What We Talk About When We Talk About K-12 Computing Education

Carsten Schulte*
carsten.schulte@uni-paderborn.de
Paderborn University
Paderborn, Germany

Sue Sentance*
ss2600@cam.ac.uk
University of Cambridge
Cambridge, United Kingdom

Sören Sparmann*
soeren.sparmann@uni-paderborn.de
Paderborn University
Paderborn, Germany

Rukiye Altin
ral@informatik.uni-kiel.de
Kiel University
Kiel, Germany

Mor Friebroon-Yesharim
mor.friebroon@weizmann.ac.il
Weizmann Institute of Science
Rehovot, Israel

Martina Landman
martina.landman@tuwien.ac.at
TU Wien
Vienna, Austria

Michael T. Rücker
michael.ruecker@uni-jena.de
Friedrich-Schiller-Universität Jena
Jena, Germany

Spruha Satavlekar
spruhasumukh@iitb.ac.in
Indian Institute of Technology Bombay
Mumbai, India

Angela Siegel
siegel@dal.ca
Dalhousie University
Halifax, Canada

Matti Tedre
matti.tedre@uef.fi
University of Eastern Finland
Joensuu, Finland

Laura Tubino
laura.t@deakin.edu.au
Deakin University
Melbourne, Australia

Henriikka Vartiainen
henriikka.vartiainen@uef.fi
University of Eastern Finland
Joensuu, Finland

J. Ángel Velázquez-Iturbide
angel.velazquez@urjc.es
Universidad Rey Juan Carlos
Madrid, Spain

Jane Waite
jane.waite@raspberrypi.org
Raspberry Pi Foundation
Cambridge, United Kingdom

Zihan Wu
ziwu@umich.edu
University of Michigan
Ann Arbor, United States

Abstract

K-12 computing education research is a rapidly growing field of research, both driven by and driving the implementation of computing as a school and extra-curricular subject globally. In the context of discipline-based education research, it is a new and emerging field, drawing on areas such as mathematics and science education research for inspiration and theoretical bases. The urgency around investigating effective teaching and learning in computing in school alongside broadening participation has led to much of the field being focused on empirical research. Less attention has been paid to the underlying philosophical assumptions informing the discipline, which might include a critical examination of the rationale for K-12 computing education, its goals and perspectives, and associated inherent values and beliefs. In this working group, we conducted an analysis of the implicit and hidden values, perspectives and goals underpinning computing education at school in order to shed light on the question of what we are talking about when we talk about K-12 computing education. To do this we used a multi-faceted approach to identify implicit rationales for K-12 computing education and examine what these might mean for the implemented curriculum. Methods used include both traditional and natural language processing techniques for examining relevant literature, alongside an examination of the theoretical literature relating to education theory. As a result we identified four traditions for K-12 computing education: algorithmic, design-making, scientific and societal. From this we have developed a framework for the exemplification of these traditions, alongside several potential use cases. We suggest that while this work may provoke some discussion and debate, it will help researchers and others to identify and express the rationales they draw on with respect to computing education.

*Working group leaders

ITiCSE-WGR 2024, Milan, Italy

ACM ISBN 979-8-4007-1208-1/24/07
https://doi.org/10.1145/3689187.3709612

CCS Concepts

• **Social and professional topics → K-12 education**; *Computing education.*

Keywords

computing education, K–12 education, philosophy, curriculum, educational traditions, rationales

ACM Reference Format:
Carsten Schulte, Sue Sentance, Sören Sparmann, Rukiye Altin, Mor Friebroon-Yesharim, Martina Landman, Michael T. Rücker, Spruha Satavlekar, Angela

Siegel, Matti Tedre, Laura Tubino, Henriikka Vartiainen, J. Ángel Velázquez-Iturbide, Jane Waite, and Zihan Wu. 2025. What We Talk About When We Talk About K-12 Computing Education. In *2024 Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR 2024), July 8–10, 2024, Milan, Italy*. ACM, New York, NY, USA, 32 pages. https://doi.org/10.1145/3689187.3709612

1 Introduction

In the rapidly evolving field of K-12 computing education, current debates and developments highlight the dynamic nature of the discipline, for example, recent discussions on integrating AI into school curricula. A report from the US National Research Council relating to scientific research in education states: "education is highly contested [as a field] because values play a central role: people's hopes and expectations for educating the nation's young are integrally tied to their hopes and expectations about the direction of society and its development" [115, p.17]. Decisions regarding why and what to teach with relation to computing at school are often thus made not on a scholarly basis, "but rather are derived directly from ideology or deeply held beliefs about social justice or the good of society in general" [115, p.17]. This is ultimately a moral and political matter and reasonable people can and will see matters differently. Hence, one can expect to see different approaches to computing education in different educational systems, or even in different schools. Moreover, progress might not be straightforward, but take the form of different parallel developments, possibly with only limited or unsystematic exchange of rationales, values and beliefs, between them.

This paper is structured as follows. After the introduction section, which provides the frame and scope of the work, we firstly consider existing work in K-12 computing education and education more generally by describing potential rationales (reasons) for teaching it and approaches to implementation (Section 2). These approaches and rationales were then used in a literature review whereby a selection of research papers were analysed through the lens of curriculum analysis (Section 3). The next section of the paper details how Natural Language Processing (NLP) techniques were used to uncover more implicit rationales for teaching K-12 computing (Section 4), followed by an analysis of these findings in the light of theories of education and learning. This leads to a description of four traditions underpinning K-12 computing education (Section 5), which form the basis of the framework for researchers (Section 6). Finally, the discussion section draws everything together to try to explicate 'What we talk about when we talk about K-12 computing education'.

1.1 Uncovering values and beliefs

In their analysis of the different rationales for computing in school, McGarr and Johnston [77] highlighted a range of different motivators, including economic development, equity and social justice, as well as the need to foster logical thinking and problem-solving skills. They argue that whatever the rationale is, it will influence policy decisions, for example, whether computing is taught as a discrete subject or integrated, whether it is for all or for some, and whether it is introduced at the primary level [77]. For example, a recent policy report from Brookings [121] argued that there was an economic benefit to both individuals and countries in studying computing, and that computing education in school could narrow the gap between lower and higher income countries. While some of the more critical examinations of computing education highlight, for example, that an economy-driven framing contrasts with socialisation-oriented perspectives of teachers [79], or that computing education embeds values of self-enhancement [128], or that the tech industry has an excessive influence on education policy [75]. Also, researchers have argued over time that the focus of computing teaching should be on principles (rather than artefacts) [70], artefacts (alongside principles) [127], or modelling [49].

The language used to describe computing education may reveal underlying values and beliefs. For example, a substantial body of work uses the phrase *computational thinking* (CT) to illustrate computing education's role in developing problem solving skills. Other loaded terms include *computational empowerment* [30], *computational literacies* [53], *computing for social good* [38][1] and *data agency* [112], all at some level value-based. Many of these are driven by the need for equity; again this term is interpreted and implemented differently [97]. The underlying values and beliefs may remain implicit.

1.2 Aims of the research

In K-12 computing education, our approach to teaching is shaped by the unique nature of the subject. Unlike science, where we recognise that our understanding can be challenged and updated as new discoveries are made, computing often feels more concrete because the topics are created and defined within the field itself. This can lead to a belief that our understanding of what should be taught is fixed and clear. However, this confidence may obscure the fact that, just as in science, the content and aims of computing education can evolve. In addition, educators themselves may have different perspectives on what is most important to teach and what the goals should be. Differing views among educators, combined with ongoing changes in the discipline and differing societal needs, suggest that we need to remain flexible and open to revising our teaching content and objectives to reflect new developments and different points of view.

There is a need to discuss what we mean by K-12 computing education –including underlying belief structures and hidden and implicit assumptions and goals of our teaching. The complexities of the situation include:

(1) There are different conceptions of what K-12 education should accomplish, leading to different rationales for teaching computing.
(2) The curriculum emphases of educators differ.
(3) There are different views on theories and learning, leading to different pedagogical approaches to teaching.
(4) There are different views about what computing is.

Our intent is to create a framework to allow people to better position themselves and to make their position explicit. By providing a language for 'What we talk about when we talk about computing education' we will not only allow but also promote the development and refinement of different computing education approaches and the accumulation of research findings in K-12 computing education research. This framework may be used by teachers, teacher trainers,

[1] Awarded best ITiCSE-WG report in 2012

resource developers, computing and associated curriculum designers, researchers and policymakers. Framing the discourse will assist K-12 computing education researchers in adopting positionality and in presenting their empirical work in a broader context.

1.3 Framing the Study: Positionality, Scope, and Terminology

In this subsection, we present a statement of *our positionality* as an ITiCSE working group, provide definitions of key terminology, and offer a broad overview of the paper's scope and approach. By addressing these elements, we aim to enhance the clarity and credibility of our work, helping readers better understand the background and perspectives that inform our analysis.

Our position is that the primary focus of computing education research often centres on *how* to teach –developing effective methods, tools, and strategies for learning. However, even when the explicit emphasis is on how to teach, research papers often address broader questions of *why* computing is taught —at least implicitly and often explicitly, particularly in their introductory sections.

To better analyse how different educational objectives may shape research in computing education, it is important to balance the discussion of rationales. While it may not be feasible to cover every rationale comprehensively, considering a range beyond the most prominent reasons may provide a broad enough framework for analysing the rationales used in computing education research and exploring how they influence key aspects of research, including methods, questions, and outcomes. We also faced the challenge of determining how to capture and frame these relevant aspects. To this end, we chose to draw on curriculum research [4] as a lens for analysis (see Figure 1).

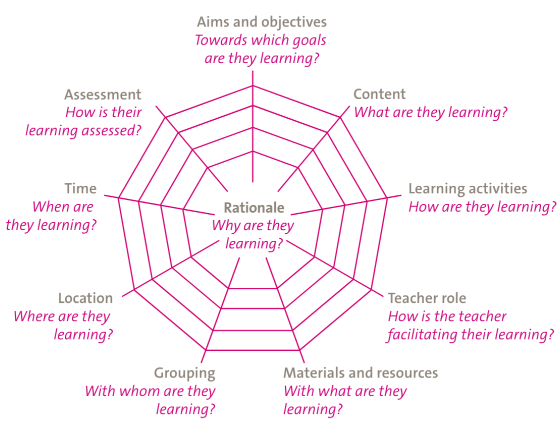

Figure 1: Curricular spider web, which addresses curriculum-related questions [118, p.59].

1.3.1 ***Positionality.*** In line with the collaborative ethos of ITiCSE working groups, our team comprises members from diverse academic institutions and countries. This diversity brings a range of perspectives to our discussions. Within the working group was considerable expertise and experience relating to K-12 education research, K-12 computing education, resource development and curriculum design, which we drew on in making initial decisions about established approaches to computing education.

Our group adopts the view on curriculum as expressed in [4]. Seen from this viewpoint, research in computing education is not only or even not merely about building a theory of learning computing (although this is an important aspect, see [84, 110]), but to put such theoretical insights into practice and, for example, to contribute to evidence-based curriculum reform. This could mean several things because a curriculum impacts several aspects of teaching and learning, see Figure 1.

At the core of the 'curricular spider web' lies the rationale: the answer to *why* students are learning. This concept goes beyond theory, shaping curriculum design by guiding decisions about content, pedagogy, and assessment.

Curriculum theory defines a curriculum as a 'plan for learning' (see [4, p.9], citing Taba, 1962), encompassing different levels and contexts. The rationale links all components, ensuring coherence and consistency [4, p.12]. Disrupting one element of this 'spider web' can destabilize the entire structure. As Tyler points out [1949, cited in 4, p.14], the rationale anchors educational priorities –knowledge preservation, social preparation, and personal development– and balances these dimensions with philosophical and psychological considerations. Similarly, Biesta's framework [9], discussed further in 5.1.1, emphasizes the interplay of these functions.

1.3.2 ***Scope: K12, but probably beyond.*** This positionality, and the background of curriculum research, frames the scope of our study. By analysing academic research papers, we uncover how rationales influence decisions within the field, making implicit assumptions explicit. This analysis aims at providing insights into the competing priorities that shape educational objectives. Without explicit attention to rationale, research risks misalignment; what seems logical from one perspective may appear counter-intuitive from another. For example, emphasizing computing for problem-solving skills may lead to different curricular priorities than framing it for digital citizenship or creative expression. This misalignment limits the field's ability to design coherent and sustainable curricula.

Analysing computing education research from a curricular perspective does not directly include learners' voices –that would be a different and presumably very useful endeavour aimed at subjective experiences. Instead we explore structural coherence and theoretical alignment in K-12 computing education research. We expect that academic research papers systematically address how educational systems balance the demands of academic knowledge, societal needs, and personal development, making them a critical focus for understanding and improving curricular design. This work aims to reduce implicit misunderstandings stemming from divergent implicit rationales and to align research contributions with broader curricular innovation efforts. By reflecting on their assumptions, computing education researchers can enhance their contributions to both pedagogy (in terms of what and how to teach) and the overarching rationale of education.

These considerations of rationale are especially crucial in the K–12 context, where societal decisions in many countries require all young generations to learn computing. In higher education, the question of rationale may seem less pressing, since learners voluntarily choose to study computing. Yet even then, reflecting on rationale can help avoid misunderstandings and align teaching with broader educational objectives.

