Understanding the Parental Involvement in Their Children's Programming Learning

Janice Jianing Si · University of Macau, China · *janice.sijianing@connect.um.edu.mo*
Xin Lin · University of Macau, China · *xinlin@um.edu.mo*
Alicia Wanyi Liu · University of Macau, China · *mc44156@um.edu.mo*
Shuhao Huang · Anhui Normal University, China · *huangshuhao@ahnu.edu.cn*
Yujun Zhu · Anhui Normal University, China · *zhuyujun@ahnu.edu.cn*
Caishi Huang · University of Macau, China · *cshuang@um.edu.mo*
Yunlong Mao · Nanjing University, China · *maoyl@nju.edu.cn*
Xiaobo Zhou · University of Macau, China · *waynexzhou@um.edu.mo*
Kanye Ye Wang* · University of Macau, China · *wangye@um.edu.mo*

Abstract. Parents play a critical yet understudied role in children's programming education. While prior research has focused on learners and educators, little is known about how parents support, evaluate, and sustain their involvement. In this study, we conducted semi-structured interviews with 26 parents in China whose children had varied programming learning experiences. Through thematic analysis, we identify two key dynamics of parental involvement: (1) parents primarily contribute through indirect forms of support—such as emotional encouragement, logistical coordination, and resource provision—rather than direct technical assistance; (2) parental involvement evolves dynamically over time, shaped by access to information, children's feedback, and social pressures. Our findings reveal that parental involvement is a dynamic process involving infrastructuring, meaning-making, and negotiation. We offer a novel perspective by reframing parents as collaborative actors in sociotechnical learning ecologies and provide design implications for supporting parents in scaffolding their children's programming education over time.

Keywords. programming education, children programming, parental perception, parental involvement

1 Introduction

In the 21st century, programming has moved beyond a professional skill to become a foundational literacy, akin to reading and writing (Kafai and Burke, 2014; Resnick and Siegel, 2015). Reflecting this shift, many education systems have integrated programming into K-12 curricula (Falkner et al., 2014; Nouri et al., 2020). Programming tools (Sheriff et al., 2017; Si et al., 2026; Yu et al., 2020) and learning guidebooks (Payne, 2015; Toys, 2017) have also been developed to support children's self-directed practice. Alongside these developments, more children are

* Corresponding author

Janice Jianing Si, Xin Lin, Alicia Wanyi Liu, Shuhao Huang, Yujun Zhu, Caishi Huang, Yunlong Mao, Xiaobo Zhou, Kanye Ye Wang. 2026. Understanding the Parental Involvement in Their Children's Programming Learning. In: *Proceedings of the 24th EUSSET Conference on Computer-Supported Cooperative Work (ECSCW) – Conference Papers*, Reports of the European Society for Socially Embedded Technologies. ISSN: 2510-2591 https://doi.org/10.48340/ecscw2026_065

learning programming from an early age (Sáez-López et al., 2016), often with varying degrees of parental involvement (Govind et al., 2020; Lin and Liu, 2012).

Parental involvement is widely recognized as a crucial factor in children's educational success, **including supporting study activities and evaluating performance** (Castro et al., 2015; Robinson and Harris, 2014; Sui-Chu and Willms, 1996). It influences not only academic outcomes, but also motivation, learning attitudes, and long-term development (Epstein, 1991; Stevenson and Baker, 1987). For example, sustained involvement is associated with achievement-oriented mindsets (Cai and Wong, 2022; Frome and Eccles, 1998; Turner and Lapan, 2002), while negative interactions can contribute to disengagement and lower performance over time (Huesmann et al., 1987; Stipek, 2001).

Despite its importance, parental involvement in programming education remains underexplored and narrowly framed. Existing studies often examine involvement in discrete programming events, such as robotics camps (Porras et al., 2019) and staged programming kits (Elkin et al., 2016; Yu et al., 2020). This work typically emphasizes parents as technical collaborators, assisting with programming tasks or tool use (Bresnihan et al., 2020; Sheehan et al., 2019; Turner and Lapan, 2002). Other studies assess the value of parental involvement through family workshops in programming or STEM, documenting how parents structure activities, guide exploration, interpret technical explanations, support emotional regulation, and negotiate goals and meanings during shared creative computing and STEAM practices (Erete et al., 2021; Fields et al., 2015; Yu et al., 2023). While valuable, these accounts remain partial, capturing snapshots of engagement tied to particular activities, tools, or instructional settings.

Even when they reveal how parents provide technical support in specific contexts, prior work rarely examines how parents stay involved across the broader trajectory of children's programming learning. We know little about how parents support learning beyond isolated events, whether they take roles beyond technical assistance, how they evaluate progress across stages and contexts, and what challenges emerge as involvement unfolds—all of which shape the sustainability of involvement. This narrow view also limits the design of programming education that can adapt to children's changing needs and support meaningful experiences for both children and parents. To address this gap, we examine parental involvement as reflected across children's broader programming learning journeys, rather than within single events. To guide our investigation, we propose the following research questions (RQs):

RQ1: How do parents support and evaluate their children's programming learning through their overall involvement?

RQ2: What challenges do parents encounter when involved in their children's programming learning?

We conducted semi-structured interviews (n = 26) with parents whose children had diverse programming experiences to capture retrospective accounts of parental involvement across broader programming learning journeys. Through thematic analysis, we found that parents supported children's programming learning in multiple ways, including helping with reading materials and organizing self-directed practice at home, relying on external courses when time or expertise was limited, and discussing real-world applications and technology careers. When evaluating progress, parents attended to emotional engagement, academic achievements, computational concept mastery, practical ability, and educator feedback. However, most parents were only loosely involved, primarily selecting resources rather than providing sustained guidance; only a few maintained consistent support and holistic evaluation. We also identified a progression of challenges that shaped involvement, where limited access to resources constrained support capacity and, over time, contributed to declining motivation. Building on this understanding of sustained parental involvement, we further discuss how the technical and long-term nature of programming education reshapes parents' roles and engagement, and we derive design implications for educational resources and sociotechnical systems that better support parents' evolving needs and constraints.

This paper makes two contributions to Computer-Supported Cooperative Work (CSCW) research on learning, education, and families. First, we offer new insights into parents' involvement across children's broader programming learning journeys, extending prior work that has mainly focused on isolated learning activities. We show that parental involvement is not limited to direct instruction, but also includes articulation work through which parents coordinate courses, tools, schedules, teacher feedback, peer information, and emotional support to sustain children's learning over time. Second, we surface challenges of parental involvement from parents' perspectives, revealing barriers that child-centered accounts often miss, including limited technical expertise, fragmented information sources, and uncertainty in making learning-related decisions. Together, these findings inform the design of collaborative learning systems and programming resources that better support sustained and low-barrier parental involvement.

2 Related Work

2.1 Parental Involvement Regarding Children's Programming Learning

Parental involvement is widely recognized as a key factor in children's educational success (Avvisati et al., 2010). It is associated not only with academic achievement, learning motivation, and self-efficacy, but also with emotional development and long-term learning trajectories (Castro et al., 2015; Pezdek et al., 2002; Robinson and Harris, 2014). Prior work largely falls into three areas: the value and impact of involvement, the roles and practices parents adopt, and tools or activities designed to support such involvement.

The first strand examines the value of parental involvement in children's coding learning. Studies suggest parental involvement supports children's engagement, exploration, and understanding of computational concepts. For example, Yu et al. (2021) linked supportive behaviors (e.g., preparing materials, guiding exploration, creative scaffolding) to sustained attention and conceptual mastery with coding kits. Parent–child interaction in programming apps can also improve concept learning while fostering motivation, confidence, and persistence (Govind et al., 2020; Sheehan et al., 2019).

The second strand focuses on parental roles in programming activities. Yu et al. (2020) identified roles such as "designer of learning activities", "resource finder", "manager", and "teacher", showing that parents contribute beyond technical help by structuring tasks and managing resources. Building on parental mediation theory, Yu et al. (2021) further described parents as "co-explorers" and "resource preparers". Additionally, family coding event studies also highlight parents' emotional scaffolding, such as encouragement during frustration and celebration of milestones (Govindarajan, 2019).

The third strand investigates tools and activities explicitly designed to support parental engagement in coding education. Family workshops often position parents as co-learners and enable joint exploration (Bresnihan et al., 2019; GOVIND and BERS, 2020; Govindarajan, 2019). Toolkits and family-oriented platforms provide accessible entry points into children's coding practices (Druga et al., 2022a), and HCI systems propose parent-configurable features that lower expertise barriers (e.g., text-free and tangible designs for families with English literacy challenges) (Banerjee et al., 2018).

