning, Creative Thinking, Socio-Emotional Communication, Numeracy, Literacy. It's born to be the perfect STEM tool to use with your youngest students, and it is.

The activities contained in this guide were created by educators for educators. New technology can sometimes be overwhelming to understand and adopt. We want to make it simple for you to integrate the Cubetto Playset and its tangible programming language into your teaching.

Where can I use Cubetto?

Here at Primo Toys, we are great fans of unplugged activities. Many of ambassadors start with an open conversation on "what is a robot", showing them examples of true robots (such as Cubettos) and robot toys that look like robots but aren't. It's extremely beneficial for the children to learn as young as possible that a robot is a machine we can programme to execute instructions.
There are loads of inspiring unplugged activities most of which involve asking the children to program each other to perform a simple task like washing their hands, putting on a coat, clearing up a table.

Setting up your playset

You'll find instructions inside each playset that make setup fast and easy.

For more information, free lesson plans and resources, you can also visit: www.primotoys.com/education

Help and support

We're here to help, and you can contact us anytime.

Educators support and purchases: edu@primotoys.com

Research and value

The Cubetto Playset is a Logo Turtle inspired, Montessori led programming system. It is powered by a coding language you can touch, and an interface specifically designed for ages 3 to 6.

This age group is ideal to begin a child's journey in computer programming, but one should not have to do so at the expense of important educational areas traditionally learned in hands on play.

LOGO (not Lego), was a milestone in coding education. The goal of Seymour Papert, who created LOGO at MIT in the 1960's, was not just to teach programming, but also to help children discover their own personal way of solving problems.

Cubetto's coding blocks can be considered an extreme simplification of LOGO. We limited the instructions to their purest form, avoiding any kind of textual or numerical language. The material choice is important: the shell of the interface, and of Cubetto are made of wood, a natural material.

During development, observations were conducted in traditional Swiss kindergartens, where toys and games made of wood are the most loved by children.

Wooden toys are durable, they have memory, they collect history through marks and scratches, signs of past love and usage.

Wood was also chosen as a material because of the stark contrast it creates with technology, hiding the complexity of the circuit boards beneath the shell.

Coding and computational thinking concepts in early years

Hands on coding

The tangible Blocks have the potential and scalability of any real procedural programming language, and children can learn and play with a variety of core programming concepts.

The queue

Instructions in programs are executed following a precise order. On Cubetto's Board, they are put together following a line, also a physical representation of the queue.

Recursions

Create a subroutine by "packaging" a sequence in the function line, and call it in the queue with a blue block when you need it.

Algorithms

Algorithms are sets of precise instructions that form a program. Cubetto's Blocks are a physical representation of an instruction that combine to create a program.

Debugging

The instructions are laid on the Board. Fixing mistakes is as easy as swapping a block if Cubetto doesn't arrive where he needs to. This is called debugging.

Development and learning in other key areas

Beyond coding

The tactile and collaborative nature of Cubetto makes it an extremely versatile tool for the classroom. Cubetto fosters learning in key development areas that go beyond programming.

Communication

Children practice listening through a range of stories and narratives in relation to Cubetto, accurately anticipating key events and responding with relevant comments, questions or actions. They also develop their own narratives and explanations.

Physical excercise

Children master control and coordination in large and small movements around the playset. They negotiate the placement of obstacles around the world map, and place blocks on our tangible interface.

Social-Emotional

Children become confident by trying new, open ended activities that remove "wrong" outcomes, and easily encourage group work. The open nature of the maps allows them to choose the resources they need for their play session.

Mathematics

Children add and subtract blocks from a sequence. They solve problems, including doubling, halving and sharing to get Cubetto from A to B. They discuss size, identify shapes and patterns, distance, position, and time to solve problems.

Logical reasoning

The blocks allow children to create and debug simple programs with their hands. They use technology purposefully to create, organise, store, manipulate and retrieve meaningful sequences.

First things first

The goal is to get children to create programs for Cubetto by arranging sequences of instructions.

The more time spent playing with Cubetto, the more children develop computational thinking skills. This can be measured by observing them create longer sequences to solve and build more complex algorithms and problems. The speed with which a child can progress through "mission difficulty" varies from child to child, but it is always important to not skip the introductory steps, no matter how fast a child goes through them.

Introducing Cubetto

Introduce Cubetto as a friendly robot that children can program. Children should be told that Cubetto cannot think for himself, and can only move as programmed by the child, just like any other machine. It's fine at this point to have students sitting in a large circle around a map, and pass a couple of Cubettos around to one another, saying hello or acknowledging its presence. Doing so provides for a great opportunity for the teacher to ask open questions such as "how do you think it moves?" and contributes to create a powerful bond with it which help engage them in solving problems through narratives, which we will see later on.

Introducing the Board

It is important for children to understand Cubetto is only able to move with a human's command. This is not only empowering, but also key to understanding computing.

Introduce the Board as a sort of remote control that children can use to send instructions to Cubetto. Without the Board, there is no way of sending Cubetto his instructions..
Ask the children to explain what other objects in their homes and lives function with a similar paradigm.
A television needs a human to change its channels for example, or a washing machine needs a human to select its settings. All of these examples, like Cubetto, are machines that need human programming to work and do their job.

Introducing the Blocks

Different blocks represent different instructions, and an unambiguous, distinct command.
These blocks are what make up Cubetto hands-on coding language, and are key in the learning of computational thinking.

After the children have been given time to think about how to make Cubetto move and about its connection to the board, show them the blocks. Ask them to sort them, spot differences and guess how they work. This is a great way for them to learn about formulating an hypothesis and validating a theory, which is part of the exploratory process of writing code.