At times, the distinction between rationale and goals or aims becomes blurry, making their analysis more complex. Goals describe what learners can *do* with the knowledge they acquire, focusing on outcomes, while rationale explains *why* they should acquire that knowledge in the first place. For instance, Jeanette Wing's [126] seminal paper on CT can be interpreted in two ways: as a rationale, it argues that CT represents a set of universally useful reasoning practices essential for navigating the modern world, thereby justifying why everyone should learn it. Alternatively, it can be seen as defining goals, with CT framed as a skill set learners can acquire through programming education to achieve specific problem-solving or analytical outcomes. This example underscores the importance of clarity in discussing research, as conflating rationale with goals may obscure the deeper reasoning behind curricular decisions and hinder coherent curriculum design.

Beyond identifying rationales as isolated priorities, it is important to recognize that curricula often embody a mix of priorities. This report will propose a framework for these mixed rationales, drawing from both educational traditions and different perspectives within computing itself. Specifically, we will outline four *educational traditions*: (i) the theoretical and mathematical tradition, emphasizing formal reasoning and foundational principles; (ii) a technological and engineering tradition, which focuses on practical applications and problem-solving; (iii) a scientific tradition, treating computing as a natural science concerned with inquiry and discovery; and (iv) a societal tradition, highlighting computing's role in addressing societal challenges and fostering citizenship. By analyzing these traditions and their implications, we aim to provide a nuanced view of how mixed rationales shape curricular design and how K-12 computing education research can align more effectively with these diverse traditions to support innovation and coherence in the field.

1.3.3 *Terminology*.

Next, we define some central terminology for our work, to enhance clarity.

A key distinction must be drawn between the rationale and related curriculum components, such as aims and objectives, and the content of learning. The rationale asks the foundational question: *why are they learning?* It provides the central justification for including a subject in the curriculum and serves as the 'core link' connecting all other components, such as learning activities, assessment, and teacher roles [4]. In contrast, aims and objectives focus on *towards which goals are they learning?* –these articulate the desired outcomes of education in terms of skills, knowledge, or dispositions. Content, meanwhile, addresses *what are they learning?* –the specific knowledge, concepts, and practices learners engage with.

Rationale. We define a 'rationale' as a coherent set of reasons and justifications linking K–12 computing education to broader values and goals. A rationale establishes why and how including computing at the K–12 level leads to something *desirable.* Such arguments often draw on diverse perspectives —technological, societal, or economic— to explain why it is important and beneficial for students to acquire computing knowledge and skills. As van den Akker [118, p.58] succinctly puts it, the rationale boils down to the question "Why are they learning?". Importantly, this is distinct from learning objectives, which specify the knowledge, skills, or dispositions students are expected to achieve.

Goal (learning objective or aim). We define 'aims and objectives' as the intended outcomes of education, describing what learners are expected to achieve through engagement with the curriculum. These terms are often used interchangeably to refer to the goals that guide teaching and learning, whether broad and aspirational or specific and measurable. In K-12 computing education, aims and objectives might include fostering CT, developing programming skills, or understanding the societal impact of technology. By providing a clear direction for learning, they serve as a bridge between the overarching rationale and the specific content.

Content. We define 'content' as the body of knowledge, concepts, and practices that learners engage with during the educational process. Content encompasses what is taught, including foundational principles, subject-specific skills, and practical applications. In the context of K-12 computing education, content might include topics such as algorithms, programming languages, CT, and the societal implications of technology. Selecting appropriate content ensures that learners acquire the competencies necessary to meet the aims and objectives, while also aligning with the overarching rationale.

Values. We define and understand 'values' as fundamental principles or ideals that guide decisions and priorities in education. Values represent what is inherently desirable, placing things, actions, or goals "on the approval-disapproval continuum" [57, p.395], a process rooted in social and cultural contexts. As such, values are not objectively right or wrong but are shared –or not– by individuals and groups. According to Biesta [10], educational practices are always value-laden, as they are constituted by *desirable* outcomes and *approvable* means. In K-12 computing education, values may emphasize equity, innovation, creativity, collaboration, or the ethical use of technology, providing a normative basis for shaping rationales, objectives, and content.

Beliefs. We define 'beliefs' as personal or collective perceptions and assumptions about teaching, learning, the nature of K-12 computing education, and also the nature of the academic discipline of computing. Beliefs are shaped by experiences, knowledge, and cultural contexts and influence how educators interpret and implement curricular decisions. For instance, a teacher's belief in the importance of hands-on learning may shape their approach to instructional activities, while societal beliefs about the role of technology in the future workforce may inform broader educational goals. Unlike values, which are normative, beliefs are descriptive and reflect subjective understandings of what is true or effective.

Approach. We define an 'approach' to K-12 computing education as a coherent framework of curricular decisions encompassing elements of the curriculum spider web; such as rationales, objectives, content, activities, teacher roles, and more. As noted by van den Akker [118], these decisions are deeply interconnected, such that a change in one component often necessitates adjustments in others to maintain the integrity of the "vulnerable curricular spider web" [118, p.57]. As illustrated in Figure 1, the rationale serves as a pivotal element, "connecting all other curriculum components" [118, p.1]. While not every approach explicitly addresses all the peripheral components in Figure 1, the underlying rationale is always present, either stated or implied.

Educational Tradition. Educational tradition is a term we coined in this study to better and more versatile name and capture some central results. Different values and beliefs form different approaches. Such approaches are kind of reactions and answers to their sometime implicitly held beliefs and values. But they are not isolated, as there is a discourse surrounding them and their development. And such a discourse, over time, can be seen as forming an educational tradition, in which some shared vision is pursued and expressed in different yet similar approaches and assumptions. That is, an educational tradition in computing education research (and in the broader discourse on computing education) is defined as a broad but coherent set of foundational assumptions about the very nature of computing education. Although the focus is on K-12 computing education, we anticipate that these traditions are useful in the analysis and description of the discourse on computing education in higher education. The traditions have some overlap, but are also distinct. The term 'tradition' denotes the traditions of computing itself, because the educational traditions also represent a distinct point of view on the relationship between the school or educational subject, and the academic research discipline; they express a certain view of what kind of discipline it is that is being taught (in school). A deeper discussion on how such traditions are related, and in what aspects they differ, can be found in [109].

1.4 Research Questions

In this paper we address the following three research questions:

- RQ1 What characteristics of a perceived implemented curriculum can be identified within K-12 computing education research? (observable)
- RQ2 What are the rationales that are being used to justify the teaching of computing at the K-12 level? (detectable)
- RQ3 What are the values and beliefs that underlie the rationales for teaching computing at the K-12 level? (hidden or implicit)

A visual representation of how the research questions relate to values and beliefs, rationales and approaches is given in Figure 2. Three different methodologies were adopted: scoping literature review (RQ1), literature analysis using Natural Language Processing (NLP) techniques (RQ2) and theoretical synthesis with relation to the findings of RQ1 and RQ2 (RQ3). By focusing firstly on what is observable in the research literature, we move towards what is hidden or implicit (see Figure 2).

2 Background on rationales and approaches

2.1 Rationales for K-12 computing education

Differing values and beliefs give rise to different rationales for learning. In the context of K-12 education, such rationales arguably need to go beyond reasons why learning computing is valuable to a select few (e.g., certain professionals or disciplines), but have to establish its potential value for everyone or to society at large. In the following, we summarise six different rationales for K-12 computing education identified within the working group. They were mainly extracted from two US reports, complemented with other sources. In 1999, an influential National Research Council (NRC) report [87] listed four rationales for computing education –which in that report was largely about using ICT– and then in 2010 another NRC report [21] extended the list to the six rationales below.

Personal Rationale. This rationale emphasises using computing and computing technology to enhance one's own life and to pursue one's own interests, whether exercising a hobby, improving one's own health, or engaging in political activism [21, 87]. It is connected to the idea of using computing as a means for personal expression and creativity [123, p.611], as well as notions of self-determination, empowerment, and the ability to use computing to shape one's own life and environment in personally meaningful ways [45, 114].

Career (or Workforce) Rationale. This rationale centers on improving students' career opportunities by strengthening their "competence, skills, and employability for the twenty-first-century job market" [88, p.9] (see also [87]). The training of future computing professionals may well be one of the oldest goals of computing education. Tedre et al. [111] trace it back to as early as the 1950s. While focused on tertiary education at that time, the notion has been carried over, with Guzdial [45] arguing, for instance, that K-12 students should learn computing "as a job skill" to address "the need for more programmers" but also to give students "new skills that have value in the economy" [45, p.34-5] .

National (or National Competitive) Rationale. This rationale focusses on national economic growth as a whole, to increase gross economic output and maintain national competitiveness [21]. While connected to the career rationale, we distinguish it as shifted less towards individual employability, and more toward the development of "a strong computing-savvy workforce" [125, p.30] and the improvement of "industry pipelines, and human capital" [123, p.610].

Educational Rationale. This rationale maintains that learning computing and being proficient with computing technology will open up additional educational and learning opportunities for students. It goes back at least to Papert's notion of 'mindstorms' [89] and the potential of computing technology to enable novel pedagogical practices [123] or to provide access "to an array of educational resources that were not previously accessible" [87]. While Guzdial describes that the dynamic nature of computer programs "gives us a powerful new way to learn science and mathematics" [45, p.33], Blikstein and Moghadam argue that computational literacy provides a "set of material, cognitive, and social elements that generate new ways of thinking and learning" and even "enables new types of mental operations and knowledge representations" [15, p.59]. In essence, this rationale values academic achievement and casts computing education as a corresponding driver. Finally, Wilson et al. [125, p.30] report that K-12 computing students "demonstrate improved readiness for post-secondary studies".

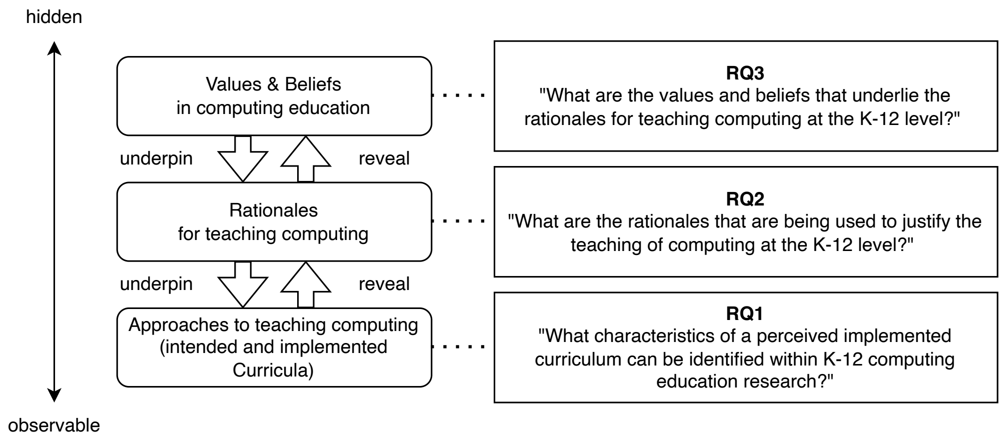

Figure 2: An overview of the research questions and how they relate to the values and beliefs, rationales and approaches in the context of K-12 computing education.

Societal Rationale. This rationale casts computing education as a facilitator for societal awareness, responsible citizenry, and informed democratic participation [87]. Our modern world is permeated by computing and its various everyday phenomena "shouldn't be magic" to students [45, p.34]. Informed, responsible and critical participation in social and political discourse requires a basic understanding of how computing impacts our everyday lives, for better and for worse, particularly with respect to social inequities like racism, sexism or classism [61, 123].

Scientific Rationale. This rationale argues that computational tools are required in an increasing number of research disciplines such that "computational thinking would assist specialists in those other disciplines" [87, p.4]. e.g., to model and simulate processes [45, p.32-3]. While sharing elements with the educational (inquiry) and career (professional skills) rationales, the scientific rationale emphasises the value of research progress rather than individual achievement or economic growth. Computing education is seen as a driver in "technological and scientific breakthroughs" that would benefit humanity as a whole, e.g., research into medical issues, climate change or technical innovations [123, p.12].

It is clear, e.g., with the national and scientific rationales, that it can be debated at what point two lines of reasoning become distinct enough to be regarded as two different rationales. Hence, we do not claim this list to be definitive and we certainly do not understand these rationales to be mutually exclusive. Individuals may endorse any number of them and, in fact, some other rationales that were put forward in the literature seem to rely on combinations of the above. As two cases in point, consider two of the rationales discussed by Blikstein and Moghadam [15, p.60-65]:

The "computational thinking rationale" [15, p.60] argues that computational problem-solving skills are highly transferable and universally applicable to numerous domains and areas of life. Skepticism about such high-level transfer notwithstanding [33], this may appear like a good-enough reason to learn such skills. However, it does not, in fact, satisfy our definition of rationale as it leaves open where exactly they ought to be applied and why that would be desirable. In the terms of the curricular spider web (Section 1.3), CT and problem-solving skills are instead better understood as learning objectives or content, and less as reasons for learning in the first place. Those reasons need to be found in subsequent areas of application, which presumably often correspond to those described above.

The "equity of participation rationale" [15, p.59] argues that computing is not equally accessible to all, particularly to women, minorities and low-income groups. Therefore, computing education at the K-12 level is important to "level the playing field" and close the digital divide [Grover cited in 15, p.65]. While this connects computing education to key values like equity, social justice and inclusion, it crucially presupposes that not learning computing will put people at a disadvantage. For instance, Blikstein and Moghadam [15, p.64] state that students excluded from computing education may struggle "to fully participate in twenty-first-century society" (societal rational) or to get "the best and most creative jobs" (career rational).