Despite these contributions, key gaps remain. Most studies examine parental involvement in bounded settings (e.g., workshops or short deployments lasting days or weeks) (Druga et al., 2022a). We still know little about how involvement unfolds over longer periods, how roles shift as children progress, or how families sustain engagement as challenges grow. This limited view constrains designs that match families' real trajectories of involvement. Therefore, we propose our first RQ.

2.2 Perceived Challenges in Children's Education

Parental involvement is widely seen as beneficial, yet it is also shaped by the challenges parents face while supporting their children. A growing body of work surfacing these challenges is essential for designing supports that sustain engagement (Coleman and Churchill, 1997; Erlangga, 2022). Researchers have therefore examined the obstacles parents report and how design interventions can help reduce them.

Prior studies document many barriers that constrain effective involvement. Parents often report low confidence in subject matter knowledge (Badrasawi et al., 2019), and additional constraints include socioeconomic status, language barriers, immigration policies, and other contextual factors (Malone, 2017; Turney and Kao, 2009; Vera et al., 2012). In technology-related domains such as programming, these issues can be amplified. Yu et al. (2021) show that parents using coding kits struggle with technical setup and may hesitate to scaffold without taking over the child's task. Observational studies of family coding workshops report similar tensions: parents are eager to participate but feel unprepared to debug or explain block logic in accessible terms (Govindarajan, 2019).

At the same time, researchers are exploring design approaches to mitigate these challenges. Some work focuses on tools (e.g., dashboards and visual trackers) that help parents understand children's learning progress (Humphrey et al., 2021). Systems like StoryBuddy (Zhang et al., 2022) show how configurable conversational agents can support engagement even when parents are not present. Ho et al.'s SET-PAiREd (Ho et al., 2025) illustrates that giving parents flexible control over AI-assisted robots can improve learning outcomes and increase parent confidence. Other efforts emphasize inclusive family communities, while family toy kits aim to lower entry barriers and broaden parental participation (Cagiltay et al., 2024; Druga et al., 2022b).

Despite this progress, we still know little about the challenges parents face when supporting children's programming learning over longer periods. Most studies capture short-term issues in workshops or prototype deployments, such as setup friction or brief hesitation during debugging. Less clear is how challenges evolve over time: whether early barriers diminish with experience, whether new tensions arise as projects become more complex, and whether persistent difficulties erode parents' willingness to stay involved. This lack of longitudinal understanding limits our ability to design supports that match parents' real trajectories of involvement, motivating our second RQ.

3 Methodology

3.1 Ethical Considerations

This study has been approved by the Institutional Review Board (IRB) at the University of Macau. We did not collect personally identifiable information from our participants. During the interviews, we ensured that participants were aware of their right to decline to answer any questions without any impact on their participation compensation. The responses we cite in our findings are meant to illustrate patterns of user behavior, not unique or potentially identifiable circumstances of any individual.

3.2 Recruitment and Demographics

We adopted a mixed-method recruitment approach, integrating both direct recruitment and snowball sampling techniques (Goodman, 1961; Noy, 2008; Parker et al., 2019). We disseminated interview recruitment information through various online social platforms, including WeChat public accounts, WeChat group chats, and Douyin, all of which have a user base of hundreds of millions[1]. Through the email provided in the recruitment information, we received contact from a total of 25 parents interested in our study. Moreover, we inquired if the directly recruited participants could share our recruitment information within their networks, leading to an additional 5 participants across two snowball sampling chains. These potential participants first completed a screening survey to determine their involvement in their children's programming learning, detailed in Supplementary Appendix A. This process resulted in the exclusion of 4 parents who had only heard of children's programming but had no involvement experience. Ultimately, we interviewed 26 who met our criteria. Table 1 summarizes the participants' demographics.

3.3 Semi-structure Interview Procedure

Interviews were conducted between July and August 2024 via face-to-face meetings. Each interview lasted 45–70 minutes, and participants received a $15 gift as compensation. All interviewees were informed that interviews would be recorded and that their statements might be quoted anonymously in the final report. All data were treated as confidential. Our interview includes three parts:

In the first part, we collected demographic information (e.g., age, occupation) and basic details of children's programming learning, including learning duration, training experiences, and programming languages learned.

[1] https://www.demandsage.com/wechat-statistics/; https://www.statista.com/statistics/1361354/china-monthly-active-users-of-douyin-chinese-tiktok/

Table 1. Interview Participants Demographics. In this study, two distinct snowball sampling chains were established, delineated as follows: 1) P08-P12-P14-P19. 2) P16-P21-P26. The remaining participants were incorporated into the study through direct recruitment. This study categorizes participants based on their children's programming learning status. The "status" column marks the first group as "continue", indicating these participants' children are in the ongoing learning process. The second group is marked as "give up", indicating that these participants' children have stopped learning programming after a certain period. The "Languages" column indicates the programming languages they learned: "L" for Lego programming bricks, "S" for Scratch, "P" for Python, and "C" for C++. The "Duration" column shows their experience length. The "Programming Exp." column indicates whether parents reported prior programming experience; only three parents (P03, P11, P15) had learned programming courses during their own student years and possessed basic programming foundations.

Participant				Child's Information				
ID	Gender	Occupation	Programming Exp.	Gender	Age	Languages	Duration	Status
P01	F	Freelancer	No	F	8	S	6mo-1yr	Continue
P02	M	Management	No	F	9	S	1-2yr	Continue
P03	F	Automotive Engineer	Yes	F	10	S, P, C	>3yr	Continue
P04	M	Cost Engineer	No	F	9	L, S	<3mo	Continue
P05	M	Freelancer	No	F	8	S	<3mo	Give up
P06	F	Translator	No	F	10	S, C	2-3yr	Continue
P07	M	Civil Engineer	No	F	11	S, P, C	1-2yr	Continue
P08	F	Receptionist	No	F	10	S	<3mo	Give up
P09	M	Teacher	No	F	10	S, P, C	>3yr	Continue
P10	F	Clerk	No	F	8	S	6mo-1yr	Continue
P11	M	Mechanical Engineer	Yes	F	8	S, P, C	6mo-1yr	Continue
P12	M	Teacher	No	F	11	S	<3mo	Give up
P13	F	Management	No	F	9	S, C	6mo-1yr	Continue
P14	F	Accountant	No	M	9	S	6mo-1yr	Give up
P15	M	Technical Engineer	Yes	M	8	S	6mo-1yr	Continue
P16	F	Construction Manager	No	M	11	S, C	2-3yr	Continue
P17	M	Auditor	No	M	7	S	<3mo	Continue
P18	M	Sales	No	M	7	S	<3mo	Give up
P19	F	Freelancer	No	M	7	S	6mo-1yr	Continue
P20	F	Corporate Clerk	No	M	8	S, P	1-2yr	Continue
P21	F	Doctor	No	M	11	S, P, C	>3yr	Continue
P22	M	Mechanical Engineer	No	M	10	S, C	2-3yr	Continue
P23	F	Cost Estimator	No	M	9	S	<3mo	Continue
P24	M	Receptionist	No	M	7	S	6mo-1yr	Continue
P25	F	Doctor	No	M	10	L, S	<3mo	Continue
P26	F	Dance Instructor	No	M	12	S, C	2-3yr	Continue

In the second part, we examined how parents were involved in their children's programming learning. This includes how they **support** learning over time through resources, assistance, and guidance. We also asked how they **evaluate** learning outcomes and adjust their involvement strategies accordingly. This theme captures parents' attention to and supervision of children's programming learning, indirectly reflecting involvement (Sui-Chu and Willms, 1996).

In the final part, parents described difficulties encountered during involvement, especially moments that weakened their willingness to stay involved. We also asked how they coped with these difficulties based on recalled experiences. The full interview guide is provided in Supplementary Appendix B.

3.4 Data Analysis

Our methodology for analyzing the interview data mainly involved thematic analysis(Fereday and Muir-Cochrane, 2006; Holton, 2007), grounded in the interview transcripts. The process encompassed developing and validating the codebook, and abstracting core themes and conceptual frameworks.

Initially, two researchers independently analyzed a subset (20%) of the interview transcripts to identify emerging themes related to parental involvement in children's programming learning and perceived challenges. They then compared and discussed the themes. Disagreements were debated, and the themes were refined and synthesized into an initial codebook.