You can decide to introduce the instruction blocks formally as the directions Cubetto follows when inserted in the Board and sent by pressing the action button. When each block is inserted in the Board, a child should be encouraged to predict what Cubetto will execute before pressing the "Go" button.

Prediction is key in understanding concepts like program design, and it helps them develop abstraction.

Action causality

The aim of the first session is to introduce the simple notion that sending a command to Cubetto will result in an action. Take the Green Block (Forward), and have a child insert it in the first slot in the Board.

Then have the child press the big blue button on the Interface Board (Go), and observe Cubetto executing the command.
Make sure the child clearly associates the colour of the block with the action performed.

Unambiguous instructions

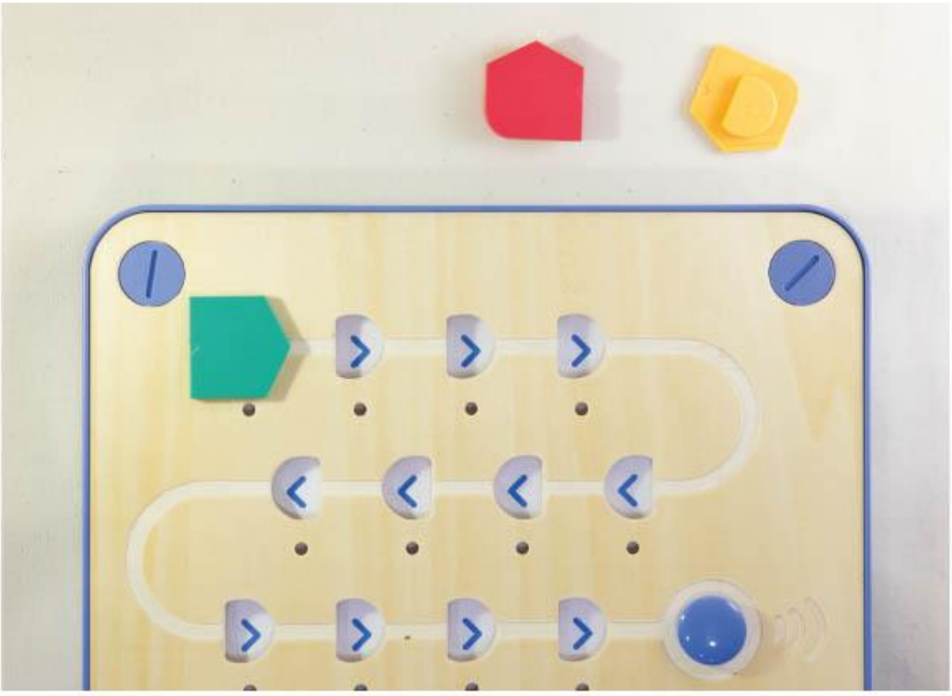
Repeat the previous step with each directional block (except the blue function block), until the children can confidently recognise each block as a distinct and unambiguous instruction.

This is an important step into understanding how a meaningful chain of commands, or a sequence if you will, can be later created to solve a specific problem.

Setting up your classroom

Communication skills are a fundamental, and sharing a Cubetto can give children lots of practice. We found small groups of 2-3 work best, up to 4 for older students who already have got the hang of sharing.

Aim to have as many children as many roles you can assign them. It's important they have a clear understanding of what is expected of them, how they should be working together and who is responsible for what.



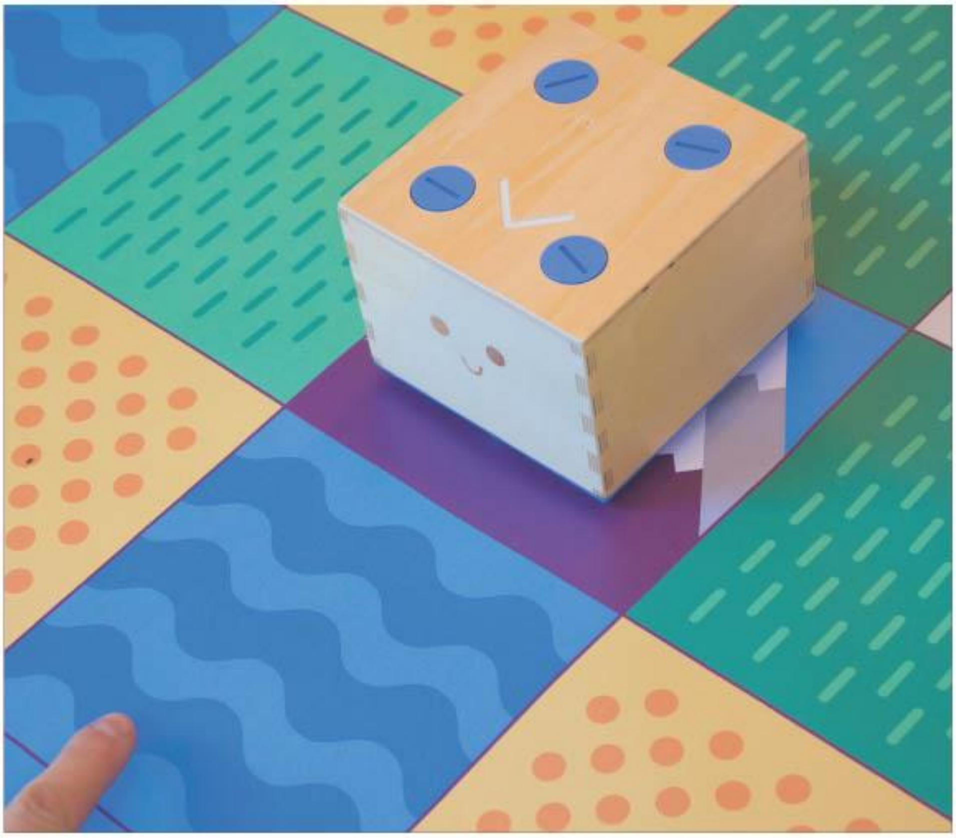
The first challenge

Unfold the map, and place Cubetto on a square. Ask the children to create a program that will get Cubetto to the square directly in front of it. They should be able to reason and tell which single instruction will allow Cubetto to reach his destination.