3 Scoping literature review (RQ1)

As detailed in Section 1.4 and illustrated in Figure 2, in order to answer the overarching question posed by the paper about what we mean when we talk about K-12 computing education, we started RQ1 by considering what is *observable* within a sample of research literature, moving through to RQ2 what could be perceived in a wider, more automated search, and finally conducted a theoretical analysis drawing on the results of RQ1 and RQ2. In this section, we describe the methods and results of RQ1:

RQ1: *What characteristics of a perceived implemented curriculum can be identified within K-12 computing education research? (observable)*

To answer this question, a scoping literature review was conducted. The working group set out to locate a sample set of papers which represented the range of approaches to K-12 computing education. The aim was to map a coherent set of rationales and curriculum decisions to these candidate approaches through consideration of papers discussing either rationales, approaches, or both.

3.1 Approaches to curriculum

Six candidates for overarching approaches to K-12 computing education were collectively and iteratively identified by the working group through a series of discussions. These provide a starting point, and while they could be argued not to fully encompass all the curriculum approaches that could be taken by researchers and practitioners, we felt that they represented a range of values and beliefs that could be potentially held by researchers. We name them 'approaches' in line with the definitions given in Section 1.3. The approaches explicate the intended, and ideal vision of a computing 'curriculum' (according to van den Akker [117, 118]). In accordance with the spider web framework, these approaches represent our attempt to distill distinguishable and coherent sets of curricular decisions related to K-12 computing education. As such, they represent somewhat abstracted ideas, which are certainly not mutually exclusive. However, we do believe that there are limits to the ways in which they can be productively combined in practice without introducing curricular inconsistencies and 'rupturing' the spider web.

Algorithmic Problem-Solving. This approach to K-12 computing education would emphasise computing as a discipline that is centred around algorithms, computation, and processes that transform information [46, 59]. The algorithmic focus can be seen in many descriptions of computational thinking [22, 24], which often focus on developing learners' ability to think algorithmically, for instance, breaking down complex problems into manageable parts, developing step-by-step procedures to solve those parts, and understanding the effectiveness and efficiency of those procedures [43]. Although it is often stressed that such cognitive abilities exist independently from programming, we hypothesise that programming will play a prominent role in this approach as a prototypical activity that requires and fosters them [106]. Yet there also exist many 'unplugged' activities aiming to foster algorithmic problem-solving [7].

Constructionist Design-Making. Building on the educational philosophy of constructivism, as supported by Piaget and others, Papert [89] developed the pedagogy of constructionism, which emphasizes the creation of personally meaningful computational artefacts in learning and has since had a marked influence on the K-12 computing education community. Physical computing and maker activities [14, 40, 98] seem especially influenced by Papert's notion of creating and sharing interactive "objects to think with" [1]. With a strong emphasis on fostering creativity, artistic expression and social interaction, educators take on the role of facilitators rather than instructors, facilitating and coaching students to reach their own goals, often in a project-based learning context.

Computational Empowerment. This approach departs from the core tenets of CT and merges them with a participatory design agenda [50]. In doing so, it aims to "shift focus from programming skills" towards the broader "means necessary to engage actively in technological development" [31, p.67]. Beyond purely technical or algorithmic issues, students are meant to also critically engage with different stakeholder perspectives, goals and values and make informed moral decisions with respect to a particular system design [32, 100]. Related activities often engage students with emerging and/or controversial technologies, such as machine learning, virtual reality or the internet of things, and have them deconstruct, experiment with and discuss the effects of different design decisions [103].

Computing for Social Good. This approach emphasizes a view of computing as a community of practice and focusses on sociocultural issues both as an important motivator for students as well as a means to decrease barriers for participation [38, 39, 80]. The overall aim is to make computing more accessible, particularly for underrepresented and marginalized groups, to foster students' interest and to have them understand and internalize that they, in fact, can participate in computing practices [19, 55, 86]. Computing is seen not only a discipline, but also as a community defined by the people who practice it, and questions of community and member identity are pivotal to the approach. Corresponding learning activities often have students work on projects that are directly related to their immediate local communities or individual sociocultural backgrounds.

Critical Computing. Critical or 'critically conscious' computing education views computing processes and products as inherently social entities, which incorporate, propagate and potentially force upon others the goals and values of their designers [60]. A central aims is to foster learners' 'critical consciousness', a term coined by Freire [35] and more recently defined by Godfrey and Rapa [37, p.2] as the ability to "analyze social realities critically" in order to assess and potentially change "conditions [that] limit access to opportunity and perpetuate injustice". Hence, learning about computing is rendered as learning –quite literally– about society's tools of oppression, and learning to shape technology thus becomes a means to enact broader social change.

Sociotechnical Approach. The sociotechnical approach focuses on integrating the technical aspects into a broader view of the social dimensions of an informatics system. This approach heavily draws on software-engineering [67]. It contrasts with other (traditional) approaches by not starting with a new project, but starting with so-called 'deconstruction' of a given project with subsequent further development of the software. In such deconstruction learners can explore the system but also e.g., the development process [73], thereby learning that in such processes different choices are possible, and for example that software and system development is not neutral to its targeted social and use context [72].

3.2 Method

As the searches we would need to undertake (initially six searches for approaches explicitly or implicitly present in a research paper, followed by a search for papers describing curriculum implementation) were going to be complex and likely to result in a large number of unwanted papers, we were not able to take an exhaustive systematic approach to searching the literature. We therefore conducted a a scoping review [83] that would elicit samples of the types of papers we were looking for, knowing that there would be an iterative process of inclusion and exclusion to find suitable papers for this purpose.

A six-stage method was developed to answer RQ1:

1. An initial search for papers aligning to the titles of the six approaches identified and highlighted in Section 3.1;
2. Analysis of a first set of papers to establish the extent to which they explicated aspects of curriculum approaches, and provided answers to curriculum questions;
3. Development of a coding system for papers for this analysis;
4. A second round of literature searches to identify papers describing curriculum implementations;
5. Re-analysis using the coding system developed in step 3, with consensus checking amongst researchers; and
6. Mapping of approaches and rationales to code frequency to reveal patterns in the literature.

3.3 Process in detail

Stage 1: Initial search. We developed six searches, one for each of the six candidate approaches that we have previously highlighted: algorithmic/problem-solving, computational empowerment, constructionist (design/making), computing for the social good, critical computing, and the sociotechnical approach. The timeframe was set to ten years as the initial pilot searches conducted from fifteen years of papers included many papers describing outreach programmes run in the years 2009-2014 when many countries did not have K-12 computing education in the curriculum.

The search was restricted to the ACM Digital Library as we were not aiming for an exhaustive search. Each query identified papers that highlighted school/K-12 curriculum content in the abstract and mentioned the name of the approach we were looking at, in the ACM Digital Library, within the last ten years (June 2014-June 2024). An example query for the computational empowerment approach is given below:

"query": Abstract:("K-12" || school) AND Fulltext:("computational empowerment") AND Abstract:(computing || "computer science" || informatics) "filter": E-Publication Date: (06/01/2014 TO *), ACM Content: DL

Variations of this method were used for the six different approaches, as some elicited larger numbers of papers than others. Searches were adapted to include 'curriculum' where there were many papers found. For the *algorithm/problem-solving* approach, so many papers were found that a random sample was taken, as both of the search terms were very common words in the K-12 computing education literature. Although we noted overlaps in use of the terms 'algorithm' and 'problem-solving' with 'computational thinking', we decided not to use the latter term in our searches as it appears in many papers in a very generic way.

While locating papers, the following inclusion/exclusion criteria were applied:

- Exclude: papers that are not research articles;
- Exclude: papers that are not focused on K-12 or on curriculum;
- Exclude: papers that do not mention the approach being searched for.

Duplicates were then removed, leaving 60 papers at this stage.

Stage 2: Analysis of initial papers. The next stage was to code all 60 papers deductively according to their approach and rationale, and inductively around the curriculum implementation.

Firstly, we coded the six approaches to K-12 computing education identified and highlighted earlier in Section 3.1 as the basis of the scoping review. We matched those approaches as explicit and implicit in each paper we read. The following methodology was used to determine the approaches:

- The approach is classified as explicit if the approach is:
  - written in the keywords.
  - mentioned in the introduction or background.
  - described even if the term is not used.
- The approach is classified as implicit if the researchers were able to judge that the sentiment of the approach was present in the text.

Secondly, we coded papers according to six rationales (personal, workforce, national competitiveness, educational, societal and scientific) [21, 87] (see Section 2.1). To code papers according to the rationales, a 'yes/no' coding indicated whether that rationale seemed to be at play in the writing of the paper.

Thirdly, we drew on van den Akker's list of components that address specific questions about the planning of student learning [117] to code the content of the curriculum as described in the paper. The spider web visualization of these components is used to arrange them as a spider web not only illustrating its many interconnections, but also underlining its vulnerability (see Figure 1).

Nine of van den Akker's original questions were used in this study, omitting the question about rationale (which we had already coded):

- Toward which goals are they learning?
- What are they learning?
- How are they learning?
- How is the teacher facilitating learning?
- With what are they learning?
- With whom are they learning?
- Where are they learning?
- When are they learning?
- How is their learning assessed?

These questions were answered for each of the set of chosen papers. Two researchers read each paper and wrote verbose descriptions to answer the curriculum questions. The answers to each question were then coded inductively and from this a set of codes was derived to be used for finding patterns across papers and across rationales.

Stage 3: Development of coding system. Following on from Stage 2, a set of codes was established to code future papers with respect to the curriculum questions we were interested in. This was achieved by two researchers working on each curriculum question to find a set of codes that described the verbose descriptions. After coding more papers, this code set was revised. The code set that was established is presented in Table 1. In the table, 'practicals' refers to lessons with non paper-based practical activities to complete –for example, programming, software development, unplugged activities.

Stage 4: Iteration of search. After developing the coding system, a further search was conducted to elicit a second set of papers that

Table 1: Coding system for literature review curriculum questions

Curriculum area	
Sub-Question	**Curriculum question**
Aims and objectives - Towards which goals are they learning?	
Computing learning goal	Computational concepts, Programming, Design concepts, Other, N/A
Societal learning goal	Promote social justice in computing, To have awareness of ethical and social issues, Understand the role of technology in society, Other, N/A
Personal learning goal	Develop criticality, Sustain interest, Develop problem-solving skills, Understand the role of technology for the individual, Develop creativity, Consider computing as a career, other, N/A
Content - What are they learning?	
Computing goals	Data, Engineering (STEM)/design, HCI and AI, AI, HCI, Computing areas- other, Data and AI, Robotics, Digital competence, Other, N/A
Programming goals	Programming, No programming, N/A
Transversal skills	Personal/social/cultural concerns, Transversal skills, Human concerns and transversal skills, Other, N/A
Learning activities - How are they learning?	
	Practicals, Writing, Research, Questions/Exercises, Mix, Other, N/A
Teacher role - How is the teacher facilitating learning?	
	Teacher directive, Teacher as facilitator, Student-led (inquiry), Other, N/A
Materials and resources - With what are they learning?	
	Programming environments, Hardware, Digital artifacts, Interactive digital resources, Other, N/A
Grouping - With whom are they learning?	
	Individual, Small group, Large group, Other, N/A
Location - Where are they learning?	
	Formal education discrete, Formal education in other subjects, Informal education (extracurricular but organised), Other, N/A,
Time - When are they learning?	
	Mandatory in school, Elective in school, Self-study, Other, N/A
Assessment - How is their learning assessed?	
	Projects, Feedback (verbal or written), Assessment test, Formal exams, Discussion, Other, N/A

detailed learning goals and lessons within the implemented K-12 computing curriculum. In this way, the final paper set included both those that mentioned one or other of the approaches or had details of curriculum implementation, which we felt would give us a representative set to answer the curriculum questions using a sample of papers from the literature. Through this search, another 72 papers were initially identified. After one researcher briefly scanned these papers to ascertain whether they could be used to answer the curriculum questions, 35 papers remained.

Stage 5: Recoding of combined set of papers with consensus checking. During this stage, six members of the working group were involved in re-coding both sets of papers deductively. Each paper was assigned one and only one code from each curriculum question, and 'other' if it did not fit any of the options in the codebook. Each paper was coded by at least two people. Further consensus checking amongst the working group led to a substantial further exclusions of papers, removing papers that did not explicitly give sufficient detail to answer the curriculum questions. Each of the final set of papers had enough detail within the paper to answer the majority of the curriculum questions needed by the analysis. The combined paper set of 95 papers (35 + 60) was hence reduced to 58 papers.

Stage 6: Synthesis of curriculum patterns according to approach. The result of this lengthy searching, coding and re-coding process was a set of tables, which show the frequency of occurrence of answers to the curriculum questions (codes) within each of the six approaches. The most frequently occurring answers were used as representative of each approach. The results will be shown in the next section.

3.4 Results: approaches and rationales

58 papers were finally included that provided example answers for the curriculum questions, and identified the prevalence of the different approaches and rationales.

In terms of rationales, Table 2 shows which of the six were identified most frequently. Note that most papers (66%) were coded under more than one rationale.