To validate reliability, we randomly selected another 10% of transcripts for independent deductive analysis by the two researchers. This yielded a Cohen's Kappa score (McHugh, 2012) of $\kappa = 88.2$, indicating high interrater reliability and shared understanding of the codebook. Remaining disagreements were discussed among all researchers, leading to further refinement. The remaining 70% of transcripts were then divided between the two researchers, who deductively applied the refined codebook. This iterative process strengthened the rigor and validity of our findings. We assessed data saturation by listing emergent codes in interview order (P01–P26) and found no new themes in later interviews, suggesting theoretical saturation.

Then, researchers moved from detailed codes to higher-level interpretations by organizing codes into meaningful groups and categories. We identified two core themes for RQ1 and six themes to summarize RQ2. We then examined relationships among themes and developed a conceptual framework for each RQ to produce coherent data representations. Regarding RQ1, we used the key directions in Section 3.3 as the core themes—**support and evaluation**. We identified parental support in two learning scenarios: at-home and extracurricular learning. Parents provide academic guidance during at-home learning; introduce additional learning resources and professional educators when required knowledge exceeds their own; and facilitate parent-child communication in flexible, everyday contexts. We also outlined five key factors parents use to evaluate the effectiveness of their children's programming learning.

For RQ2, which addresses challenges faced by parents, we organized seven challenges into three groups based on their sources: personal limitations, concerns about children's learning performance, and external environmental pressure. We further examined whether parents reported mitigation strategies alongside these challenges, offering insights into the types of support parents find helpful.

3.5 Limitation

Our study has two main limitations: the geographic concentration of interviewees and the imbalance between sample groups.

First, all participants were recruited from China, which may introduce cultural and educational-context bias and limit transferability to other regions. To reduce sample homogeneity, we used multiple recruitment channels and controlled key demographics (e.g., gender, age, occupation) and children's programming experience to ensure diversity. Our findings should therefore be interpreted as context-grounded evidence that motivates broader cross-regional validation.

Second, parents who had ceased supporting their children's programming learning were underrepresented, which may skew perceptions toward those who remained engaged. We mitigated this by recruiting parents whose children had diverse learning trajectories, including those who paused one language, shifted to another, or were considering the next stage. Given the valuable insights from the current sample, some imbalance does not compromise the validity of our findings.

4 Parental Involvement of Children's Programming Learning (RQ1)

Based on our thematic analysis, we characterize parental involvement in children's programming learning along two dimensions: support and evaluation (as summarised in Table 2). *Support* captures parents' actions and investments that facilitate learning (e.g., opportunities, guidance, and encouragement), making their involvement visible. *Evaluation* captures how parents assess learning progress and outcomes, which in turn shapes how they adjust involvement strategies. This section reports the forms of support parents provide and the factors they use to evaluate learning outcomes.

Table 2. Dimensions of parental involvement in children's programming learning. Items marked with * in the second column reflect involvement practices that were more salient in programming learning and differ from common patterns in traditional learning contexts.

Parental Involvement		Explanation
Parental Support	Academic Guidance*	Clarifying task requirements, explaining unfamiliar terms, and supporting basic tool use to reduce entry barriers.
	Learning Resource Support	Brokering external resources (e.g., courses, platforms, instructors) when in-home guidance is limited.
	Parent-Child Communication	Sustaining interest through everyday conversations that connect programming to real-world applications and future goals.
Parental Evaluation	Computational Concepts Mastery*	Assessing understanding of core concepts (e.g., execution order, loops, operators) through targeted exercises and error patterns.
	Practical Abilities*	Judging competence via the ability to produce functional projects, debug problems, and complete tasks with increasing independence.
	Emotional Expressions	Using affective engagement (e.g., curiosity vs. frustration) as a signal of learning quality and continuation.
	Academic Achievements	Using grades or school performance as a familiar benchmark for the perceived payoff of programming learning.
	Educators' Comments	Using instructor feedback as an external reference for progress, especially under limited self-evaluation capacity.

4.1 Parental Supports of Children's Programming Learning

We found that parents support children's programming learning through three approaches across two learning scenarios. During at-home practice, parents provide academic guidance by helping children navigate reading comprehension and tool-use barriers and by structuring practice tasks. When direct guidance is not feasible, they broker external resources by enrolling children in extracurricular programs and relying on professional educators. Parents also facilitate parent–child communication through everyday conversations about programming applications and the broader impacts of technology.

Providing external resources and fostering everyday communication largely align with established patterns of parental involvement in prior work. Parents often compensate for limited in-home expertise by seeking structured programs and sustaining children's interest through contextualized discussions (Hill and Tyson, 2009; Jaiswal, 2017; Park, 2008). Accordingly, although we retain these categories in Table 2, we foreground *academic guidance* in the main text, as it becomes substantially more complex and demanding than guidance in more conventional learning contexts.

ACADEMIC GUIDANCE. Programming presents a certain learning curve for children, requiring them to understand task instructions, operate programming tools, and gradually learn abstract computational concepts (Evangelidis et al., 2001; Khalife, 2006; Wang et al., 2025). In our interviews, parents frequently mentioned the need for some form of academic guidance at home. However, the scope and depth of this guidance depend on the parents' own background. Most parents in our sample had no prior programming experience, so their support mainly focused on helping children overcome immediate comprehension and operation barriers, such as reading task descriptions, explaining unfamiliar terms, typing, using the computer, or understanding what an exercise required them to do. As one parent stated, *"When she asks me questions at home, I do not really understand programming, so I cannot teach her what to do. But sometimes she cannot even understand the task requirements, so I usually need to explain what the task requires to do with her"* (P05).

By contrast, some parents possessed more programming-related knowledge and were therefore able to provide more programming-specific support at home. These parents were not limited to those who had taken relevant courses during university and thus had direct programming experience; some had also gradually become familiar with their children's course content through the process of supporting their programming learning. Beyond helping children clarify task requirements, they searched for additional exercises, Scratch project kits, code examples, or online course materials that matched the child's current learning stage. As one parent noted, *"I search for publicly available Scratch project kits online to see if they match my child's current knowledge level"* (P22).

Nevertheless, even when parents were willing to guide children, their support was often limited once children encountered code-level problems. This was especially evident in debugging, where parents needed to understand not only whether the program could run, but also why a script failed and how it could be revised. Even parents with some programming exposure found a gap between their own limited knowledge and the content their children were currently learning. As one parent explained, *"Although I learned programming during university, what she is learning now is still quite different from what I know. I am also not a professional programmer now, and I have forgotten many details over time. At most, I can help her check whether the program runs properly, but I cannot solve the specific errors"* (P11).

4.2 Parental Evaluation Factors of Children's Programming Learning

Parental involvement encompasses the practices parents undertake to support their children's academic success (Fan and Chen, 2001). Beyond the direct supports described in Section 4.1, evaluating children's programming outcomes constitutes a central component of such involvement. Through evaluation, parents gauge progress, build confidence in providing support, and sustain their engagement. It also enables them to identify challenges and seek additional assistance or resources when needed.

Our findings show that parents consider five factors when assessing programming learning outcomes: children's emotional expressions, academic achievements, understanding of computational concepts, practical programming abilities, and educators' comments. The first three—emotional expressions, academic achievements, and educators' comments—reflect assessment practices commonly observed in traditional learning contexts (Knox et al., 2023; Kraft and Rogers, 2015; Yamamoto and Holloway, 2010). We therefore summarize them in Table 2, while focusing the remainder of this section on the two dimensions more specific to programming education: mastery of computational concepts and practical programming abilities.

Computational concepts mastery. In programming education, some parents recognize the importance of cultivating computational thinking and interpret specific constructs (e.g., loop nesting) as exercises in logical reasoning. They treat children's fluent use of particular code blocks or functions as evidence of mastery of related concepts. In our sample, only a small number of parents—typically those with prior programming experience—reported adopting this evaluative approach. These parents often used open-source online exercises to assess children's understanding of execution order, loops, and operators [2], aligning with established computational thinking assessment frameworks (Brennan and Resnick, 2012). By reviewing response accuracy, especially incorrectly answered items, they sought to identify conceptual gaps. As one parent explained, *"An online question bank is available, and each question usually indicates the specific knowledge point being tested, allowing precise identification of areas that have not been mastered"* (P15).

However, due to the inherent thresholds of programming sill, this is the least mentioned evaluation factor. Parents typically possess insufficient understanding of the computational concepts integral to programming education and lack awareness of those indispensable for the development of computational thinking, thereby impairing their ability to conduct a proficient evaluation. Additionally, contrary to evaluating children's progress in traditional academic subjects, parents are unable to assess their children's comprehension of these concepts through daily interactions such as assessing arithmetic progress through counting or language skills through conversation. *"I have no means to determine her progress from her daily behavior since our interactions do not incorporate programming concepts"* (P05).