Let them insert the block in the first slot of the Interface Board, and press the action button.

Don't worry if the wrong block was selected. Just reset Cubetto's position, and encourage the children to reason their choice, voice out what didn't work and try new options.

One-block sequences are also a great way to help your children memorise different commands, so it's perfectly fine to give them a complex challenge for them to solve one block at time.



The queue

As your children will make progress and familiarise themselves with the idea of programming Cubetto, challenge them to compose longer and longer sequences, where the objective is to arrive to the objective with the directive of pressing the "Go" button only once.

As for the first challenge, let them operate the board autonomously. Give them plenty of time to examine the outcome and reason their choice of blocks and the result of that sequence.

Finally, encourage them to try new options.

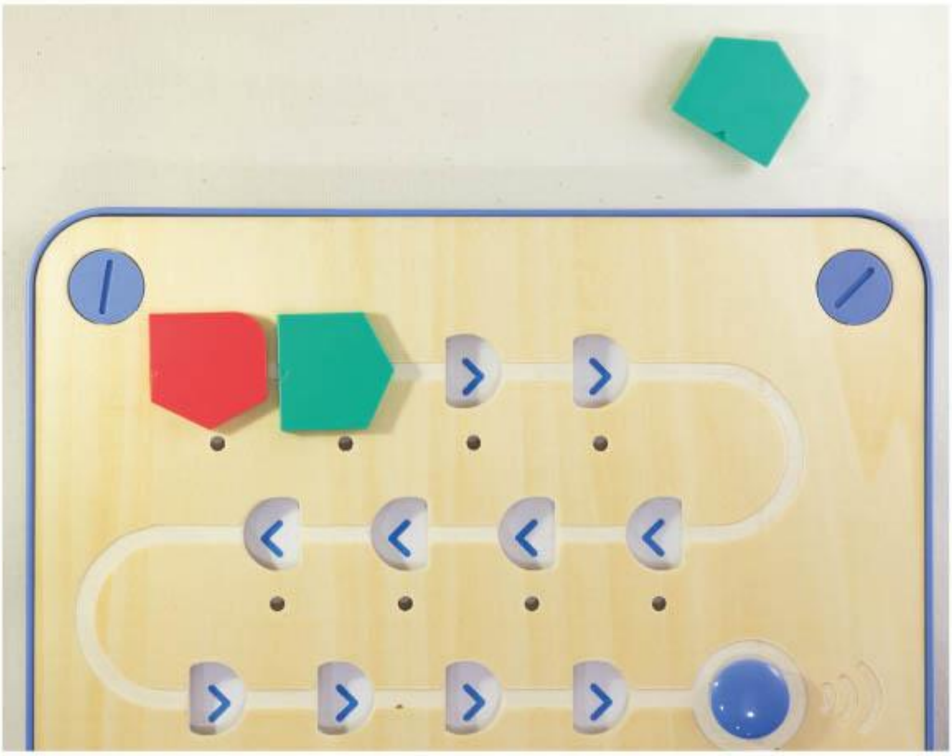
A good example is a L-shaped route, setting the arrival square one square ahead of Cubetto and one square to his left (or right). Let them reason and create the sequence that drives Cubetto to his destination.

Just reset Cubetto's position, and encourage the child to reason his/her choice, and try new options.

Debugging

Debugging is an integral part of learning to code.

As the challenges increase in difficulty your children will find that often they need to go back to their sequence and debug their code to obtain the desired result. The blocks will be a great help, as their algorithm is not only visible, it's tangible too.



Introducing the function

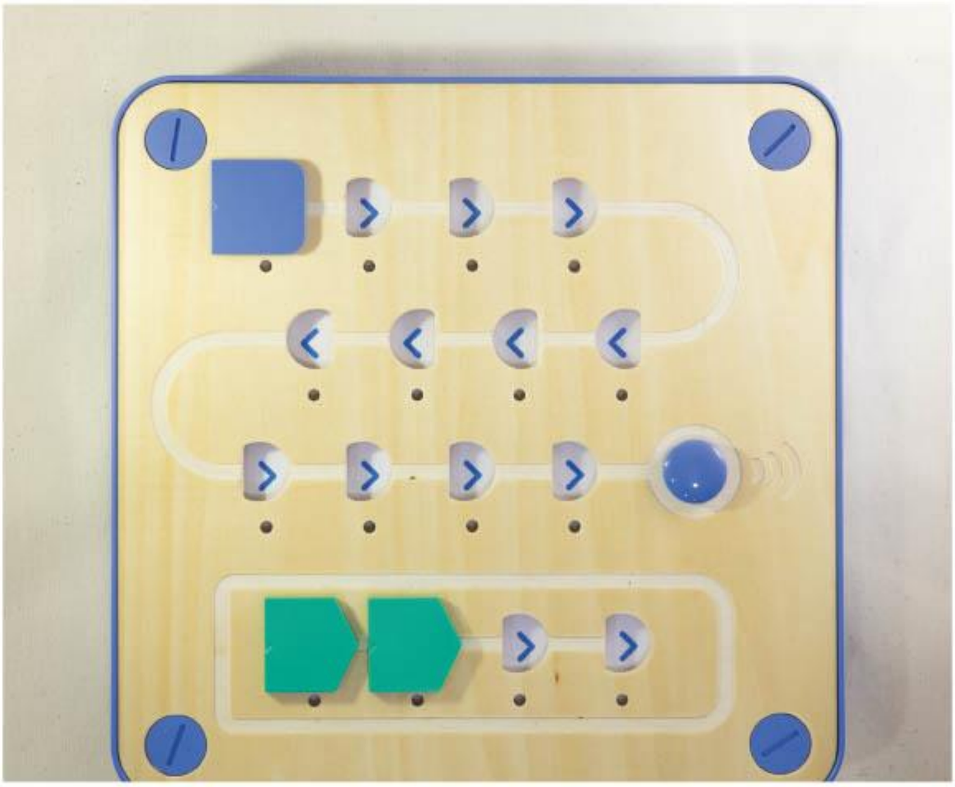
After the child is familiar with the basic blocks, and the idea of an algorithm, it is time to introduce the function block. To explain how it works, you can use the "../pack" metaphor, explaining that it is possible to pack more instructions inside a blue block.