For the curriculum approaches, all six approaches were identified in the papers to a greater or lesser extent. Most papers were observed to relate to at least two approaches. The most commonly

Table 2: Occurrence of rationales in 58 manually coded papers

	Frequency	
Rationale	n	%
Personal	39	67%
Educational	30	52%
Societal	21	36%
Scientific	17	29%
Workforce	9	15%
National competitive	2	3%

observed was the algorithmic/problem-solving approach followed by the constructionist (design/making). Computing for social good was only noted explicitly or implicitly as an approach being adopted in nine of the papers.

Table 3: Occurrence of curriculum approaches in 58 manually coded papers

Curriculum approach	Exp	Imp	Either	%
Algorithmic/ Problem-solving	32	11	43	74.1%
Sociotechnical	9	4	13	22.4%
Constructionist	13	9	22	37.9%
Critical consciousness	7	5	12	20.7%
Computational empowerment	13	7	20	34.5%
Computing for social good	6	3	9	15.5%

The relationship between curriculum approaches and rationales is interesting to note. Figure 4 shows that the personal rationale was observed across all curriculum approaches. Not surprisingly, the workforce rationale was not observed in papers that took a critical consciousness or computing for social good approach, while the societal rationale was noted in almost all of the papers coded under one of those two approaches.

3.5 Results: curriculum implementation

To answer RQ1, it was necessary to find out how the adopted curriculum approaches were implemented in terms of the curriculum questions that had been our focus: for example, what the learning goals were, how the teacher facilitated learning, when and where learning took place, etc.

For example, it was observed that for the algorithmic/problem-solving curriculum approach, the learning goal was most likely to be programming, with the teaching taking place in school in a mandatory curriculum, with students who were learning problem-solving and the teacher was acting as a facilitator in the lessons. The learning was at the individual level, and there was not an obvious consensus about approaches to assessment.

In contrast, for the approaches that came from a more societal approach, the most frequent answers were rather different. The learning goals were most commonly computational concepts in general, but also to promote social justice in computing and develop the critical consciousness of students. Content might focus on data and AI as well as programming, and also cover personal, social or cultural concerns. Teaching was likely to be in a small group, facilitated by a mix of learning activities (which may include research, discussion, as well as practical projects). Assessment was most commonly reported as using projects or not mentioned at all, but discussion was also mentioned as a formative assessment tool in some (three) papers. However, it has to be noted that only 12 of the 58 papers were noted as observing the critical consciousness approach.

For a third example, the 20 papers noted as using the computational empowerment approach had similarities with the curriculum implementation of the algorithmic/problem-solving, with programming and practicals being common, although working in small groups was more commonly seen than working individually. These papers were most commonly observed in an informal learning context, with a goal of sustaining interest. Table 5 shows these differences.

For the other three approaches, the data from our coding exercise (perhaps surprisingly) showed the constructivist (design/-making) approach having a similar curriculum implementation to algorithmic/problem-solving lessons in the 22 papers included in our sample. The papers that indicated a computing for social good approach had similar answers to those coded as critical consciousness (and there was a significant overlap of papers). For the socio-technical approach, 13 papers were coded, and the most common responses included papers indicating that developing criticality and awareness of social justice was a learning goal, with engineering or STEM design within the content of the curriculum.

For the scientific and societal rationales, Table 6 shows the most common curriculum implementation.

3.6 Summary

In this section, we have described a manual scoping literature review with the intention of answering the first research question: 'What characteristics of a perceived implemented curriculum can be identified within K-12 computing education research?' To tackle this question, we used a variety of search criteria to find literature that seemed to evidence 'approaches' to curriculum and gave details of curriculum implementation.

Within these papers, 57% were coded as having more than one approach evident, with some having as many as four identified in one paper. 66% of the papers were coded as having more than one rationale observed in the paper. This provides evidence that the majority of researchers are not influenced by one rationale or curriculum implementation approach alone but have multiple perspectives on why and how we should implement computing education. While we may be influenced by one particular view about K-12 computing education, this does not need to be mutually exclusive with other views. However, as researchers, we may have a dominant perspective to a greater or lesser extent.

4 Identifying Rationales in K-12 Computing Education Literature Using NLP (RQ2)

In Section 3, we looked at various curricular approaches and how they relate to different rationales for justifying computing education in school. The manual examination of 58 papers showed that the personal and educational rationales were most commonly noted in

Table 4: Relationship between rationales and approaches in 58 manually coded papers (Percentages refer to the number of the papers for each approach that also are coded with the rationale given)

Rationale	Curriculum Approach					
	Algorithmic, Problem-solving	Sociotechnical	Constructionist	Critical Consciousness	Computational Empowerment	Computing for Social Good
Personal	70%	69%	64%	75%	75%	78%
Workforce	14%	23%	9%	0%	0%	0%
Educational	53%	54%	50%	33%	25%	11%
Societal	26%	62%	50%	92%	50%	89%
National Competitive	5%	0%	5%	0%	0%	0%
Scientific	33%	8%	18%	17%	20%	22%

Table 5: Results from the scoping literature review showing the most commonly coded curriculum answers for three of the approaches investigated: algorithmic problem-solving, critical consciousness and computational empowerment

Question	Most commonly mentioned		
	Algorithmic/problem-solving approach (n=43)	**Critical consciousness approach (n=12)**	**Computational empowerment approach (n=20)**
Towards which goals are they learning?			
– Computing learning goal	Programming	Computational concepts	Computational concepts
– Societal learning goal	N/A	Promote social justice in computing	N/A
– Personal learning goal	Develop problem-solving skills	Develop criticality	Sustain interest
What are they learning?			
– Computing goals	N/A	Data and AI	Computing areas- other
– Programming goals	Programming	Programming	Programming
– Transversal skills	N/A	Personal/social/cultural concerns	Personal/social/cultural concerns
How are they learning?	Practicals	Mix	Practicals
How is the teacher facilitating learning?	Teacher as facilitator	Teacher as facilitator	Teacher as facilitator
With what are they learning?	Programming	Programming	Programming
With whom are they learning?	Individual	Small group	Small group
Where are they learning?	Formal education discrete	Formal education discrete OR Informal education (extracurricular but organised)	Informal education (extracurricular but organised)
When are they learning?	Mandatory in school	Other	Other
How is their learning assessed?	N/A	Projects	N/A

these papers (see Table 2), and that the algorithmic/problem-solving curriculum approach was most commonly observed (see Table 3).

As discussed in Section 2.1, several efforts have been made to capture the rationales for K-12 computing education, often relying on human expertise or reviewing a compiled selection of literature [91]. However, there is still limited understanding of how these rationales are perceived and whether they are widely agreed upon within the research community.

To truly understand what we, as the (ACM) computing education community, talk about when we talk about K-12 computing education, a broader perspective on the computing education landscape is needed.

RQ2 *What are the rationales that are being used to justify the teaching of computing at the K-12 level? (detectable)*

This involves identifying the rationales commonly, explicitly or implicitly used in the literature and analysing their prevalence (over time). This data can provide insight into the values and beliefs that

Table 6: Results from the scoping literature review showing the most commonly coded curriculum answers for two of the rationales investigated: scientific and societal

Question	Most commonly mentioned **Scientific rationale (n=17)**	 **Societal rationale** (n=21)
Towards which goals are they learning?		
– Computing learning goal	Programming	Computational concepts
– Societal learning goal	N/A	Promote social justice in computing
– Personal learning goal	Develop problem-solving skills	Develop criticality
What are they learning?		
– Computing goals	N/A	Data and AI
– Programming goals	Programming	Programming
– Transversal skills	N/A	Personal/social/cultural concerns
How are they learning?	Practicals	Mix
How is the teacher facilitating learning?	Teacher as facilitator	Teacher as facilitator
With what are they learning?	Programming	Programming
With whom are they learning?	Individual	Small group
Where are they learning?	Formal education discrete	Formal education discrete
When are they learning?	Mandatory in school	Other
How is their learning assessed?	N/A	Projects

underpin K-12 computing education research by revealing what drives researchers in their shared pursuit of teaching computing to students. By leveraging NLP techniques, we can examine a large corpus of K-12 computing education literature, providing a more comprehensive and representative picture of the research community. In this section, we report on a large-scale, semi-automated literature review using NLP techniques to identify and analyse the rationales prevalent in the K-12 computing education community.

4.1 Background on NLP techniques for automated literature review

A rapidly increasing number of AI-driven tools leveraging NLP techniques, including the use of Large Language Models (LLMs), have been developed to assist with and partially automate systematic literature reviews (SLR). For a detailed overview of these tools, see [16].

A systematic literature review generally involves six phases [16, 56]: (1) the *planning phase*, which lays the foundation for the review by formulating precise and specific research questions to guide the process; (2) the *search phase*, aimed at identifying relevant studies using search strategies, snowballing, or a hybrid approach; (3) the *screening phase*, which applies inclusion and exclusion criteria to refine the set of papers obtained during the search; (4) the *data extraction and synthesis phase*, where relevant information is systematically extracted from the selected studies; (5) the *quality assessment phase*, which evaluates the rigour and validity of the included studies; and (6) the *reporting phase*, where findings are presented in a structured and coherent manner in a research paper. Most tools focus on the search, screening, data extraction and synthesis phases, as these are the most time-consuming and offer the greatest potential for automatisation.

For this study we applied two NLP techniques, involving the use of LLMs for the screening and data extraction and synthesis phases; we give a little more background on the techniques used.

The data extraction and synthesis phase involves extracting some kind of information related to the research question from a collection of literature, and then grouping and aggregating (synthesising) the results in an adequate way [104]. Both of these steps can be largely automated using NLP techniques.

To extract information from a document (or a collection of documents), one approach that has emerged recently is called Retrieval-Augmented Generation (RAG) [66]. RAG is an NLP technique that combines retrieval-based and generative approaches to automatically answer questions about one or more documents. It involves retrieving information relevant to answering the question from the collection of documents and using this retrieved-context as input to a text-generation model (usually an LLM) to answer the question [66]. The answer is, therefore, based on the context retrieved from the documents, which, in general, yields more precise outcomes and reduces the occurrence of hallucinations [36]. There are examples of RAG being used to find relevant information in collections of scientific literature in medical contexts [27, 69].

Topic modelling is a technique for clustering documents based on linguistic features. The most commonly used technique in the past has been Latent Dirichlet Allocation (LDA) [13]. LDA clusters documents based on the frequency of tokens within documents and weights them by their total frequency. This is done by calculating

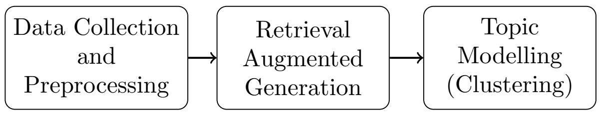

Figure 3: Schematic overview of the data processing pipeline used to identify rationales in K-12 computing education literature through NLP techniques.

the term frequency - inverse document frequency (TF-IDF). A disadvantage of this technique is that it does not take into account the semantic relationship between words and the order in which the tokens appear.

A more recent approach to topic modelling is to cluster documents based on document embeddings. These embeddings capture some of the semantic meaning of the text and also take into account the order of tokens. In this way, documents that have a similar meaning but use very different words are still considered to be closely related. One example for this technique is BERTopic, a topic modelling technique based on Bidirectional Encoder Representations from Transformers [41].

Both LDA and BERTopic have been used for reviewing literature in the past [5, 64, 71]. Although both techniques, RAG and Topic Modelling, are commonly used for different use cases, we did not find any literature that combined these two techniques to perform a literature review.

Although there are existing NLP methods to perform literature review, we are aware of the biases and limitations introduced by NLP-based methods [44, 76]. To mitigate the challenge of bias, we have involved researchers in the topic modelling evaluation and validation within our approach.

4.2 Methodology

We applied a series of NLP techniques, including the use of LLMs, to automatically identify and categorise the rationales presented in K-12 computing education literature published over the past decade. Our method involved (1) collecting relevant literature, (2) using Retrieval-Augmented Generation (RAG) to extract responses from the texts regarding the provided rationales, and then (3) fitting these responses to a topic model to cluster them into distinct rationale categories. To streamline this process, we developed a custom tool, *LitRevAI*, which is publicly available on GitHub[2].

Figure 3 shows an overview of this process. The following sections explain each step of the process in detail.

4.2.1 Data Collection & Prepossessing. We collected all literature from the ACM Digital Library[3], spanning January 2014 to June 2024, from various computing education conference and journal venues, namely:

- Innovation and Technology in Computer Science Education (ITiCSE)
- ACM Technical Symposium on Computer Science Education (SIGCSE TS)
- Koli Calling International Conference on Computing Education Research (Koli Calling)
- ACM Transactions on Computing Education (TOCE)
- Workshop in Primary and Secondary Computing Education (WiPSCE)
- ACM Conference on Global Computing Education (CompEd)
- Conference for Research on Equitable and Sustained Participation in Engineering, Computing, and Technology (RESPECT)
- ACM Conference on International Computing Education Research (ICER)
- Conference on United Kingdom & Ireland Computing Education Research (UKICER)

[2]https://github.com/soespa/litrevai
[3]https://dl.acm.org/

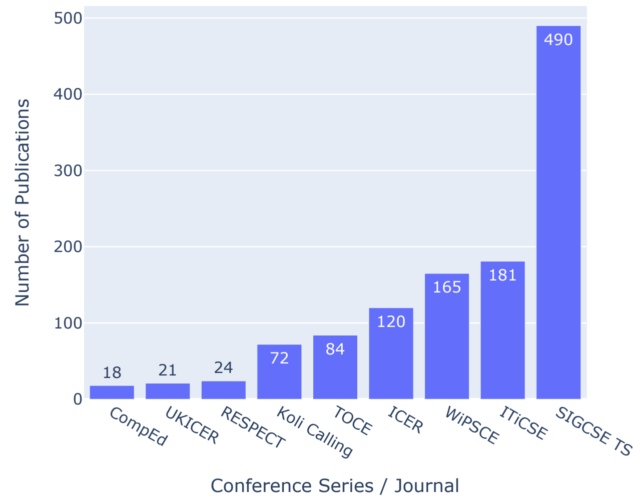

Figure 4: Number of publications by conference series / journal used to train the topic model.