Practical abilities. Some parents contend that focusing solely on conceptual understanding to cultivate computational thinking is insufficient to demonstrate a

[2] https://scratched.gse.harvard.edu/ct/defining.html

child's mastery of programming. Instead, they emphasize children's design practices in project creation, which involve iteratively applying and remixing acquired coding knowledge to generate ideas beyond existing examples. In their view, the ability to use programming as a creative medium—rather than merely solving assigned problems—reflects deeper understanding and more proficient application. Consequently, they regard sustained programming practice as a meaningful indicator of proficiency, growth, and creativity. As P22 noted, *"If he can effectively apply the concepts taught by the teacher and independently design his own programs, it is a strong indication that they have mastered the knowledge"*.

However, influenced by cultural and environmental factors, some parents hold highly utilitarian expectations for their children's programming education, focusing solely on exam and competition results. These parents believe that programming practice and code innovation are less significant and prioritize competition achievements. They may even discourage their children from spending too much time on personal coding practice. *"I want him to focus on practice competition questions repeatedly and not make irrelevant code attempts"* (P26).

5 Parents' Perceived Challenges of the Involvement in Children's Programming Learning (RQ2)

Parents' perceived challenges refer to difficulties they encounter while supporting children's programming learning, which can strongly shape whether they remain involved over time. This section summarizes three sources of challenges—personal limitations, concerns about children's learning performance, and external environmental pressures (Table 3)—and describes how they affect parents' investment and mitigation strategies.

5.1 Challenges Due to Personal Limitations

The first category of challenges parents face stems from their limitations, primarily related to financial issues and the inability to bridge their information gaps. Specifically, these challenges can be divided into the following two types.

FINANCIAL CONSTRAINT. For some families, the financial burden of programming learning can exceed their budget, leading them to reconsider involvement. The cost of learning resources is a primary concern. Two parents (P04, P25) involved in tangible programming described the high expense of required materials. *From large-particle building blocks to small particles to robots, a substantial amount of materials must be purchased, which often remain unused after a short period. The cost is very high"* (P25). Parents of children moving to text-based languages—especially those receiving additional competition training—face even greater pressure. *They are preparing for this year's competition, with extra lessons each week and intensive summer training. This expense is significant"* (P21). Financial constraints are also salient for families with multiple children. *Both of my children initially attended programming classes, but as the content advanced, they needed additional lessons, which became unaffordable"* (P08). These parents often assume that only paid

Table 3. Challenges Encountered by Parents in Their Children's Programming Learning. We have identified 5 specific challenges and categorized them into three types based on their sources. The three main sources are personal limitations, concerns about children's learning performance, and external environment pressure.

Source of Challenges	Specific Challenges	Explanation
Personal Limitations	Financial Constraint	Limited funds to afford the hardware, software licenses, or programming courses.
	Information and Resource Insularity	Limited to a single source of information, unaware of other open-source materials.
Concerns about Children's Learning Performance	Accumulated Negative Emotions of Children	Difficulty in soothing and managing the child's frustration, anxiety, and lack of confidence resulting from continuous struggles and setbacks in learning.
	Imperceptibility of Children's Learning Progress	Difficulty in perceiving the progress, or positive impacts in children's programming learning, leading to doubts about the effectiveness of programming learning.
External Environment Pressure	Competition Pressure	Parents with utilitarian expectations feel pressured in a competitive environment, while those without such expectations may also experience anxiety about their initial decisions due to the competitive atmosphere.

courses provide proper instruction; when they stop paying for tutoring, they also limit children's access to free learning resources.

INFORMATION AND RESOURCE INSULARITY. Parents also identified information and resource insularity as a key challenge in sustaining involvement, particularly in accessing professional, systematic resources and reliable reference information. As newcomers to programming, many hoped to support their children with high-quality resources but did not know where to begin. Consequently, they relied heavily on friends' recommendations. This peer-based search often led parents to converge on the same courses or instructors; when these proved unsatisfactory, they struggled to locate better alternatives. *"In our previous city, my child's programming class was excellent, and parental involvement activities helped us understand their progress. However, the current institution lacks such activities, and my child's enthusiasm has diminished. I don't know where to find better resources, and apart from asking friends, I don't know what to do"* (P25).

Even parents who actively search face similar constraints, as their strategies are often limited to basic internet searches. They described online information as homogeneous and overly general, with limited stage-specific guidance. As shown in Figure 1, parents noted that search results are abundant but largely promotional, while actionable information for particular learning stages is scarce. *"Most of the online search content emphasizes the benefits of programming but lacks practical advice. Some topics, like dynamic programming in algorithms, are beyond our comprehension. What is that? I don't understand"* (P09). Without effective ways to obtain professional guidance, parents found it difficult to move beyond these constraints.

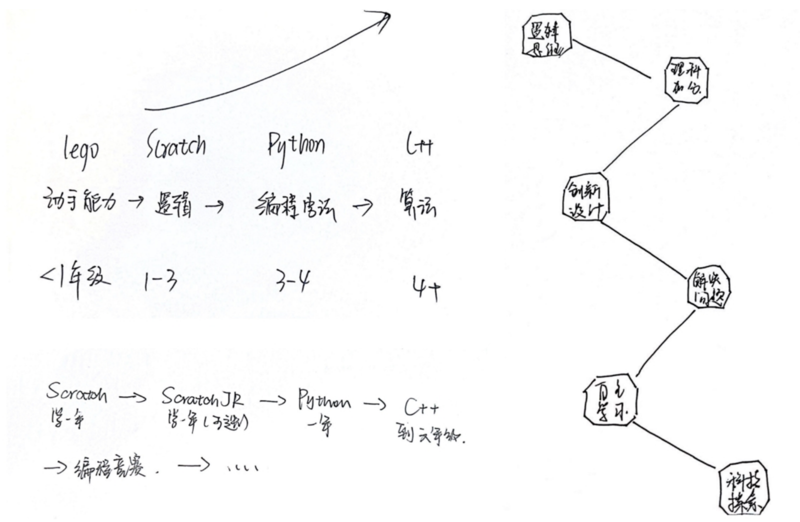

Figure 1. Programming Education Information Resources Accessed by Parents. The images show sketches drawn by three respondents, depicting the promotional information about programming education they have encountered. The Chinese text roughly translates to the benefits of programming education for children's logical thinking and problem-solving abilities, as well as the programming languages suitable for children at specific age stages.

5.2 Challenges Due to Concerns about Children's Learning Performance

For parents, children's programming progress is a salient concern. When children show negative attitudes or when short-term progress is hard to perceive, parents may feel anxious and uncertain, which can weaken their commitment to continued involvement. If these challenges persist, some parents consider withdrawing their investment in programming learning.

Accumulated negative emotions of children. Accumulated negative emotions refer to the frustration, anxiety, and self-doubt children may experience as they encounter difficulties in programming learning. These emotions can build up over time, especially as content becomes more complex. For parents, managing these emotions and providing guidance is a major challenge. Most parents do not readily indulge negative emotions or allow children to quit; guidance and encouragement are seen as necessary. *"Fear of difficulty is natural, even for adults, but I won't let her stop learning just because programming is hard. I'll actively guide and encourage her"* (P02). Parents typically consider stopping only when children show long-term fatigue, loss of interest, or repeated emotional breakdowns. However, some parents with utilitarian views see a temporary pause as acceptable. *"He must persist until the competition ends. If he has an emotional breakdown and wants to stop, I can allow a one-month break, but quitting entirely is not an option"* (P26). This strong-willed approach, which turns expectations into mandatory goals, may inadvertently intensify children's resistance to programming learning.

Imperceptibility of children's learning progress. Imperceptibility of learning progress occurs when parents cannot observe expected improvements or discern the benefits programming is supposed to bring. When this happens early, some parents may decide to stop to avoid further sunk costs. For example, parents new to graphical programming may conclude that their child is not suited for programming. *"I send her to classes, but I never see her independently applying her programming skills. I feel this investment is not worthwhile"* (P08). When children have learned for longer, parents may instead feel pressured to continue, worrying that quitting would nullify prior effort. *"He's been learning for so long, but if he gives up now, all the previous efforts will be in vain. Although we've considered quitting, we hope he can stick with it for future benefits"* (P22). We identify two reasons why progress is hard to perceive. First, many parents lack a clear evaluation framework and rely on limited cues, such as educators' comments; when feedback is vague, their support may waver. *"the teacher just said he wasn't distracted in class. I'm not sure how this benefits him in other areas, so we decided not to continue"* (P18). Second, parents often judge short-term learning against long-term goals, overlooking incremental milestones. This mismatch can make early-stage learning appear unproductive, especially in graphical programming where gains may not translate into visible daily-life changes. *"I haven't seen any progress after learning graphical programming. I think it's useless"* (P18).