To show this, first place two green blocks in the main sequence and press the Go button.

This will move Cubetto forward by two grid tiles on the map. Now clear the interface Board, and place the two forward blocks in the function line (The last line in the Board) instead, while placing a blue block in the main sequence.

Let the children observe that Cubetto can perform the same actions with two different sequences.

Leaving the same sequence in the function line, place two blue blocks in the main sequence, and let the children observe how two blue blocks cause Cubetto to repeat the function twice.

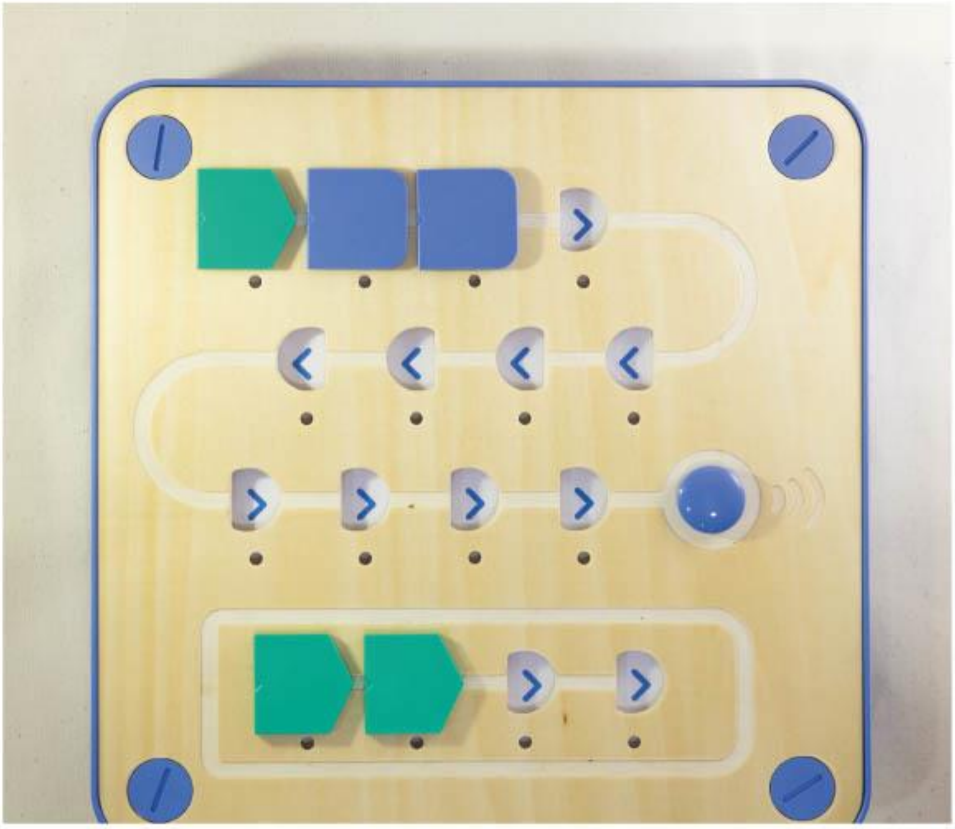


Solving problems with the function

Unfold the map, place Cubetto on the compass, and give the child only the following blocks: 3x Green blocks, and 2x Blue blocks (Function). Ask the child to create a program that will get Cubetto 5 squares ahead of him.

Since there are not enough forward blocks to make this happen, the child should be able to reason that a function should be created in order to complete the project.

Let the child create the right sequence, including a function, and press the Go button. If the sequence is wrong, just reset Cubetto's position, and encourage the child to reason his/her choice, and try new options.



Additional help and resources

Here is a list of very helpful resources to effectively implement coding with your youngest students:

- Code.org (https://code.org/)
- Coding as a Playground, Marina Umaschi Bers, Routledge, 2018
- Computing At School (https://www.computingatschool.org.uk/)
- Cubetto Playset Lesson Plans, Primo Toys Education, 2017
- Mindstorms, Seymour Papert, Harvester Press, 1980
- Robotics for Young Children, Ann Gadzikowski, Redleaf Press, 2017
- Teaching in the Digital Age, Brian Puerling, Redleaf Press, 2012
- Teaching in the Digital Age for Preschool and Kindergarten, Brien Puerling, Redleaf Press, 2018

Ultimately, The best judge of what makes a child tick is you. You know your classroom, your environment and your group, which is why we only provide a framework for progression instead of a ready made solution. It's your story, your classroom, your Cubetto.