Our collection includes all types of publications found within these proceedings and issues, such as full papers, short papers, posters, panels, and keynotes. For the purposes of this work, the term publications is used to refer to all of them.

The pre-processing phase involved extracting raw texts from the PDF files along with relevant metadata, followed by filtering the documents to ensure they aligned with our research scope.

To focus specifically on K-12 computing education, we included a paper in our analysis only if at least one of the following keywords appeared in the title, abstract, or keywords: *K-12, primary, secondary, school, middle school, high school* (case-insensitive).

Figure 4 shows the number of publications from each conference series and journal that were used for topic modelling. In total, we examined 4,848 papers on K-12 computing education, of which 1,172 met our criteria and were used to train the topic model. For some of the newer conferences and journals, such as CompEd, UKICER, and RESPECT, there were only a small number of publications within our scope (each with $n < 25$). This should be taken into consideration when interpreting the results, particularly those discussed in Section 4.3.4.

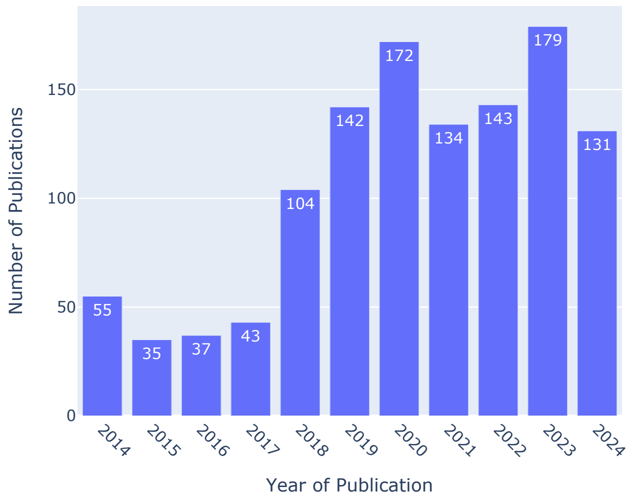

Figure 5: Number of publications by year of publication used to train the topic model.

Figure 5 shows the number of publications by the year they were published. As expected, the number of publications increases with time, reflecting both a trend among conferences to accept more papers each year and the fact that some of the conferences and journals we included have not been around for all 10 years.

4.2.2 Retrieval Augmented Generation. We applied RAG [66] to extract rationales from the raw texts. RAG operates through a two-step process: First, it retrieves relevant context from the text, and then it generates a response to the question ("Based on the article, what is the rationale or justification for teaching computing / computer science / informatics in schools?") based on the retrieved context.

For context retrieval, the document was split into overlapping chunks, each up to 1,024 characters long with a 256-character overlap between consecutive chunks. Each chunk was then embedded in a vector space and stored in a vector database. Using semantic search, the 10 most relevant chunks were selected to address the question. These selected chunks were provided as context to an LLM to generate the final answer to the question. We chose the Meta Llama 3.1 8B Instruct model [3], which we ran on a local computer to avoid uploading papers that were not publicly available on third party servers.

The LLM was instructed to generate a maximum of three bullet points in response to each query, as seen in Figure 6. With each bullet point providing one rationale, this allows for up to three rationales per paper.

We experimented with different prompts using different questions and instructions. We tested the prompts on a set of papers that were used to identify the rationales outlined in section 2.1. We compared the responses with our expectations in order to refine the prompt. We discuss some of the limitations we identified from this process in section 4.3.1.

The responses from the RAGs yielded a total of 3,576 bullet points, each of which was considered a rationale or justification

```
You are an assistant for question-answering tasks.
You will be provided with some retrieved context from a
     research article to answer a question.
If the context doesn't provide enough information for you to
      answer the question, say that you don't know the
     answer.
Give your answer in form of a unnumbered markdown list with
     a maximum of 3 short bullet points. Keep the answer
     concise.
Base your answer solely on the article's perspective and do
     not add any additional information that does not
     directly answer the question.

Question: Based on the article, what is the rationale or
     justification for teaching computing / computer science
      / informatics in schools?

Context: <context>
```

Figure 6: Prompt used for Retrieval Augmented Generation.

for teaching computing in schools according to the LLM. While the initial prompt specified a maximum of three bullet points per response, some responses exceeded this limit with up to five bullet points. This discrepancy was only noticed after merging the topics (as described in Section 4.2.3), which led to the decision to allow more than 3 bullet points per article. In retrospect, we believe that a value of 5 would have been more appropriate.

4.2.3 Topic Modelling. In the final step, we used topic modelling to cluster the generated responses based on their semantic similarity. We used BGE-M3 [129] embedding model to create vector embeddings for the responses and BERTopic [41] to create clusters based on these embeddings.

We tried different values for the minimum cluster size before settling on 7, which gave us a manageable number of clusters ($n = 71$) while retaining enough detail to detect less commonly used rationales. For all of the other parameters we used their default values.

The 71 fine-grained clusters were semi-manually merged by three of the authors, guided by their pairwise class-based term frequency - inverse document frequency (cTF-IDF) distances. cTF-IDF is a class-based variant of TF-IDF, which is a measure of how important a word is in a document relative to a collection of documents by combining how often the word occurs in the document (TF) and how unique it is across the collection (IDF) [41].

This process reduced the number of clusters down to 11. The labels for the clusters were chosen based on the corresponding keywords and representative examples (see Table 7).

By merging the initial clusters, some details were inevitably lost, leading to the creation of broader categories for the identified rationales. While this approach helps in simplifying and organising the data, it may obscure finer distinctions between different justifications that were originally identified.

For each paper, we identified which of the rationales were mentioned at least once, i.e., at least one item from the paper was assigned to the corresponding cluster. This resulted in a Boolean matrix where each element denotes whether or not a rationale is present in a paper. This matrix was then used for calculating the relative frequencies of the rationales.

4.2.4 Validation of the RAG responses. There are several limitations associated with the use of LLMs, including potential model biases, social biases related to the input [63], hallucinations [130], challenges in interpretability of results, and difficulties in handling the subtleties of intricate academic discussions [76, 82]. To evaluate the accuracy of the responses generated by the LLM, we compared automatically generated responses with human-generated responses on a sample dataset of publications. These papers were included in a manual review and already had researchers-generated explanations for each dimension of the curricular spider web framework [118]. We asked the LLM to describe seven out of the nine different aspects of the spider web framework (see Figure 1) which elicit the K-12 computing education approaches behind the rationale that comes out of topic modelling. One of the researchers who performed the manual review graded LLM's answers based on 42 papers. The criterion applied for comparison was *closeness to answers identified by human coders*. The grading was critically done with an interpretive approach. It was noted that some of the articles do not specifically answer a question. In such cases, we expect to have an LLM text response along the lines of "The article does *not* explicitly mention..." for a perfect match. However, if the LLM results in a response to such a question, we treat it as an unmatched answer. We adapted the evaluation method from Barany et al. [6]'s work in benchmarking LLM's ability to develop qualitative codebooks. The grading schema to evaluate LLM's answer for each dimension included assignment as follows:

- 1 (a perfect match with the human answer)
- 0.5 (a close yet not perfect match)
- 0 (an unmatched answer)

The evaluation of 336 responses from 42 papers shows a 79.69% accuracy in the LLM responses.

4.3 Results & Discussion

4.3.1 Rationales identified through Topic Modelling. Following the process described in section 4.2.3, a total of 11 clusters were derived from the LLM responses.

The following list presents the labels for each cluster, which were derived from the keywords and examples provided by the topic model. The clusters are ordered by their relative frequency within the papers, starting with the highest frequency:

(1) Develop Computational Thinking and Problem Solving Skills
(2) Increase Equity and Access to Computing Education
(3) Foster Interest, Self-Efficacy, and Engagement in Computing
(4) Develop Programming Skills: Reading, Writing, and Debugging
(5) Broaden Participation in Computing for Underrepresented Groups
(6) Build a Foundation in CS for Future Careers and Further Education
(7) Prepare Students for a Technology-Driven, Digital World (Digital Literacy)
(8) Address the Growing Demand for CS Professionals and National Competitiveness
(9) Data Literacy and AI Education
(10) Integrate CS with Other Disciplines
(11) Understand and Evaluate the Impact of Technology on Society

Table 7 provides detailed information about each cluster, including its label, the number and percentage of articles containing at least one item from that cluster, as well as representative keywords and examples.

Although our prompt explicitly requested rationales or justifications for teaching computing in schools (see Figure 6), some of the clusters can be more accurately characterised as learning objectives or educational goals rather than rationales. These clusters highlight specific topics or skills that students are expected to acquire, such as programming or data science. While these do not strictly align with the definition of rationales we sought (see Section 1.3), they are all closely linked to at least one underlying rationale. For example, *(6) Build a Foundation in CS for Future Careers and Further Education* is closely related to the Educational and Workforce Rationale, while *(9) Data Literacy and AI Education* is closely related to the Scientific Rationale, although it can also be attributed to other rationales such as the National Rationale, given the growing relevance of this area.

While these clusters are not rationales in the strict sense, they do provide a valuable insight into the values and beliefs of the authors by reflecting what they consider important for students to learn. For example, an emphasis on certain skills, such as programming, suggests that the authors may consider these skills to be so critical or compelling that they serve to justify the inclusion of the subject itself. In section 1.3.2 we discussed how the distinction between rationales and goals is sometimes blurred.

Other clusters, such as *(2) Increase Equity and Access to Computing Education*, *(5) Broaden Participation in Computing for Underrepresented Groups*, and *(3) Foster Interest, Self-Efficacy, and Engagement in Computing*, can be characterised as motivators (see [77]) for teaching computing rather than direct answers to the question of *why* it is important to learn about it. However, they might hint at a hidden rationale. For example, broadening participation contributes to addressing the demand for CS professionals increasing the human resources that could contribute to the CS workforce. Additionally, involving under-represented groups through educational approaches created with this motivation enriches the discipline with new ideas and perspectives.

There are a number of factors that may have contributed to this outcome. It is possible that the LLM was unable to fully capture the complexity of the authors' chain of reasoning. We used LLama 3.1 8B Instruct [3], a relatively small model, to allow for local deployment. In addition, the sentence embedding model may have been inadequate for capturing rationales, since the task on which it was trained was to distinguish between topics rather than rationales.

Another possible explanation is that in many papers the authors did not articulate their rationale in a way that was consistent with our definition (see Section 1.3). The rationales described in Section 2.1 are expected to be more abstract than the concrete arguments commonly found in the K-12 computing education literature.

Finally, the prompt used could potentially be further refined. As described in Section 4.2.2, we experimented with various prompts, incorporating different questions and instructions to mitigate this issue. The generated responses were promising and aligned well with our expectations when applied to papers where a clear rationale was explicitly stated. However, when applied to papers lacking an explicit rationale, we observed that the LLM tended to produce responses resembling learning objectives instead. This suggests that the model struggles to distinguish between rationales and learning objectives. In retrospect, it may have been beneficial to include a definition of what we define as a rationale in the prompt.

While we acknowledge that many of the clusters are not rationales in the sense we were looking for, for ease of presentation we keep using the term rationale to refer to them in the following sections.

We found that some of the rationales identified in the literature review were consistent with rationales found in similar studies. For example, the "computational thinking rationale" [15, p.60] and the "equity of participation rationale" [15, p.59] proposed by Paulo and Hejazi [91] were seen as the first two rationales to emerge from the topic modelling, namely *(1) Develop Computational Thinking and Problem-Solving skills* and *(2) Increasing Equity and Access to Computing Education*. While this represents a clear one-to-one mapping, these rationales can also be associated with the six broader rationales outlined in Section 2.1. For instance, the rationales develop *(1) Computational Thinking and Problem Solving Skill*, *(4) Develop Programming Skills*, and *(6) Build a Foundation in CS for Future Careers and Further Education* justify skill development in K-12 age groups to foster personal educational goals as well as to become eligible for future careers. Thus, these might be rooted in the educational, personal, as well as workforce rationales. However, the degree to which each LLM-generated rationale may be aligned to a rationale identified in the literature is subjective and variable based on the research context.

The LLM-generated rationale *(10) Integrate CS with Other Disciplines* is centred around integrating foundational CS skills into Science, Technology, Engineering, and Mathematics (STEM), rather than offering them as standalone units or out-of-class experiences. The rationale is aligned with the values of utilizing the power of computing as a means to understand the phenomena in the world, as the scientific rationale [33, 45] previously discussed in the literature.