5.3 Challenges Due to External Environment Pressure

As parents become involved in children's programming learning, they inevitably compare their children with peers in the broader educational environment. In contexts where social, institutional, and cultural factors intensify competition, this competitive landscape becomes a major source of pressure and challenge for parents.

COMPETITION PRESSURE. In certain cultural and social contexts, such as the Chinese educational environment, limited access to high-quality educational resources can intensify competition (Kai, 2012; Zhao, 2015). Some parents described an educational environment in which children could enter more advanced or leading classes through selection based on programming competitions. They believed that these classes offered access to more experienced teachers, more systematic instruction, and a stronger peer learning atmosphere. As one parent explained, *"Entering a good class not only proves the child's ability, but more importantly provides access to better resources, because a good class usually means better teachers, stronger classmates, and a better learning atmosphere"* (P10). Therefore, in this context, programming learning became closely tied to parental stress and anxiety, as many parents worried about whether their children could stand out and secure such opportunities. *"I feel more anxious than my child; he might be indifferent, but I understand the implications of failure"* (P21).

Some parents therefore explicitly oriented children's learning toward contests, treating persistence as contingent on short-term milestones and disengaging once a goal was achieved (e.g., after a competition). This perspective was particularly common among parents of upper primary students learning text-based programming who were approaching entrance exams. *"Isn't the current stage of learning programming for children primarily aimed at preparing for competitions"* (P26)? At the same time, even parents who were initially uninterested in competitions felt pressured to follow suit. When they saw other families treating competitions as a route to better classes and teachers, they became less certain about supporting programming as an interest-driven activity. *"I see that parents around me who send their children to learn programming are all pushing them to compete in contests. I* *didn't focus on this aspect initially, but now I'm increasingly wavering. I also don't know if this will diminish my child's interest in programming*" (P15). Although many parents felt uneasy about this pressure, they perceived it as pervasive and difficult to resist.

6 Discussion

This study examines parents' sustained involvement in children's programming education, with a focus on the support and evaluation strategies they adopt and the challenges they encounter. Building on these insights, we consider how parental involvement can be more effectively integrated into programming education. Specifically, this section first reflects on the gap between how parental involvement is currently envisioned in programming education and how it is enacted in practice, highlighting the distinctive support demands of programming learning and the sustained nature of parents' engagement. We then discuss how we can enhance the effectiveness of parental involvement when designing future educational resources.

6.1 Rethinking Parental Involvement in Programming Education

This study examines parents' sustained involvement in children's programming education, focusing on the support and evaluation strategies they adopt and the challenges that shape these practices over time. Across our interviews, we observed that most parents rarely engaged as direct technical instructors. Instead, their involvement was largely expressed through indirect forms of support, such as coordinating learning resources, organizing extracurricular courses, and helping children manage frustration during difficult programming tasks (Section 4.1).

Rather than simply extending familiar patterns of parental involvement into a new subject area, programming education appears to reconfigure how parental involvement is enacted in practice. Specifically, our findings suggest that parental engagement is shaped by two structural characteristics of programming learning environments: (1) its technical specificity and (2) its staged, long-term developmental trajectory. Together, these characteristics shift parental participation from direct pedagogical authority toward distributed forms of infrastructural and interpretive support within a sociotechnical learning environment.

First, the technical and professional specificity of programming often creates epistemic asymmetry between parents and children, in which children may quickly develop domain knowledge that parents do not possess. Parental involvement has long been associated with improved learning outcomes and motivation (Gonzalez-DeHass et al., 2005; Sapungan and Sapungan, 2014), and existing work in programming education often portrays parents as instructional collaborators who assist with tool use, explain programming logic, or troubleshoot code during structured learning activities (Ehsan et al., 2023; Elkin et al., 2016; Fields et al., 2015; Lavigne et al., 2023; Sheehan et al., 2019).

However, our findings indicate that sustained hands-on parental involvement was uncommon. As discussed in Section 4.1, only a small subset of parents, primarily those with prior programming experience, were able to act as technical mentors, while most struggled to provide direct assistance. Many reported difficulty interpreting debugging errors, grasping abstract programming concepts, navigating

English-dominant documentation (Dasgupta and Hill, 2017), or keeping pace with rapidly evolving platforms. Importantly, these challenges extend beyond general academic difficulty. Programming learning entails tool dependence, escalating abstraction (e.g., transitions from graphical to text-based languages) (Tsukamoto et al., 2015), iterative debugging practices, and fast-changing technological ecosystems. Unlike subjects such as mathematics or language learning, where parents can gauge progress through everyday interactions, programming competence often remains invisible in routine family contexts.

As a result, many parents repositioned themselves into alternative yet consequential roles. Instead of acting as instructors, they became brokers of learning resources, coordinators of time and logistics, interpreters of fragmented online information (Mandinach et al., 2020), and regulators of children's emotional responses to frustration (Sheehan et al., 2019).

We conceptualize these practices as forms of domestic infrastructuring work—ongoing efforts through which families coordinate resources, information, and emotional support to sustain learning activities. This conceptualization is consistent with prior CSCW work describing children's technology-mediated learning as sustained through parents' ongoing coordination of tools, resources, routines, and adult support, rather than through technical systems alone (Cingel and Piper, 2017; Yu et al., 2021). In our context, parents assembled courses, platforms, teachers, peer information, schedules, financial resources, and emotional support into a workable learning arrangement for the child. Our findings show that such coordination was essential for sustaining children's participation, especially when parents lacked the expertise to act as technical instructors. Programming education therefore did not diminish parental involvement; rather, it redistributed involvement from co-instruction toward infrastructural coordination, affective support, and interpretive mediation. This reframing highlights the often invisible work required to maintain children's participation in programming learning, particularly for families without professional technical backgrounds.

Second, the long-term and staged developmental trajectory of programming education reshapes parental involvement as an adaptive and temporally negotiated process. Programming learning typically unfolds through transitions—from exploratory graphical programming to text-based languages, and from creative experimentation to competition-oriented preparation. Each stage introduces increasing abstraction, cognitive demand, and often greater financial investment.

At the same time, the benefits commonly associated with programming learning—such as computational thinking, logical reasoning, and future career advantages (Kafai and Burke, 2014; Nouri et al., 2020)—are often delayed and difficult to observe in everyday contexts. As parents in our study described (Section 5.2), early-stage learning frequently produced few visible behavioral changes, making short-term progress difficult to assess.

We describe this dynamic as longitudinal sensemaking under delayed payoff conditions, where parents must repeatedly interpret uncertain signals of progress and decide whether to continue investing time and resources. As shown in Section 5, parents struggled with imperceptible learning progress, accumulated negative emotions in children, fragmented resource ecosystems, and competitive institutional pressures (Kai, 2012; Zhao, 2015).

In response, parents engaged in ongoing recalibration of their involvement strategies. They intensified support when competition milestones or teacher feedback signaled progress, experimented with alternative platforms or courses when learning appeared stagnant, and reconsidered continuation when emotional fatigue accumulated. Parental involvement therefore emerged not as a stable commitment but as a cyclical process of interpretation, adjustment, and reinvestment shaped by feedback loops across home, school, platforms, and peer networks.

Taken together, these findings suggest that parental involvement in programming education is less accurately characterized as direct co-teaching and more productively understood as distributed infrastructuring under epistemic asymmetry and temporal negotiation under uncertainty.

While competitive pressures may be culturally intensified in contexts such as China, the two structural conditions identified in our study—epistemic asymmetry and delayed visibility of learning progress—are likely to extend across many programming learning environments. Programming therefore offers a particularly revealing case for examining how families navigate technically complex learning domains. Reframing parental involvement in this way shifts the design question for HCI research. Instead of asking how parents can become better technical instructors, designers should consider how sociotechnical learning systems can better support parents' infrastructural, affective, and interpretive labor over time. This perspective encourages future systems to support not only children's interaction with programming tools, but also caregivers' ongoing coordination, decision-making, and sensemaking across evolving learning trajectories.

6.2 Design Implications

Our findings call for a shift in how sociotechnical systems conceptualize and support parental involvement in children's programming education. Many tools and curricula implicitly assume parents act as direct pedagogical agents, providing hands-on guidance or technical expertise. However, as shown in Sections 6.1, most parents—especially those without programming backgrounds—engage through indirect, distributed, and affective support. Accordingly, we propose two design directions that mirror these domain characteristics. One focuses on enabling low-barrier, non-technical involvement under the technical demands of programming, and the other supports longitudinal sensemaking and adaptive engagement along extended learning pathways.