PRIMO
Visit our website:
www.primotoys.com
Get in touch:
edu@primotoys.com

Appendix I

Beliefs Questionnaire Completed by Teachers (Chapter 6).

Information

You have been asked to complete this survey as you are taking part in our robotics project.

You will be asked to consider a number of statements. You should then report your response on a scale of 1 "Strongly Disagree" to 5 "Strongly Agree".

This survey will take around 10 minutes to complete.

If you have any problems completing the survey, please email Amy (Hughesaa1@cardiff.ac.uk).

Please click the button below to begin the survey.

Demographics

Your name:

School name:

Gender:

- Male
- Female
- Non-binary / third gender
- Prefer not to say

Age:

What year group(s) do you currently teach?

- ☐ Reception
- ☐ Year 1
- ☐ Year 2
- ☐ Other

Do you have any additional job roles?

- ○ Yes
- ○ No
- ○ I am unsure

Please provide details about your additional job roles

What year did you qualify as a teacher?

Years teaching experience:

What teaching qualification do you have?

- ☐ BEd
- ☐ PGCE
- ☐ PGDE

- [ ] HLTA Certificate
- [ ] Other ______

What is your primary degree?

______

Do you have any postgraduate certificates or degrees?

- ○ Yes
- ○ No
- ○ I am unsure

What postgraduate qualifications do you have?

______

Thinking about access to technology resources in your classroom, please indicate which of the following statements are relevant.

- [ ] Our pupil : device ratio is high.
- [ ] Our pupil : device ratio is low.
- [ ] My pupils have infrequent access to devices.
- [ ] My pupils have frequent access to devices.
- [ ] I'm happy with the number of devices we have access to.
- [ ] I wish we had access to more devices.
- [ ] I wish we had access to devices more often.
- [ ] I'm happy with how often we can use our devices.

Since September, have you taught programming or robotics to your class?

- ○ No

○ Yes

○ I am unsure

If yes, please tell us a bit about how you have taught programming or robotics in the classroom.

Value beliefs

Please consider the following statements:

	Strongly Disagree 1	2	3	4	Strongly Agree 5
I think that programming should be included in primary education as early as possible.	○	○	○	○	○
More programming should be taught in the early childhood classroom.	○	○	○	○	○
Because programming education is so important in primary school, I think that inexperienced teachers should receive additional training in this area.	○	○	○	○	○
Programming-related activities help improve pupils' math skills.	○	○	○	○	○
It is not appropriate to introduce programming to children at an early age.	○	○	○	○	○
Programming-related activities help improve pupils' language skills.	○	○	○	○	○
Young children cannot learn programming until they are able to read.	○	○	○	○	○
Programming related activities are too difficult for young children.	○	○	○	○	○

	Strongly Disagree 1	2	3	4	Strongly Agree 5
Programming related activities help improve pupils' social skills.	O	O	O	O	O
Young children are curious about programming concepts	O	O	O	O	O
I think that most primary school teachers find programming content to be a difficult subject to teach	O	O	O	O	O
I think that most elementary school teachers find the topics that come up in programming class complicated.	O	O	O	O	O
Experimenting hands-on with materials and objects is how young children learn best.	O	O	O	O	O
Programming-related activities help improve pupils' approaches to learning.	O	O	O	O	O

Future impact

Please consider the following statements...

	Strongly Disagree 1	2	3	4	Strongly Agree 5
I think that programming education is essential for helping primary school students become more involved with society's problems.	O	O	O	O	O
I believe that programming education is essential for primary school children's development.	O	O	O	O	O
I believe that programming education in the primary school is essential for students to be able to make good educational and career choices.	O	O	O	O	O
Programming activities help foster children's interest in programming in later grades.	O	O	O	O	O

Gender

Please consider the following statements...

	Strongly Disagree 1	2	3	4	Strongly Agree 5
I believe that boys in primary schools are more enthusiastic about programming than girls.	○	○	○	○	○
I think that in primary schools, boys are more likely than girls to choose assignments concerned with programming.	○	○	○	○	○
I think that I would unconsciously be more likely to choose a boy for a programming demonstration than a girl.	○	○	○	○	○
I think that in primary schools, male teachers experience more enjoyment in teaching programming than female teachers.	○	○	○	○	○

Individual feelings

Please consider the following statements...

	Strongly Disagree 1	2	3	4	Strongly Agree 5
Teaching programming makes me cheerful.	○	○	○	○	○
I feel happy while teaching programming.	○	○	○	○	○
I feel enthusiastic when teaching programming.	○	○	○	○	○
I enjoy teaching programming	○	○	○	○	○
I feel nervous while teaching programming.	○	○	○	○	○
I feel tense while teaching programming.	○	○	○	○	○
Teaching programming makes me anxious.	○	○	○	○	○

	Strongly Disagree 1	2	3	4	Strongly Agree 5
I feel stressed when I have to teach programming.	○	○	○	○	○
I enjoy doing programming activities with my pupils.	○	○	○	○	○

Self-efficacy

Please consider the following statements...

	Strongly Disagree 1	2	3	4	Strongly Agree 5
I have enough programming content knowledge to teach this subject well in primary school.	○	○	○	○	○
I do not have enough programming knowledge to teach programming to young children.	○	○	○	○	○
I feel uncomfortable using programming tools such as robots, and coding apps when teaching science lessons.	○	○	○	○	○
I am able to deal effectively with questions from students about programming.	○	○	○	○	○
I think I can succeed in helping primary students reach a solution during assignments about programming.	○	○	○	○	○
I feel comfortable planning and demonstrating classroom activities related to programming topics (e.g., algorithms, debugging)	○	○	○	○	○
I feel comfortable doing programming activities in my early childhood classroom.	○	○	○	○	○
I am afraid that children may ask me a question about programming principles that I cannot answer.	○	○	○	○	○
Planning and demonstrating hands-on programming activities is a difficult task.	○	○	○	○	○

Barriers

Please consider the following statements...