Some of the rationales, such as *(9) Data Literacy and AI Education* and *(7) Digital Literacy*, are missing from the previous list of rationales described in section 2.1. Digital literacy rationale is about providing students with the skills and knowledge necessary to succeed in an increasingly technology-driven world. The *(9) Data Literacy and AI Education* rationale justifies developing AI literacy through an integration of AI concepts and the adoption of AI in future jobs. It also encompasses learning about the ethical and societal implications of AI, which is relevant to the future generation of humankind. These rationales are evident in recent years with digital technologies based on data and AI increasingly becoming ubiquitous and tightly associated with society. In this aspect, these rationales, in addition to *(11) Understand and Evaluate the Impact of Technology on Society*, seem to be connected to the societal rationale of being able to influence the technological world and being critical about the technological influence on society. Through these rationales, enabling students to recognize and evaluate the ubiquitous impact of computing technology on society is found to be along the lines of computational literacy [45] in the context of Big Data, AI and other modern technologies.

4.3.2 Frequencies of rationales across all publications. Figure 7 shows the overall usage of the 11 rationales identified by topic modelling across all publications as relative frequencies. The most frequently used rationale is *(1) Develop Computational Thinking and Problem-Solving Skills* which was mentioned in 25.6 % of all the publications reviewed, followed by *(2) Increase Equity and Access to Computing Education* (21.42 %), *(3) Foster Interest, Self-Efficacy, and Engagement in Computing* (21.16 %), and *(4) Develop Programming Skills: Reading, Writing, and Debugging* (19.03 %). Considering the relevance of the term "computational thinking"(CT) in recent years and its role in justifying the integration of the subject into national curricula in many countries, including Germany, USA, India and UK, this is not surprising.

We advise against putting too much emphasis on the percentage values, as they are partially dependent on model parameters and human influence during the merging process. However, the general trends were consistently observed regardless of the different model parameters.

4.3.3 Relevance of rationales over time. By grouping the publications by their year of publication and then calculating the frequencies of each rationale within each year, we can observe how the relevance of the rationales has shifted over time. Figure 8 illustrates these changes in rationale frequencies over the years. We also computed Spearman correlations [47, 105] between the frequencies and publication years to determine if there are statistically significant trends.

The rationale *(2) Increase Equity and Access to Computing Education* had relatively low prominence in the early years (2014–2016) but has gained significant traction and now drives much of the current computing education research ($r = .83$, $p = .002$). Similarly, we observed a sharp increase in relevance for *(9) Data Literacy and AI Education* ($r = .93$, $p < .001$), reflecting the growing importance of technology in this area. These upward trends emphasize the influence of the social impact of technology on computing education research in recent years.

In contrast, the most frequently occurring rationale, *(1) Develop Computational Thinking and Problem-Solving Skills* (see Figure 7), has shown a steady decline since approximately 2017–2019 ($r = -.74$, $p = .009$). Likewise, *(8) Address the Growing Demand for CS Professionals* and *(6) Build a Foundation in CS for Future Careers and Further Education* peaked in relevance around 2016 but have since been in decline. The relevance of other rationales has remained relatively stable over time.

4.3.4 Relevance across series and journals. Figure 9 presents the usage of each rationale across the various conference series and journals included in the corpus. The values represent the percentage of publications from each conference series or journal that includes the corresponding rationale.

Table 7: Overview of clusters ("rationales") identified through Topic Modelling. Each cluster includes the number (*n*) and percentage (%) of articles associated with the rationale, keywords ranked by cTF-IDF scores, and a representative example from the RAG responses. Note: The percentages exceed 100% as articles may state more than one rationale.

	Rationale	*n*	%	Keywords	Example
1	Develop Computational Thinking and Problem Solving Skills	300	25.6	CT, computational thinking, thinking CT, computational thinking CT, develop computational thinking, develop computational, solving skills, problem solving skills, skill, CT skills	To develop computational thinking skills.
2	Increase Equity and Access to Computing Education	251	21.4	access, equity, equitable, disabilities, CS education, participation, accessible, promote equity, ensure	To increase equity in access to and participation in computer science education.
3	Foster Interest, Self-Efficacy, and Engagement in Computing	248	21.2	interest, motivation, engagement, self, self efficacy, efficacy, interest computer science, interest computer, attitudes, increase student	To increase students' self-efficacy and interest in computing, as well as general school engagement and motivation.
4	Develop Programming Skills: Reading, Writing, and Debugging	223	19.0	debugging, children, primary, programs, program, scratch, primary school, programming concepts, code, artifacts	To develop program comprehension in novice programmers, allowing them to read and reason about code while writing it.
5	Broaden Participation in Computing for Underrepresented Groups (e.g. Girls)	184	15.7	girls, participation, underrepresented, broaden, broaden participation, women, diversity, groups, gender, participation computing	To broaden participation in computing, particularly for underrepresented groups.
6	Build a Foundation in CS for Future Careers and Further Education	206	17.6	foundation, careers, foundation computer science, foundation computer, skills knowledge, solid, quantum, future careers, solid foundation, professional	To help students build a strong foundation in computer science and programming, which can benefit their future careers and personal lives.
7	Prepare Students for a Technology-Driven, Digital World (Digital Literacy)	156	13.3	digital, increasingly, skills knowledge, century, 21st, 21st century, students skills knowledge, technology driven, students skills, succeed	To provide students with the skills and knowledge necessary to succeed in an increasingly technology-driven world.
8	Address the Growing Demand for CS Professionals / National Competitiveness	145	12.4	demand, jobs, growing, becoming, increasingly, demand computer, professionals, high demand, countries	To address the growing demand for computer science professionals and to prepare students for careers in the field.
9	Data Literacy and AI Education	92	7.8	AI, machine, machine learning, data science, intelligence, artificial intelligence, artificial, literacy, ml, technologies	To develop AI literacy through an integration of AI concepts, ethical and societal implications of AI, and the adoption of AI in future jobs.
10	Integrate CS with Other Disciplines	88	7.5	mathematics, integrate, interdisciplinary, math, physics, disciplines, subjects, integrate computing, conceptual, natural	To integrate foundational CS skills, such as designing an algorithm, generating abstractions, etc., into core academic Science, Technology, Engineering, and Mathematics (STEM) classes, rather than offering them as stand-alone units or out-of-class experiences.
11	Understand and Evaluate the Impact of Technology on Society	35	3.0	impact, technology society, impacts, evaluate, society internationally, proclaimed goal, proclaimed, internationally proclaimed, evaluate impacts, evaluate impacts computing	To enable students to recognize and evaluate the ubiquitous impact of computing technology on society, which is an internationally proclaimed goal of a K-12 computing education.

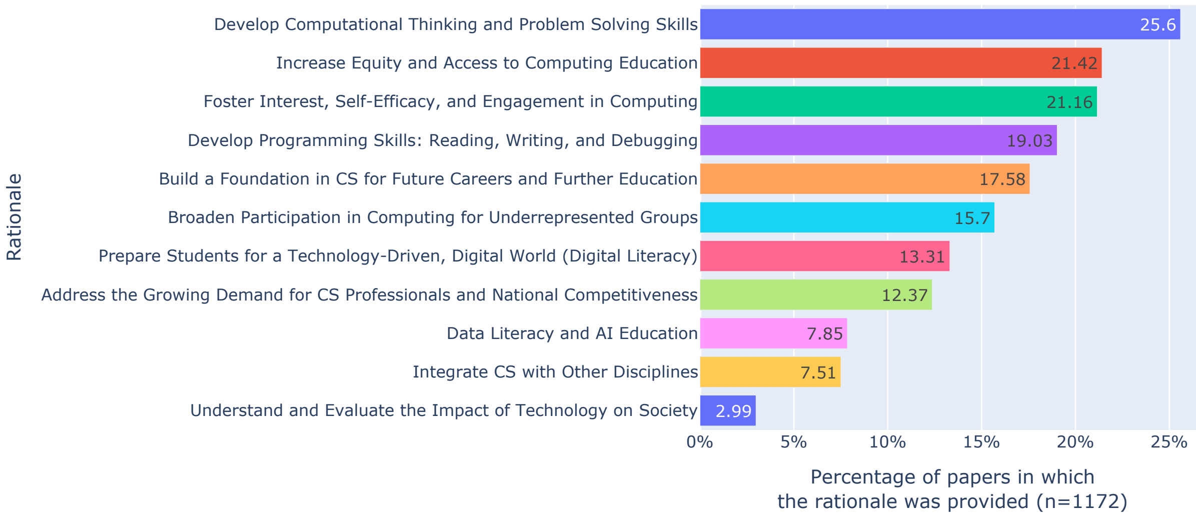

Figure 7: Relative frequencies of the retrieved rationales among all K-12 publications

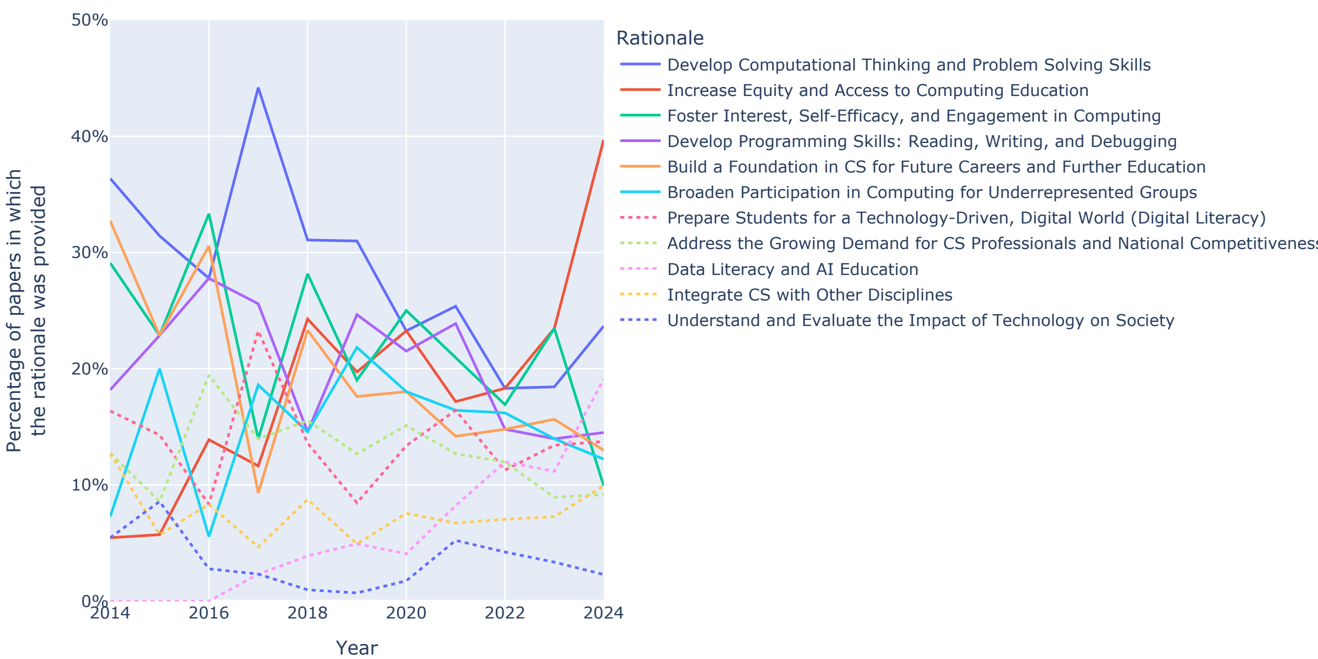

Figure 8: Change in the relevance of stated rationales over time. The percentages refer to the proportion of papers that provide the rationale in relation to all papers in the year that were included in our analysis.

The heat map highlights the focus of different conferences and journals. One notable case is the RESPECT conference, which focuses on diversity, equity, inclusion and justice in K-12 computing and computing education. As expected, the rationale *(2) Increase Equity and Access to Computing Education* stands out, with 70.8% of all the publications from RESPECT stating this rationale. RESPECT's emphasis on equity, thought-provoking reflection to inform and enrich collective scholarship was reflected in some of these LLM-generated societal rationales.

We observed similar findings for CompEd. The CompEd conference was conceived as an opportunity for SIGCSE to reach new audiences around the world, and has only two years of limited publications. The rationales *(1) Develop Computational Thinking and Problem Solving Skills* and *(6) Build a Foundation in CS for Future Careers and Further Education* were found to be the most frequently

Figure 9: Usage of rationales by conference series and journals. The values represent the percentage of publications from each conference series/journal that includes the corresponding rationale.

stated in CompEd, which is organised in venues other than those of the Global North.

Both ICER and WiPSCE have high frequencies for the *(1) Develop Computational Thinking* and *(4) Develop Programming Skills* rationales, indicating a focus on programming competencies.

On contrary, UKICER is primarily concerned about *(3) Foster Interest, Self-Efficacy, and Engagement in Computing* (28.6 %). However, it should be noted that UKICER is a new, small, single-track conference with only four years of output, at this point. On average, only around 8 full or working papers are accepted per year, and around the same number of posters. Therefore, caution is urged regarding the generalization of this pattern in relation to these new conferences.

4.4 Summary

In this section, we have described an NLP-based literature review with the intention of answering the second research question: What are the rationales that are being used to justify the teaching computing at the K-12 level? (through the last 10 years of K-12 computing education research publications). We introduced a research method informed by RAG and topic modelling techniques to analyse the large data corpus and identified the LLM-generated rationales.