Supporting low-barrier parental involvement through shared creative artifacts. Our results in Section 4.1 and the discussion in Section 6.1 show that many parents felt hesitant to directly help children with programming because they lacked relevant expertise. However, they still supported children's learning through encouragement, observation, resource coordination, and emotional support. We therefore propose that systems should enable non-technical involvement that helps families sustain children's learning over time.

One possible approach is to design programming projects as shared creative artifacts that parents and children can use for communication. This is especially suitable for Scratch and other block-based programming environments, where learning is expressed not only through code, but also through visible works such as stories, games, animations, characters, and scenes. Drawing on the constructionist tradition, Papert described computational artifacts as "objects to think with", because they allow learners to make their ideas visible, examine them, and revise them through creation (Lodi and Martini, 2021; Papert, 1981). From this perspective, children's projects can also become family-facing objects for conversation and reflection. For example, co-creative platforms could allow parents to take part in expressive aspects of a project, such as sketching visual elements, selecting assets, refining narratives, arranging scenes, or commenting on the emotional tone of a story, while children remain responsible for coding and logic construction. These non-code contributions allow parents to discuss problem-solving strategies, creative decisions, and learning guidance without directly debugging scripts (Kelty and Wakabayashi, 2020; Zhang et al., 2022).

In addition, observation-based involvement should also be supported as part of broader family work. Public showcases, project walkthroughs, or debugging demonstrations can help parents see children's progress and offer timely encouragement, rather than relying only on second-hand feedback from educators (Section 4.2). Prior work on family technology practices also shows that parental support is often embedded in everyday negotiation, rule-setting, encouragement, and care, rather than formal instruction alone (Yen et al., 2018). Designing for these roles can recognize parents' emotional support, celebration of milestones, and help during moments of frustration, all of which are important for sustaining children's long-term engagement in programming education.

Scaffolding parental decision-making and sensemaking. As discussed in Section 6.1, parental involvement evolves through a dynamic sensemaking process: parents interpret children's interest, performance, motivation, and emotional state to decide whether to sustain, reduce, or redirect their involvement (Walker and MacLure, 2005). However, many parents struggled with information overload, fragmented course markets, and limited structured feedback from educational providers (Yu et al., 2020). From a CSCW perspective, these challenges can be understood as breakdowns in the family learning infrastructure. Parents were not only choosing tools or courses; they were trying to make sense of distributed information across children's projects, teacher feedback, platform reports, peer recommendations, and market claims. When these elements were poorly connected, parents had difficulty knowing whether a child's frustration reflected productive struggle, inappropriate course difficulty, declining interest, or a mismatch between learning goals and instructional design.

Systems should therefore support parental sensemaking and infrastructural repair by making children's learning trajectories more visible and interpretable. Rather than providing only performance scores or generic progress dashboards, professionally guided platforms could offer longitudinal development maps, stage-based learning goals, examples of age-appropriate project outcomes, and explanations of common learning difficulties. These features would help parents interpret where their child is within a broader learning pathway and adjust expectations accordingly. For instance, a platform could help parents distinguish when self-driven exploration is appropriate, when structured instruction may be needed, and when emotional encouragement may be more useful than additional technical intervention. In this sense, the system would not replace parental judgment, but provide resources that make parents' existing sensemaking work more informed and less isolating.

Such platforms could also support collective sensemaking among families. CSCW research has long emphasized that coordination and learning are distributed across people, artifacts, and social arrangements rather than located only in individual users or tools (Lee and Paine, 2015). In programming education, forums or peer-sharing spaces could help parents surface hidden concerns, compare strategies, and learn how other families respond to similar moments of uncertainty. This is particularly important in fragmented and sometimes contradictory programming education markets, where parents must navigate competing claims about courses, platforms, and children's future opportunities (Kai, 2012; Zhao, 2015). By connecting expert guidance, peer experience, and children's own project histories, learning systems can better support the ongoing work of maintaining a feasible, emotionally sustainable, and developmentally appropriate programming learning infrastructure at home.

7 Conclusion

This study primarily investigates parental involvement in children's programming learning as well as the challenges they encountered during their involvement. We elucidate the efforts parents make to support their children's programming learning, and how challenges influence their decision-making process. Building on the discussion surrounding the value and role of parents in programming education, we proffer design insights aimed at enhancing parental integration within the domain of programming education.

Acknowledgments

This work is supported by the Science and Technology Development Fund (FDCT), Macau SAR (File no. 0078/2023/AMJ) and the University of Macau (File no. MYRG-GRG2024-00052-FST-UMDF, no. MYRG-GRG2025-00111-FST, no. EF2025-00141-FST and no. EF2024-00231-FST).

References

Avvisati, Francesco, Bruno Besbas, and Nina Guyon. 2010. Parental involvement in school: A literature review. *Revue d'économie politique*, 120 (5): 759–778.

Badrasawi, Kamal, Humaer Yahefu, and Madihah Khalid. 2019. Challenges to parental involvement in children's education at a primary school: A Rasch analysis. *IIUM Journal of Educational Studies*, 7 (1): 47–57.

Banerjee, Rahul, Leanne Liu, Kiley Sobel, Caroline Pitt, Kung Jin Lee, Meng Wang, Sijin Chen, Lydia Davison, Jason C Yip, Amy J Ko, et al. 2018. Empowering families facing english literacy challenges to jointly engage in computer programming. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–13.

Brennan, Karen and Mitchel Resnick. 2012. 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.

Bresnihan, Nina, Glenn Strong, Lorraine Fisher, Richard Millwood, and Áine Lynch. 2019. OurKidsCode: Facilitating Families to Be Creative with Computing. In: *CSEDU (1)*, pp. 519–530.

Bresnihan, Nina, Glenn Strong, Lorraine Fisher, Richard Millwood, and Áine Lynch. 2020. Increasing parental involvement in computer science education through the design and development of family creative computing workshops. In: *Computer Supported Education: 11th International Conference, CSEDU 2019, Heraklion, Crete, Greece, May 2-4, 2019, Revised Selected Papers 11*, pp. 479–502.

Cagiltay, Bengisu, Hui-Ru Ho, Kaiwen Sun, Zhaoyuan Su, Yuxing Wu, Olivia K Richards, Qiao Jin, Junnan Yu, Jerry Alan Fails, Jason Yip, et al. 2024. Methods for Family-Centered Design: Bridging the Gap Between Research and Practice. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pp. 1–6.

Cai, Haiyan and Gary KW Wong. 2022. Parental Involvement in Computational Thinking Education: A Systematic Review. In: *2022 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, pp. 572–579.

Castro, María, Eva Expósito-Casas, Esther López-Martín, Luis Lizasoain, Enrique Navarro-Asencio, and José Luis Gaviria. 2015. Parental involvement on student academic achievement: A meta-analysis. *Educational research review*, 14: 33–46.

Cingel, Drew and Anne Marie Piper. 2017. How parents engage children in tablet-based reading experiences: An exploration of haptic feedback. In: *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pp. 505–510.

Coleman, Mick and Susan Churchill. 1997. Challenges to family involvement. *Childhood Education*, 73 (3): 144–148.

Dasgupta, Sayamindu and Benjamin Mako Hill. 2017. Learning to code in localized programming languages. In: *Proceedings of the fourth (2017) ACM conference on learning@ scale*, pp. 33–39.

Druga, Stefania, Thomas Ball, and Amy Ko. 2022a. How families design and program games: a qualitative analysis of a 4-week online in-home study. In: *Proceedings of the 21st Annual ACM Interaction Design and Children Conference*, pp. 237–252.

Druga, Stefania, Fee Lia Christoph, and Amy J Ko. 2022b. Family as a Third Space for AI Literacies: How do children and parents learn about AI together? In: *Proceedings of the 2022 CHI conference on human factors in computing systems*, pp. 1–17.

Ehsan, Hoda, Carson Ohland, and Monica E Cardella. 2023. Characterizing child–computer–parent interactions during a computer-based coding game for 5-to 7-year-olds. *Education Sciences*, 13 (2): 164.

Elkin, Mollie, Amanda Sullivan, and Marina Umaschi Bers. 2016. Programming with the KIBO robotics kit in preschool classrooms. *Computers in the Schools*, 33 (3): 169–186.

Epstein, JL. 1991. Effects on student achievement of teacher's practices of parent involvement. *Advances in reading/language research*, 5.