	Strongly Disagree 1	2	3	4	Strongly Agree 5
For me, the availability of a ready-to-use existing package of materials (e.g. robotics kits) is an essential prerequisite for being able to teach programming in class.	○	○	○	○	○
For me, the support of my colleagues is decisive for whether or not I will teach programming in class.	○	○	○	○	○
Given other demands, there is not enough time in a day to teach programming.	○	○	○	○	○
Preparation for programming teaching takes more time than other subject areas.	○	○	○	○	○
I do not have enough materials to do programming activities.	○	○	○	○	○

Teaching practices

Please consider the following statements...

	Strongly Disagree 1	2	3	4	Strongly Agree 5
I discuss ideas and issues of programming teaching with other teachers.	○	○	○	○	○
I use all kinds of classroom materials (e.g., robotics, apps, computer programs, paper materials) for programming activities.	○	○	○	○	○
I use resource books to get ideas about programming activities for young children.	○	○	○	○	○
I use the internet to get ideas about programming activities for young children.	○	○	○	○	○

	Strongly Disagree 1	2	3	4	Strongly Agree 5
I get ideas for hands-on activities from what my pupils do, say, and ask.	○	○	○	○	○
I include some books about robots during story time.	○	○	○	○	○
I do not have enough materials to do programming activities.	○	○	○	○	○
I make an effort to include some programming activities throughout the week.	○	○	○	○	○

Powered by Qualtrics

Appendix J

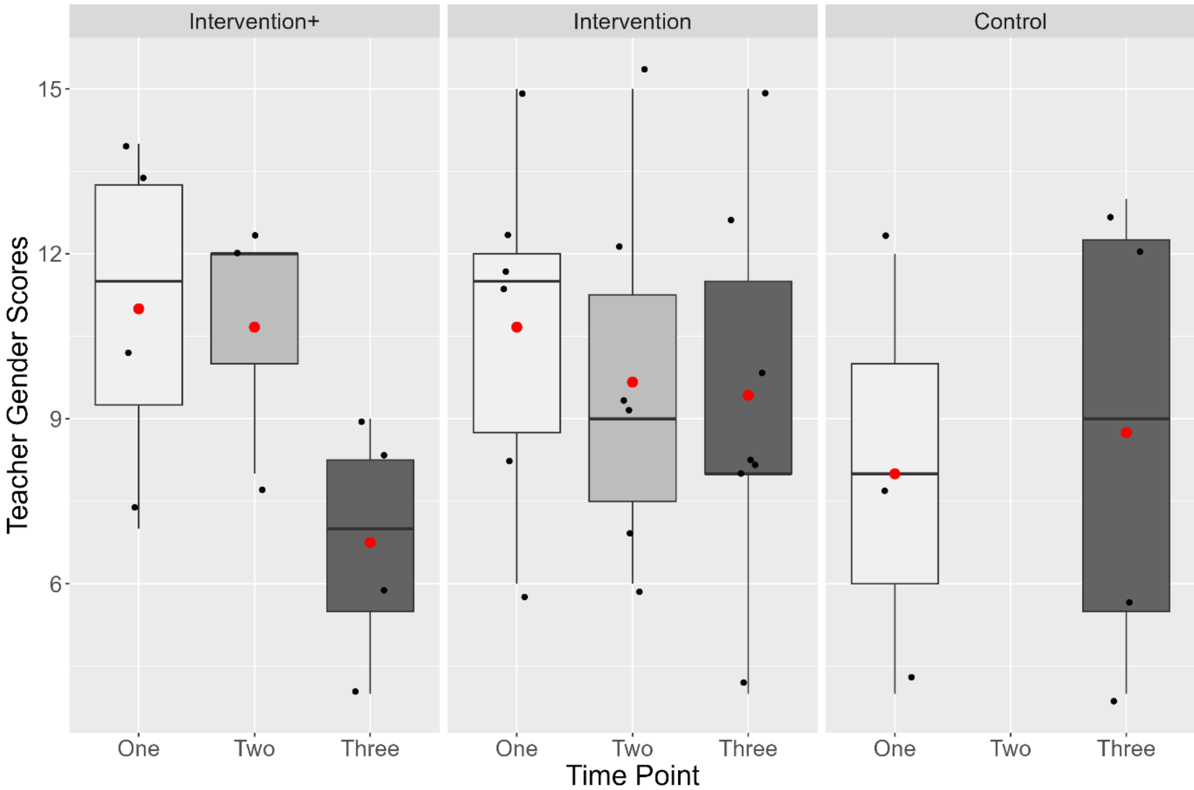
Analysis of Teacher's Beliefs Across the Robotics Intervention (Chapter 6).

Gender Beliefs

Figure 1 illustrates teachers' gender beliefs scores from the administered questionnaire at three time points, grouped by condition.

Figure 1

Teacher Gender Beliefs' Scores, Grouped by Condition and Time Point.



Note: Group means at each time point are shown in red.

A Linear Mixed-effect Model (LMM) from the *lme4* package in R (Bates et al., 2015) was used to analyse group differences across time points. The current model included teacher gender scores as an ordinal outcome variable. The interaction between Condition (Intervention+, Intervention and Control) and Time (pre-workshop, post-workshop, and post-intervention) was included as a fixed effect and observations were grouped by Teacher. Confidence intervals were computed with the *confint()* function. P values were obtained using the *lmeTest* package. The model that was estimated used the following structure:

Teacher Gender Score ~ Condition*Time + (1|TeacherID).