Through the NLP-based review, we could confirm some of the rationales from the literature mentioned in Section 2.1. Within the timespan of the last 10 years, we could observe some of the rationales concerned with developing CT or programming skills to be on the decline in the publication focus, giving growing value to the concerns of equity, access, and reflection on the social impact of technology. We have looked at how the rationales were represented across the computing education publication venues to uncover the values and beliefs observable in the respective publications. However, we recognize that 10 years is a short timespan to justify the possible decline or growth in research values which may be, by nature, layered deeper within the longstanding traditions of K-12 computing education. We discuss these rationales further in Section 5 while answering RQ3.

We observed that the LLM-generated answers to rationales exhibit ambiguity due to the vague presentation of the research, particularly regarding what is proposed as a rationale versus what is implemented as an approach within the publication. We discuss the probable reasons for this ambiguity in the next section, where we connect the rationales observable in publications to the underpinning values and beliefs, hidden in computing education research traditions.

5 Values and beliefs underpinning K-12 computing education (RQ3): towards educational traditions

In this section, we synthesise the results obtained from both manual analysis and NLP-based analysis. Each method has its own advantages and limitations. By aligning their outcomes, we aim to abstract and frame the findings within Biesta's general educational functions [9]. Ultimately, this process led to the proposal of four educational traditions for computing education and computing education research, in order to answer our third research question.

RQ3 *What are the values and beliefs that underlie the rationales for teaching computing at the K-12 level? (hidden or implicit)*

The manual literature analysis identified six rationales by examining 58 selected papers. These papers were chosen based on theoretical considerations to represent a wide range of perspectives, values, and beliefs. An advantage of the manual approach is the depth of interpretation it provides, particularly for uncovering implicit rationales.

In contrast, the NLP analysis included a broader dataset of 1,175 papers from nine different Computing Education Conferences (from the ACM Digital Library), focusing on those related to K-12 education. This approach offers a higher likelihood of identifying all potential rationales due to its extensive scope. However, it faces challenges in accurately extracting rationales because they are often implicit. Additionally, current NLP techniques struggle to distinguish between rationales, goals, and aims of computing education, as well as to separate these from the general motivations or research rationales of the papers themselves.

Aligning and combining these two approaches allows us to build a better understanding and provides a basis for developing a general and abstract framework of *educational traditions* in K-12 computing education. In summary, we have undertaken the following three steps in this phase of the research:

(1) Synthesising the results from manual and NLP analyses using Biesta's framework of educational functions.
(2) Conducting an additional manual analysis of the six initially identified approaches, interpreting them through the lens of Biesta's three functions of education.
(3) Deriving four traditions in computing education based on the previous two steps, complemented by our background knowledge of diverse perspectives and traditions within the academic discipline of computing.

5.1 Synthesizing Manual and NLP Analyses

To synthesise and abstract the results, we framed them within Biesta's three functions of education. We begin by introducing these functions, followed by the results of the synthesis and their interpretation.

5.1.1 Educational Functions and Values. As defined in Section 1.3, we understand values as inherently social constructs describing something that is perceived by an individual or group as intrinsically desirable. The values underlying various rationales and approaches for K-12 computing education can be extremely diverse such that an exhaustive inventory or even just a list of most commonly held values appears infeasible. To structure the field, we draw on the three general functions of education suggested by Biesta [9]: qualification, subjectification and socialisation. While they do not constitute values in themselves, they do, as we argue below, readily suggest certain value clusters and may thus serve as a pragmatic framework for analysis. It is also worthy of note that the three functions seem to quite neatly align with the three basic psychological needs for competence, relatedness and autonomy, established by self-determination theory as central drivers in human behavior and well-being [94]. While needs are not the same as values, *basic* needs arguably satisfy our definition as things that are primarily pursued for their own sake.

Qualification describes the process of imparting students with certain knowledge, skills or dispositions that allow them to "do something" [9, p.40], whether that is pursuing a specific job or profession, or generally coping with everyday challenges like reading street signs, doing taxes, or following a political debate. While this suggests a myriad of possible underlying values, e.g., personal and economic prosperity, or self-efficacy, according to Biesta [9], debates about the qualification function of education have been dominated by the "common sense" value of "academic achievement", as evidenced and proliferated by large-scale assessments like TIMMS, PIRLS and PISA.

Socialisation describes the process of exposing students to and the aim to have them internalise "particular social, cultural and political 'orders'" [9, p.40]. It is similar to what with H. W. Heymann can be called "promoting cultural competence", see [8], the goal for students to develop an identity as members of a certain society or culture (see also Merry [78]). In principle, these may be geographic, ethnic, political, spiritual, professional, or based on any other shared value system around which people may form a community. Biesta [9] argues that such processes may be pursued actively, through the deliberate "transmission of particular norms and values", but are also often part of the "hidden curriculum"[9, p.40]. While the former case presupposes that the imparted norms and values were judged as desirable, in the latter case, this may not be so. Hence, biases, prejudices or inequities may be propagated unwittingly as parts of the existing social order.

Subjectification can be described as "the opposite of the socialisation function" [9, p.40], in that it does not aim to integrate students into an existing order, but to develop them as "subjects of initiative and responsibility" [11, p.77], who are able to think and act for themselves and in accordance with their very own values and interests. It emphasises development as an individual and incorporates notions like empowerment, independence, or agency.

These functions are, of course, neither mutually exclusive nor disjoint. For instance, it seems self-evident that individual agency and social participation should require certain knowledge and skill. Biesta [9] himself states that qualification can also take the form of "an introduction to modern culture"[9, p.40], thus blending it with socialisation. Moreover, subjectification may be difficult to distinguish from socialisation if one views autonomy and independence as primarily individualist and Western social values. Indeed, if values are, as we defined them above (Section 1.3), inherently social constructs, any transmission of values, any educational action aiming to bring about something desirable in another person, can technically be seen as a form of socialisation. These points notwithstanding, we believe these three functions can provide a meaningful analytical lens on the review results reported above.

5.1.2 Results of the Synthesis. To frame this study, we identified six rationales: personal, career, national, educational, societal, and scientific (Section 2.1). We observed their complex, overlapping implementation in our manual literature review (Section 3). Personal rationales were the most frequently identified (67%), followed by educational (52%) (Table 2), with most of the 58 reviewed papers incorporating multiple rationales. Some correlations between rationales and approaches were evident: the personal rationale appeared across all curriculum approaches, while the workforce rationale was absent in papers emphasising critical consciousness or computing for social good. Conversely, the societal rationale was prominent in these approaches (Figure 3).

The NLP-based literature review identified eleven rationales (Section 4). Some of these, such as "CS for future careers and further education", aligned with the original six rationales, particularly career and educational (Table 8). However, other NLP-derived rationales, like "Develop programming and debugging skills", were less straightforward, reflecting curriculum implementation rather than explicit rationales. This ambiguity highlights the difficulty of inferring rationales when they are not explicitly stated, as authors may have varying implicit motivations shaped by cultural or temporal contexts. For instance, programming might reflect a scientific rationale in the 1990s but signify societal or personal empowerment in the 2020s, or serve as an economic rationale at any time.

In summary, at an abstract level, some consensus exists in the literature on rationales for K-12 computing education (Section 2.1). However, this consensus diminishes when examining the literature on specific approaches to designing, implementing, or refining K-12 computing education programs (Section 4.3). Table 8 summarises the synthesis.

5.1.3 Interpretation of synthesis. The career, national, educational and scientific rationales were mapped on to the qualification function as they all revolve around the skills and competencies necessary for certain professional or academic pursuits, whether a well-paid or highly sought-after job or scientific and educational achievement. As already mentioned, Biesta [9] directly discusses “academic achievement” as one of the focal points of the qualification debate in recent years. The societal rationale, which revolves around responsible and informed citizenship, was mapped onto the socialisation function as such citizenship arguably constitutes participation in an existing social and political order. Finally, the personal rationale, which revolves around creativity and self-actualisation, was mapped onto the subjectification function as it is through such processes that an individual may express their own individual values and interests.

As shown in Table 8, several of the NLP-derived rationales also seem to align with the original six rationales. Training for future careers and further education clearly combines the career and educational rationales, while addressing a national demand for CS professionals aligns with the national rationale. The aim of integrating CS with other subjects seems to be more immediately concerned with implementation questions, i.e., *how* to teach computing. However, it also implies *that* computing can be beneficially integrated there, presumably to support either the learning or the practice of these disciplines, which respectively align with the educational and scientific rationale. The aim of fostering interest and self-efficacy in computing arguably includes elements of the personal rationale. One might argue that to actually count as a rationale, the objective should be to foster that *through* computing, but it is difficult to gauge how much weight to put on a single preposition when interpreting the NLP results. Preparing students for a technology-driven world clearly aligns with the societal rationale and potentially also with the personal one. Finally, the aim of allowing students to assess the social impacts of technology is also quite clearly aligned with the societal rationale.

However, several of the NLP-derived rationales are also seemingly unrelated to those used in the manual literature review. We would argue that those do not actually constitute genuine rationales for learning computing, but focus on other areas of the curricular spider web. The aims to foster CT, problem-solving, programming or debugging skills seem to align with the so-called ‘computational thinking rationale’, which we already discussed in Section 2.1 as primarily focussed on learning objectives and content, whose actual value presumably relies on some other rationale. The same argument can be made for AI education. Likewise, the aim of broadening participation in and increasing access to K-12 computing education aligns with the ‘equity of participation rationale’ also already discussed in Section 2.1 as essentially dependent on other rationales.

Overall, some of the NLP-derived rationales – including three of the four most frequently stated ones: (1) Develop CT and (2) programming skills, and (3) increase equity and access to computing education (see Table 7) – do not align with the rationales outlined in Section 2.1. We would argue that these do not constitute genuine rationales for *why* people should learn computing in the first place. However, given their relative frequency, they clearly play a prominent role in what we talk about when we talk about K-12 computing education and thus cannot be simply discarded as artefacts or model hallucinations. Instead, we propose that these rather constitute rationales for K-12 computing education *research*. For instance, fostering CT and programming skills are long-established objectives of K-12 computing education and may well be among the most prominent areas of computing education research [42, 62]. Although a more recent development, the same can currently be said about research on AI education [68, 85]. Similarly, recent years have seen a growing body of work, particularly in the US, on equity, diversity and inclusion in K-12 computing education [80, 113].

In essence, there seems to exist a notable consensus, at least in certain parts of the community, that learning CT, programming or AI basics, and fostering equitable access to related education are valuable objectives and legitimate reasons for related research – and we do not wish to contest that notion. On the one hand, where such consensus exists, repeatedly presenting an underlying rationale can quickly feel redundant or even suggest a controversy where, in fact, none exists. On the other hand, such practices may also appear, particularly to novice researchers or external onlookers, to take on a life of their own, to become a seemingly self-evident paradigm dissociated from the rationales and values that ground it. While we do not suggest that this is the case for the research areas above, we do believe that it is important for researchers to be able to distinguish these kinds of rationale and be aware of what is being justified: a research effort or a learning objective?

5.2 Secondary Manual Analysis of Approaches

Based on this mapping of rationales and educational functions, we conducted a secondary analysis of the manual review results to assess how different educational approaches correlate with different educational functions.

Table 8: Mapping of manually identified rationales (Section 2.1), NLP-derived rationales (Section 4.3) and educational functions (Section 5.1.1)

	Qualification				Socialisation	Subjectification	Implementation
(NLP) rationales	Career	National	Educational	Scientific	Societal	Personal	
Build a Foundation of CS for Future Careers and Further Ed	✓		✓				
Address the Growing Demand for CS Professionals		✓					
Integrate CS with Other Disciplines			✓	✓			✓
Foster Interest, Self-Efficacy, Engagement in Computing						✓	
Prepare Students for a Technology-Driven, Digital World					✓	✓	
Understand and Evaluate the Impact of Tech on Society					✓		
Develop CT and Problem-Solving Skills							✓
Develop Programming Skills							✓
Data Literacy and AI Education							✓
Broaden participation for under-represented groups							✓
Increase Equity and Access to Computing Education							✓

Table 9: Correlation of educational functions and approaches based on the manual review results (Section 3). This is analogous to Table 4, but aggregates rationales into educational functions in accordance with Table 8.

Approach / **Function**	Algorithmic Problem-solving	Sociotechnical	Constructionist	Computational Empowerment	Critical Computing	Computing for Social Good
Subjectification	70%	69%	64%	75%	75%	78%
Socialisation	26%	62%	50%	50%	92%	89%
Qualification	72%	85%	64%	45%	42%	33%

The result is shown in Table 9. The table is analogous to Table 4 but aggregates the six approaches into the three educational functions. That is, it shows what percentage of papers coded with a certain approach were also coded with at least one rationale mapped onto a certain educational function. For example, of the 43 papers coded with the algorithmic problem-solving approach, 31 (72%) were also coded with at least one of the four rationales mapped onto the qualification function. The subjectification function seems to be rather highly correlated with all six approaches, while the socialisation and qualification functions exhibit notably greater variance. Overall, the different approaches exhibit different correlation patterns with the educational functions.

As mentioned above, the six approaches to K-12 computing education exhibit different correlation patterns with the three educational functions (Table 9). While we would caution against giving too much weight to the statistical significance of these numbers, we do believe they suggest underlying differences with respect to how computing education and its value tend to be conceptualised in the context of different educational approaches.