Erete, Sheena, Naomi Thompson, Miranda Standberry-Wallace, Bo Ju, Denise Nacu, and Nichole Pinkard. 2021. Honoring black women's Work: creating a parent and caring adult community to support youth STEAM engagement. In: *2021 Conference on Research in Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*, pp. 1–5.

Erlangga, Dion Tira. 2022. Student problems in online learning: solutions to keep education going on. *Journal of English Language Teaching and Learning*, 3 (1): 21–26.

Evangelidis, Georgios, Vassilios Dagdilelis, Maria Satratzemi, and Vassilios Efopoulos. 2001. X-compiler: Yet another integrated novice programming environment. In: *Proceedings IEEE International Conference on Advanced Learning Technologies*, pp. 166–169.

Falkner, Katrina, Rebecca Vivian, and Nickolas Falkner. 2014. The Australian digital technologies curriculum: challenge and opportunity. In: *Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148*, pp. 3–12.

Fan, Xitao and Michael Chen. 2001. Parental involvement and students' academic achievement: A meta-analysis. *Educational psychology review*, 13: 1–22.

Fereday, Jennifer and Eimear Muir-Cochrane. 2006. Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development. *International journal of qualitative methods*, 5 (1): 80–92.

Fields, Deborah, Veena Vasudevan, and Yasmin B Kafai. 2015. The programmers' collective: fostering participatory culture by making music videos in a high school Scratch coding workshop. *Interactive Learning Environments*, 23 (5): 613–633.

Frome, Pamela M and Jacquelynne S Eccles. 1998. Parents' influence on children's achievement-related perceptions. *Journal of personality and social psychology*, 74 (2): 435.

Gonzalez-DeHass, Alyssa R, Patricia P Willems, and Marie F Doan Holbein. 2005. Examining the relationship between parental involvement and student motivation. *Educational psychology review*, 17: 99–123.

Goodman, Leo A. 1961. Snowball sampling. *The annals of mathematical statistics*, pp. 148–170.

GOVIND, MADHU and MARINA UMASCHI BERS. 2020. Family coding days: Engaging children and parents in creative coding and robotics. *Connected Learning Summit, Cambridge, MA*, 66.

Govind, Madhu, Emily Relkin, and Marina Umaschi Bers. 2020. Engaging children and parents to code together using the ScratchJr app. *Visitor Studies*, 23 (1): 46–65.

Govindarajan, Madhumita. 2019. Families That Code Together Learn Together: Exploring family-oriented programming in early childhood with ScratchJr and KIBO Robotics. Master's thesis, Tufts University.

Hill, Nancy E and Diana F Tyson. 2009. Parental involvement in middle school: a meta-analytic assessment of the strategies that promote achievement. *Developmental psychology*, 45 (3): 740.

Ho, Hui-Ru, Nitigya Kargeti, Ziqi Liu, and Bilge Mutlu. 2025. SET-PAiREd: Designing for Parental Involvement in Learning with an AI-Assisted Educational Robot. In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–20.

Holton, Judith A. 2007. The coding process and its challenges. *The Sage handbook of grounded theory*, 3: 265–289.

Huesmann, L Rowell, Leonard D Eron, and Patty W Yarmel. 1987. Intellectual functioning and aggression. *Journal of personality and social psychology*, 52 (1): 232.

Humphrey, Gayl, Rosie Dobson, Varsha Parag, Marion Hiemstra, Stephen Howie, Samantha Marsh, Susan Morton, Dylan Mordaunt, Angela Wadham, and Chris Bullen. 2021. See How They Grow: Testing the feasibility of a mobile app to support parents' understanding of child growth charts. *PLoS One*, 16 (2): e0246045.

Jaiswal, SANDEEP KUMAR. 2017. Role of parental involvement and some strategies that promote parental involvement. *Journal of International Academic Research for Multidisciplinary*, 5 (2): 95–104.

Kafai, Yasmin B and Quinn Burke. 2014. *Connected code: Why children need to learn programming*. MIT press.

Kai, Jiang. 2012. The origin and consequences of excess competition in education: A mainland Chinese perspective. *Chinese Education & Society*, 45 (2): 8–20.

Kelty, Noel E. and Tomoko Wakabayashi. 2020. Family Engagement in Schools: Parent, Educator, and Community Perspectives. *Early Childhood Education Journal*, 48 (5): 561–572. https://doi.org/10.1007/s10643-020-01012-2

Khalife, Joseph T. 2006. Threshold for the introduction of programming: providing learners with a simple computer model. In: *28th International Conference on Information Technology Interfaces, 2006.*, pp. 71–76.

Knox, P, Amber Simpson, and Adam V Maltese. 2023. Fortitude in frustration, failure: Exploring emotional responses within an at-home elementary engineering program. In: *American Society for Engineering Education*.

Kraft, Matthew A and Todd Rogers. 2015. The underutilized potential of teacher-to-parent communication: Evidence from a field experiment. *Economics of Education Review*, 47: 49–63.

Lavigne, Heather J, Ashley Lewis Presser, Deborah Rosenfeld, Leslie Cuellar, Regan Vidiksis, Camille Ferguson, Marisa Wolsky, and Jessica Andrews. 2023. Computational thinking with families: Studying an at-home media intervention to promote joint media engagement between preschoolers and their parents. *Early Childhood Research Quarterly*, 65: 102–114.

Lee, Charlotte P and Drew Paine. 2015. From The matrix to a model of coordinated action (MoCA) A conceptual framework of and for CSCW. In: *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pp. 179–194.

Lin, Janet Mei-Chuen and Shu-Fen Liu. 2012. An investigation into parent-child collaboration in learning computer programming. *Journal of Educational Technology & Society*, 15 (1): 162–173.

Lodi, Michael and Simone Martini. 2021. Computational thinking, between Papert and Wing. *Science & education*, 30 (4): 883–908.

Malone, Debra. 2017. Socioeconomic status: a potential challenge for parental involvement in schools. *Delta Kappa Gamma Bulletin*, 83 (3).

Mandinach, Ellen B., Edith S. Gummer, and Kavita Kapadia Matsko. 2020. Parental Educational Decision Making: The Information They Seek and What They Want from Data Systems. *Teachers College Record*, 122 (1): 1–34. https://eric.ed.gov/?id=EJ1265754 🏛

McHugh, Mary L. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22 (3): 276–282.

Nouri, Jalal, Lechen Zhang, Linda Mannila, and Eva Norén. 2020. Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11 (1): 1–17.

Noy, Chaim. 2008. Sampling knowledge: The hermeneutics of snowball sampling in qualitative research. *International Journal of social research methodology*, 11 (4): 327–344.

Papert, Seymour. 1981. Computers and computer cultures. *Creative Computing*, 7 (3): 82–92.

Park, Hyunjoon. 2008. The varied educational effects of parent-child communication: A comparative study of fourteen countries. *Comparative education review*, 52 (2): 219–243.

Parker, Charlie, Sam Scott, and Alistair Geddes. 2019. Snowball sampling. *SAGE research methods foundations*.

Payne, Bryson. 2015. *Teach your kids to code: A parent-friendly guide to Python programming*. No Starch Press.

Pezdek, Kathy, Tiffany Berry, and Paul A Renno. 2002. Children's mathematics achievement: The role of parents' perceptions and their involvement in homework. *Journal of Educational Psychology*, 94 (4): 771.

Porras, Jari, Antti Knutas, Jouni Ikonen, Ari Happonen, Jayden Khakurel, and Antti Herala. 2019. Code camps and hackathons in education-literature review and lessons learned.

Resnick, Mitchel and David Siegel. 2015. A different approach to coding. *International Journal of People-Oriented Programming*, 4 (1): 1–4.

Robinson, Keith and Angel L Harris. 2014. *The broken compass: Parental involvement with children's education*. Harvard University Press.

Sáez-López, José-Manuel, Marcos Román-González, and Esteban Vázquez-Cano. 2016. Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97: 129–141.

Sapungan, Gina Madrigal and Ronel Mondragon Sapungan. 2014. Parental involvement in child's education: Importance, barriers and benefits. *Asian Journal of Management Sciences & Education*, 3 (2): 42–48.

Sheehan, Kelly J, Sarah Pila, Alexis R Lauricella, and Ellen A Wartella. 2019. Parent-child interaction and children's learning from a coding application. *Computers & Education*, 140: 103601.

Sheriff, Aviv, Rona Sadan, Yasmin Keats, and Oren Zuckerman. 2017. From smart homes to smart kids: Design research for CataKit. In: *Proceedings of the 2017 conference on interaction design and children*, pp. 159–169.