Table 1

Teacher Gender LMM Model Results: Fixed Effects.

	Model summary					
	β	*SE*	*t*	CI 95%	*p*	
Intercept	8.96	1.92	4.66	[5.43, 12.53]	< 0.001	
Intervention+ Vs Control	2.64	2.58	1.02	[-2.16, 7.38]	0.32	
Intervention Vs Control	0.89	2.35	0.38	[-3.53, 1.22]	0.71	
Time	-0.10	0.81	-0.13	[-1.60, 1.51]	0.90	
Intervention+ vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	-2.02	1.08	-1.88	[-4.15, -0.01]	0.08	.
Intervention vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	-0.21	0.99	-0.21	[-2.23, 1.61]	0.84	

Note: **.** $p < 0.1$

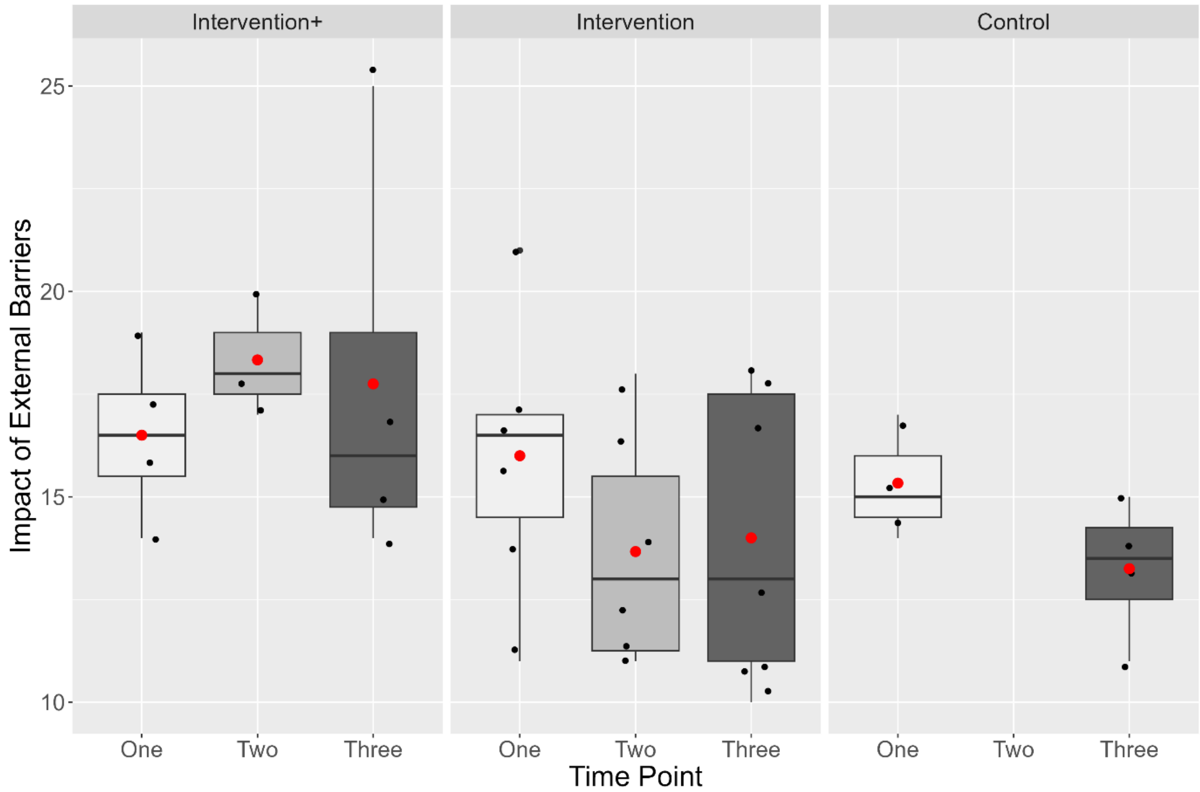
The fixed effects from the model results are described in Table 1. The model revealed no significant interaction effect of Condition x Time on teachers' gender scores. Results illustrate that there were no significant differences in reductions in gender belief scores over time, between the Intervention+ and Control groups, and the Intervention and Control groups. Similarly, there was no significant difference within the Intervention+ and Intervention conditions (B = -1.82, SE = 0.92, t = -1.98, CI 95% [0.02, 3.54], p = 0.06).

Perceptions of External Barriers

Figure 2 illustrates teachers' beliefs about how external barriers may limit teaching of programming and robotics as measured by a questionnaire.

Figure 2

Teachers' Perceptions of External Barriers, Grouped by Condition and Time Point.



Note: Group means at each time point are shown in red.

An LMM was used to analyse group differences across time points. The model included teacher external barrier scores as an ordinal outcome variable. The interaction between Condition and Time was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following model structure:

Teacher External Barrier Score ~ Condition*Time + (1|TeacherID).

Table 2

Teacher External Barriers LMM Model Results: Fixed Effects.

	Model summary				
	β	*SE*	*t*	CI 95%	*p*
Intercept	15.29	1.71	8.96	[12.16, 18.42]	< 0.001
Intervention+ Vs Control	1.51	2.26	0.67	[-2.63, 5.66]	0.51
Intervention Vs Control	-0.10	2.07	-0.05	[-3.92, 3.70]	0.96
Time	-1.02	0.85	-1.21	-2.62, 0.58]	0.24

Intervention+ vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	1.65	1.14	1.45	[-0.50, 3.80]	0.16
Intervention vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	0.21	1.04	0.20	[-1.78, 2.16]	0.84

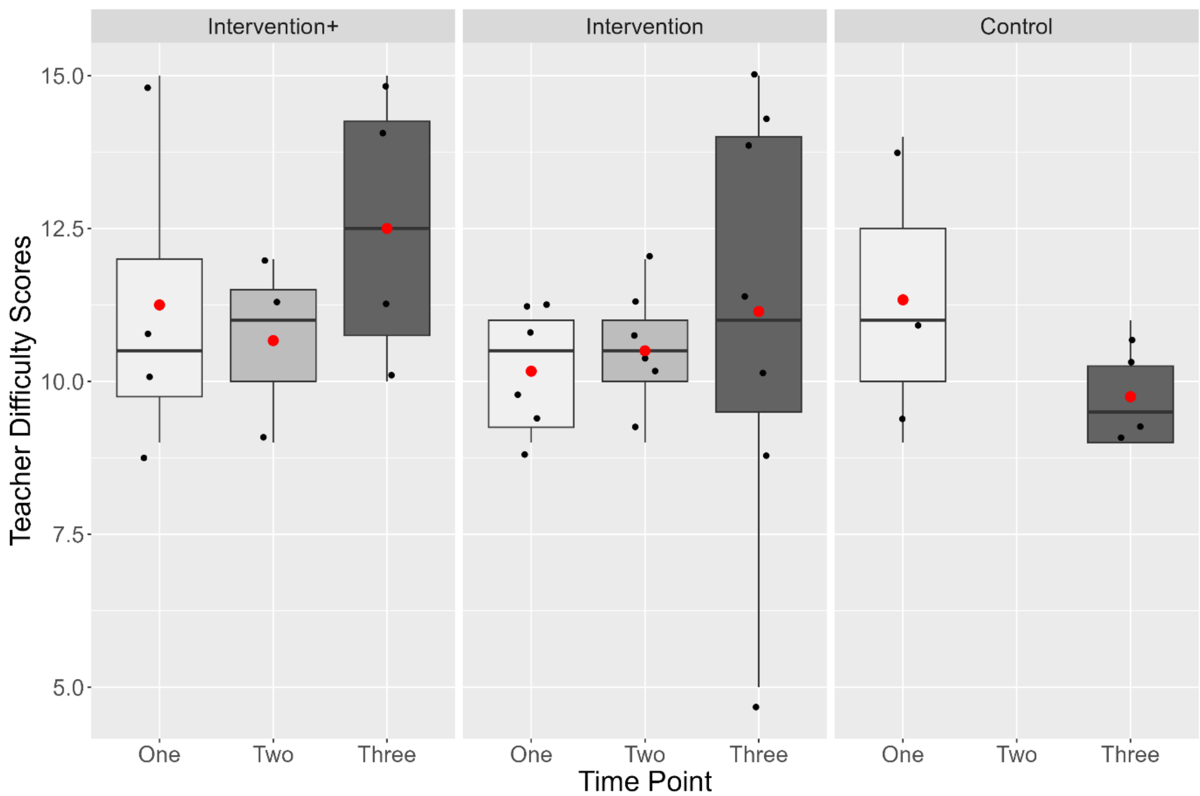
The fixed effects from the model results are described in Table 2. The model revealed no significant interaction effect of Condition x Time on teachers' external barrier scores. Results illustrate that there were no significant differences in reductions in beliefs about external barriers over time, between the Intervention+ and Control groups, and the Intervention and Control groups. Similarly, there was no significant difference within the Intervention+ and Intervention conditions (B = 1.43, SE = 0.97, t = 1.48, CI 95% [-0.38, 3.31], p = 0.15).

Difficulties Teaching Programming

Figure 3 illustrates teachers' beliefs about how difficult it is to teach programming and robotics, as measured by a questionnaire.

Figure 3

Teachers' Difficulty Scores, Grouped by Condition and Time Point.



Note: Group means at each time point are shown in red.

An LMM was used to analyse group differences across time points. The model included teacher difficulty scores as an ordinal outcome variable. The interaction between Condition and Time was included as a fixed effect and observations were grouped by Teacher. The model that was estimated used the following model structure:

Teacher Difficulty Score ~ Condition*Time + (1|TeacherID).

Table 3

Teacher Difficulty LMM Model Results: Fixed Effects.

	Model summary					
	β	*SE*	*t*	CI 95%	*p*	
Intercept	11.56	1.31	8.84	[9.17, 13.97]	<0.001	
Intervention+ Vs Control	-0.43	1.74	-0.25	[-3.62, 2.75]	0.81	
Intervention Vs Control	-2.09	1.59	-1.31	[-5.10, 0.88]	0.20	
Time	-0.91	0.63	-1.44	[-2.09, 0.35]	0.17	
Intervention+ vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	1.53	0.84	1.82	[-0.14, 3.14]	0.09	.
Intervention vs Control (Pre-workshop vs Post-workshop vs Post-intervention)	1.68	0.77	2.18	[0.03, 3.10]	0.04	*

Note: . $p < 0.1$, * $p < 0.05$.

The fixed effects from the model results are described in Table 3. The model revealed a significant interaction effect of Condition x Time on teachers' difficulty scores. On average, improvements in teacher difficulty scores in the Intervention condition were significantly larger than in the Control group ($p = 0.04$). Improvements in teacher difficulty scores in the Intervention+ condition were not significantly different from those in the Control condition ($p = 0.09$). When comparing both intervention groups, improvements in relevance scores in the Intervention+ condition did not differ significantly from those in the Intervention condition, ($B = 0.15$, $SE = 0.72$, $t = 0.21$, CI 95% [-1.36, 1.49], $p = 0.83$).