Si, Janice Jianing, Donglin Li, Qiuning Wang, Alicia Wanyi Liu, Xin Lin, Yujun Zhu, Xiaobo Zhou, April Yi Wang, and Kanye Ye Wang. 2026. Exploring the Impacts and Challenges of Vibe Coding Paradigm to Children's Programming Learning and Practices. In: *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pp. 1–24.

Stevenson, David L and David P Baker. 1987. The family-school relation and the child's school performance. *Child development*, pp. 1348–1357.

Stipek, Deborah J. 2001. Pathways to constructive lives: The importance of early school success.

Sui-Chu, Esther Ho and J Douglas Willms. 1996. Effects of parental involvement on eighth-grade achievement. *Sociology of education*, pp. 126–141.

Toys, Primo ToysPrimo. 2017. *Beginning computer programming for kids*.

Tsukamoto, Hidekuni, Yasuhiro Takemura, Hideo Nagumo, Isamu Ikeda, Akito Monden, and Ken-ichi Matsumoto. 2015. Programming education for primary school children using a textual programming language. In: *2015 IEEE Frontiers in Education Conference (FIE)*, pp. 1–7.

Turner, Sherri and Richard T Lapan. 2002. Career self-efficacy and perceptions of parent support in adolescent career development. *The career development quarterly*, 51 (1): 44–55.

Turney, Kristin and Grace Kao. 2009. Barriers to school involvement: Are immigrant parents disadvantaged? *The Journal of Educational Research*, 102 (4): 257–271.

Vera, Elizabeth M, Marla Susman Israel, Laura Coyle, Joanna Cross, Laura Knight-Lynn, Isabel Moallem, Gina Bartucci, and Nancy Goldberger. 2012. Exploring the educational involvement of parents of English learners. *School Community Journal*, 22 (2): 183–202.

Walker, B and Maggie MacLure. 2005. Home-school partnerships in practice.

Wang, Siyu, Janice Jianing Si, Huanghuang Liang, Chuang Hu, Yujun Zhu, Xiaobo Zhou, Kanye Ye Wang, and Dazhao Cheng. 2025. Understanding the Challenges Students Face in Non-English Programming Environments Due to the Programming Language Transition: A Case Study of Keywords in the Chinese Version of Scratch. In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–18.

Yamamoto, Yoko and Susan D Holloway. 2010. Parental expectations and children's academic performance in sociocultural context. *Educational Psychology Review*, 22 (3): 189–214.

Yen, Kate, Yeqi Chen, Yi Cheng, Sijin Chen, Ying-Yu Chen, Yiran Ni, and Alexis Hiniker. 2018. Joint media engagement between parents and preschoolers in the US, China, and Taiwan. *Proceedings of the ACM on human-computer interaction*, 2 (CSCW): 1–19.

Yu, Junnan, Chenke Bai, and Ricarose Roque. 2020. Considering parents in coding kit design: Understanding parents' perspectives and roles. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–14.

Yu, Junnan, Julisa Granados, Ronni Hayden, and Ricarose Roque. 2021. Parental Facilitation of Young Children's Technology-based Learning Experiences from Nondominant Groups During the COVID-19 Pandemic. *Proceedings of the ACM on Human-Computer Interaction*, 5 (CSCW2): 1–27.

Yu, Junnan, Sari Widman, and Ricarose Roque. 2023. Family Negotiation in Joint Media Engagement with Creative Computing. In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pp. 1–15.

Zhang, Zheng, Ying Xu, Yanhao Wang, Bingsheng Yao, Daniel Ritchie, Tongshuang Wu, Mo Yu, Dakuo Wang, and Toby Jia-Jun Li. 2022. Storybuddy: A human-ai collaborative chatbot for parent-child interactive storytelling with flexible parental involvement. In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–21.

Zhao, Xu. 2015. *Competition and compassion in Chinese secondary education*. Springer.

A Exclusion Method

Before the interview, we guided potential participants to answer the following questions to determine whether they have been involved in their children's programming learning.

- What is your age? What is your occupation?
- What is your child's gender? What is his/her age?
- Has your child ever engaged in or is currently engaged in programming learning?
- How long has your child been learning programming?
- Which programming languages has your child learned?
- What resources have you provided for your child's programming learning?

We employed an exclusion method, where parents who expressed interest but had only recently been introduced to the concept of children's programming—without actual involvement—were excluded from the study. However, potential participants with limited experience who are progressively deepening their engagement in children's programming were retained.

B Interview Protocol

We used the following interview protocol as a guiding framework for semi-structured interviews, allowing flexibility in question order and follow-up to accommodate participants' experiences.

Opening Thank you very much for taking the time to participate in our user research. My name is ***, and I am a researcher at ***. Our study aims to understand parents' perceptions, involvement, and influence regarding their children's programming education. Throughout the interview, I will ask you a series of questions. Please remember, there are no right or wrong answers; feel free to respond based on your actual situation. If there are any questions you find inconvenient to answer, you can skip them. You can also ask questions or pause the interview at any time, and this will not affect your participation compensation.

We are very interested in your unique experiences and opinions. Please rest assured that your identity will remain confidential, and your real name will not appear in any of our publications or presentations. Our interview will serve as data for subsequent research, so it needs to be recorded. However, the recording will not be made public, and there will be no name identifiers. Do you agree to the recording and to start the interview?

- What is your child's age?
- What is your child's gender?
- What grade is your child currently in?
- Which programming languages have your child specifically learned? How long has each been studied?
- What is your age group?
- What is your field of work?
- What is your level of education?
- Have you ever studied programming yourself?
- Where do you get your information about children's programming?
- What is the main reason your child is learning programming? What is his/her willingness? What is your willingness?
- Do you often discuss programming-related topics with your child? Are you willing to learn more in-depth to help him/her solve problems?
- Has your child shared his/her experiences of learning programming with you? What was his/her feedback?
- Do you check if your child understands various concepts, such as sequences, loops, events, conditions, operators, and data?
- Do you observe your child's daily programming practice? What do you focus on?
- Is your child curious about what programming can be used for? Or do you pay attention to whether she has changed his/her perspective and actively tells you what programming can do after learning it?

- Do you often communicate with school teachers? What aspects of your child's situation at school are you mainly interested in?
- Do you frequently communicate with other parents of your child's classmates about their children's programming learning? What aspects do you mainly discuss?
- Have you provided your child with additional programming-related learning resources, such as textbooks, online courses, etc.? How do you usually obtain them?
- Are you aware of any other learning resources?
- Do you feel any financial pressure from the institution's courses?
- Are you willing to encourage your child to participate in programming competitions or activities?
- Are you willing to help your child solve problems when he/she encounters difficulties in learning programming? What kind of help do you provide when he/she faces difficulties?
- What are your expectations for your child's programming learning?
- What personal growth do you hope your child will achieve through learning programming? (Innovation and creativity, imagination, abstract thinking, hands-on ability, problem-solving skills, cooperation skills, expression skills, stress resistance, frustration resistance, thinking ability, concentration, computer skills, mathematical ability, algorithm level)
- Has the learning met your expectations? What progress have you observed?
- How do you evaluate your child's progress?
- In which areas do you hope for better improvement?
- Has your child encountered any challenges or difficulties while learning programming?
- In what areas do you think your child has made progress after learning programming? What behaviors or indicators lead you to this judgment? How do you evaluate your child's progress?
- Do you hope for further improvement in your child's abilities through continued learning? Specifically, what abilities do you want to enhance?
- Do you think learning programming could have any negative impacts on your child?
- Has your child ever shown a lack of interest in learning programming? How did you handle it as a parent?
- Has your child ever experienced any negative emotions? What specific problems did he/she encounter?
- When considering whether to continue your child's programming education, what factors do you mainly consider? How do you evaluate her learning performance? How do you measure programming skills?
- Have you ever been conflicted about whether to let your child continue learning programming?

- Have you ever experienced any negative emotions yourself? In what specific scenarios?
- Have you ever felt at a loss? What measures did you take?
- Are you worried that learning programming will take up too much of your child's time and affect his/her other studies or activities?
- Have you considered the difficulty of programming? Are you concerned that the increasing difficulty will affect his/her enthusiasm for learning?
- Have you considered your child's learning pressure? Are you worried that the pressure of learning programming will have a negative impact on him/her?
- Are you concerned that this learning method will turn programming into a test-oriented subject rather than skill development?
- Are you worried that your child will only focus on exams and not on innovation and creativity? How do you guide him/her? Are you aware of any resources that can help them practice?