The algorithmic approach exhibits a distinct pattern, while the three approaches on the right side of Table 9 show similar characteristics. These approaches score relatively high on subjectification, with the two on the far right also scoring very high on socialisation. In contrast, the algorithmic approach has the lowest scores across all dimensions. These differences likely stem from underlying beliefs about the nature of the discipline. Values and assumptions about computing appear to shape educational approaches. In Section 5.3, we investigate this hypothesis further, interpreting the results in Table 9 in more detail and linking the observed patterns to educational traditions, informed in part by prior research on disciplinary traditions.

5.3 Developing a model of educational traditions

We now present four traditions in K-12 computing education, based on our background knowledge of different perspectives and traditions within the academic discipline of computing. As mentioned above, while the focus is on K-12, it can also be useful in thinking about academic computing education.)

Traditions play an important role in shaping the rationales for why a particular subject should be taught in schools, as they are inherently tied to specific perspectives on the discipline itself. For example, a previous ITiCSE working group from 2010 examined how computing education can be defined by redefining computing itself [49]. The same was recognized by the ACM and IEEE Computer Society (IEEE-CS) in the 1980s, when the organisations decided to update their computing curricula and define a “new teaching paradigm” for computing [116]. The curriculum committee wanted the new recommendations to be tied to a new working definition of computing as a discipline, and set up two task forces, one for defining computing as a discipline, and another to develop new curriculum recommendations based on the new definition of computing as a discipline [108, p53]. The pair of documents,

Table 10: Overview of the three disciplinary traditions adapted from Tedre and Apiola [109].

	Theoretical Tradition	Engineering Tradition	Scientific Tradition
Goals & aims	Aims at coherent structures	Aims at working implementations	Aims at new findings about the world
	Concerned with coherence and correctness	Concerned with utility, reliability, and usability	Concerned with accuracy and validity
	Extending and refining theoretical knowledge	Changing the world	Understanding the world
	Algorithms and theoretical structures	Products and inventions	Discoveries
Operational principles	Conjectures	Actions	Observations
	Axioms and theorems	Processes, rules, and heuristics	Models, theories, and laws
	Analytic; deductive	Empirical; constructive	Empirical; deductive and inductive
	Concerned with structures	Concerned with processes	Concerned with causes
	Transformations between abstract ideas	Concretizations of abstract ideas	Generalizations from particular findings
	"Publish or perish"	"Demo or die"	"Publish or perish"
	Rarely makes propositions that are not proven	Must be able to act under very little information	Reluctant to make claims if there is not enough information
Fundamental assumptions	Mostly value-free	Often value-laden	Claimed to be value-free
	General and universal	Partly generalizable	Highly generalizable
	Reductionist	Holistic, can integrate competing ideas	Reductionist
	Collection of validated interconnected propositions	Propositional and procedural knowledge	Propositional knowledge
	Descriptive	Descriptive, normative, and tacit	Descriptive

"Computing as a Discipline" [25] and the ACM/IEEE-CS Computing Curricula 1991 [116], became vastly influential in shaping the field. A certain conception of computing as a discipline directly influences curricular design decisions and rationales. For instance, the *Computing as a Discipline* report [25] distinguished three different traditions of computing: theoretical, engineering and scientific. While the traditions are overlapping and each tradition contains elements of the others, some differences can be found (see Table 10).

The *educational traditions* are briefly summarised in Table 11 and in more detail outlined in the following subsections. A possible use of the traditions as framework to explicate a distinctive positionality towards computing education research is then outlined in Section 6.1.

5.3.1 The algorithmic problem-solving tradition. This tradition takes its name from the algorithmic problem-solving approach described in Section 3.1, which exhibits a unique pattern of correlation in Table 9. It is centred on algorithms and computational processes that manipulate information to solve a given problem. It is influenced by a more theoretical tradition of computing as a discipline, asking questions about computability, abstraction, logic, pattern recognition and decomposition, which are key elements, for example, in many CT initiatives [43]. It draws on the notion that computational or algorithmic ways of thinking and problem-solving are genuinely different from those in other fields [26, 29, 58, 59] and likely represents a major strand of research in computing education research; as evidenced, for example, by the frequency of associated objectives in the NLP results (Figure 7).

Proponents of the theoretical tradition of computing as a discipline have criticised more data- and information-centred perspectives as focusing too much on what algorithms manipulate, when the focus should be on the algorithmic processes instead [58]. Taken to the extreme, this may become reminiscent of Dijkstra [28]'s infamous firewall, separating questions of formal correctness from those of social context and desirability. Hence, it is at least plausible that of the six approaches, the algorithmic problem-solving approach exhibits the lowest correlation with the socialisation function.

Instead, learning objectives within the algorithmic problem-solving tradition tend to focus on developing learners' abilities to, e.g., use basic control structures to advance in the computation of a solution, effectively apply algorithmic design patterns, or improve efficiency or other algorithmic properties [e.g., 2, 102]. An algorithm, in this perspective, is primarily seen as a solution to a problem, which takes the form of a finite set of unambiguous symbol manipulations. It may have inputs, always has outputs, and terminates in a finite length of time [58]. However, some of these conditions can be relaxed; for a detailed discussion on the nature of algorithms, refer to de Miguel and Velázquez-Iturbide [23].

5.3.2 The scientific tradition. This tradition takes its name directly from the scientific tradition of computing as a discipline (Table 10), which also aptly captures the educational notions encapsulated here. Computing can be treated both as an object of learning in its own right, and also as a means of further learning about the world. It might, however, also be treated primarily as a means to an end.

Alternatively, the tradition might be called the epistemic tradition, as it emphasises the acquisition of knowledge about the world with the help of computing and by means of, for instance, exploration, description, prediction, or explanation. As an example, Isbell et al. [49], propose that computing curricula should better emphasise such core scientific competencies and content: modelling, scales and limits, simulation, abstraction, automation, and interpretation of data. This view is closely related to the educational and scientific rationales described in Section 2.1, which emphasise personal inquiry and academic achievement and both map onto the qualification function of education. In Table 9, the sociotechnical approach exhibits the highest correlation with qualification. Referring back to Table 4, we can see that this association indeed mainly stems from the educational rationale.

Computational modelling and simulation are used to teach computational approaches using real-world data: for instance, K-12 students modelling with unstructured textual data [51]. Hüsing et al. [48] have proposed the notion of epistemic programming, which "focuses on the acquisition of new insights and the expression and representation of ideas" through the act of programming [48, p.294], often in data-centric contexts. Moreover, software artefacts themselves may be seen as executable models to experiment with: for instance, an algorithm whose runtime behaviour or outcome quality is to be measured empirically [122].

The tradition is thus well suited for learning activities like inquiry-based learning and simulation-based pedagogy in which students are supported to develop their understanding by conducting experiments, as well as observing and exploring objects and processes that are otherwise beyond perception or control in the physical world. In the context of K-12 computing education, this tradition fits naturally with the cross-curricular computing needs of other disciplines, especially the STEM subjects.

5.3.3 The design and making tradition. This tradition takes its name from the constructionist design-making approach described in Section 3.1, which exhibits another unique pattern of correlation in Table 9. It emphasises the creation of useful artefacts, tools, products, or inventions that fulfil a social need or want.

This tradition borrows from the engineering tradition of computing as a discipline, which casts it as more of a practical endeavour that not only has to account for algorithms and data in theory, but also for "available material" and "the laws of nature" [109, p.103] in practice. It therefore has overlaps with both the algorithmic and societal traditions. It is also arguable that the design and making tradition exhibits an almost balanced correlation with three educational functions: qualification, socialisation and subjectification.

In an educational context, disciplinary engineering rigour is often replaced by more playful, open-ended and creative projects. Variants of constructionism [89] as well as socially oriented theories of learning [17, 65, 124] are frequently used as pedagogical and theoretical background [52]. Students are regarded quite literally as creative builders of knowledge [52, 90]. When designing and making external artefacts whether sandcastles on the beach or computer programs, evolving ideas and thinking become visible and shareable [89]. Such externalisation, in turn, makes learning and reasoning processes visible for joint evaluation and development [1, 90, 92]. As mentioned above, physical computing and maker initiatives are representative of this tradition [14, 40, 98]. They often emphasise collaborative projects and provide learners with powerful tools and social settings that aim to cultivate imagination, creativity, communication and self-expression in increasing levels of expertise and complexity [52, 93].

5.3.4 The societal tradition. This tradition of computing education is rooted in the understanding that computing, its practices, and products are fundamentally embedded within broader social, cultural, and ethical contexts. The approaches that emphasise such embedding most prominently are arguably the critical and computing for social good approaches (Section 3.1). Both exhibit very similar correlation patterns in Table 9 with a very strong focus on socialisation yet notably less qualification.

While we are not aware of a comprehensive account of a corresponding disciplinary tradition of computing, it includes a number of key aspects and positions. Firstly, as an underlying core assumption, technology in general and computing in particular is seen not as a morally neutral tool, which only becomes *good* or *bad* in the hands of the people who use it, but as fundamentally value-laden, incorporating and propagating the goals and values of the people who produce it. Hence, teaching ethics to aspiring computing professionals has a long-standing history in tertiary education, albeit still with notable practical challenges [18]. Connolly [20] has even argued that computing is essentially a social science, calling "to move the academic discipline of computing away from engineering-inspired curricular models and supplement it with the methods, theories, and perspectives of the social sciences" [20, p.54].

Secondly, Mahoney [74] describes what he calls the "impact theory" of computing technology [74, p.121], which sees it as an important driver in social change, for better or worse. A central aspect of impact theory is that it has "people reacting to technology rather than actively shaping it" [74, p.122]. Conversely, the people who do shape it essentially hold positions of social power. Consequently, the discipline of computing may be described as a community of practice [107], fundamentally shaped by its members. Their identities, values, world views, and sociocultural backgrounds thus become important disciplinary factors.

Some of these disciplinary notions carry over into the corresponding educational tradition. Related approaches, particularly critical approaches to computing education (see Morales-Navarro and Kafai [81]), often aim to enable learners to understand and evaluate how computing technologies are shaped by the sociocultural contexts in which they are designed and how, in turn, they influence social realities in their contexts of application. Students should develop an awareness of such mechanisms, in order to discover social biases and inequities in existing technologies. Classroom discussions and reflection practices are common [e.g., 12, 54, 60, 95, 96].

Another category of approaches within the societal tradition derives from the view of computing as a community of practice and aims to diversify its members, particularly with respect to underrepresented student groups such as women or minorities [39]. A key pedagogical idea is to make the practice of computing more relevant to these groups of students, e.g., by embedding learning material in sociocultural or local contexts directly related to those of the learners [e.g., 19, 55, 80, 86].

5.3.5 Interpretation of the four educational traditions. The four educational traditions described above derive and abstract from different curricular approaches. As already mentioned, we do not understand these traditions as mutually exclusive, nor do we suggest that any one tradition is superior to another. In fact, as we will further argue in Section 7, individual educators may subscribe to a combination of them, giving each a particular weight or priority. Similarly, curricular approaches may well combine elements of different traditions. One example, which we have not yet discussed, is the computational empowerment approach. As described in Section 3.1, it departs from the core tenets of CT yet shifts them towards a more social and critical perspective [31, 50]. Its correlational pattern shown in Table 9 can be interpreted as a blend of the patterns exhibited by the algorithmic problem-solving approach on the one hand, and the two "societal" approaches on the other. In that sense, computational empowerment incorporates aspects from both traditions.

Moreover, consider the computational empowerment and sociotechnical approach in contrast to each other. Both advocate for explicitly including the societal tradition and propose similar curricular suggestions. The computational empowerment approach revolves around the central concepts of decoding and coding, while the sociotechnical approach focuses on deconstruction and later on a similar dual notion of exploring and designing. At first glance, these approaches seem almost indistinguishable. However, the computational empowerment approach, with its focus on CT, is more closely aligned with the algorithmic problem-solving tradition. In contrast, the sociotechnical approach leans towards the scientific tradition with its emphasis on epistemic inquiry and model building.

6 Discussion: Building a Framework

We have seen that in the field of K-12 computing education a broad range of rationales are being discussed that inform and shape approaches to computing education. A mix of rationales and approaches that are constantly being invented, re-invented and being re-shuffled is quite natural for scientific research in education [101]. New approaches often build on current or older approaches with the intent of remedying shortcomings in those predecessors such that progression in the field is – at least implicitly – being construed as incremental. For example, in a textbook from Germany in 2010 (English translation not available), it is suggested that the algorithmic approach is now outdated [99]. We believe that the field should be structured with broad, deep, durable and stable views or traditions of K-12 computing education, rather than isolated and individual approaches.

In this section, we propose that the four traditions can be used as an analytical framework providing different lenses to examine particular curricular approaches, their goals, and potential rationales. The framework is presented as four separate but connected traditions or views on K-12 computing education, as introduced in Section 5.3. This framework can add two insights to the debate on refining K-12 computing education approaches. Firstly, an approach is usually context-bound, and has to react to and interact with the current geographical, social and time-based context issues (school system, societal needs, state of computing as discipline, infrastructure, and so forth), as well as pervading technological developments. It therefore makes more sense to conceptualise an approach not as an endpoint of the current scientific debate, but as an instantiation of a tradition - and hopefully as actualisation and refinement of such a tradition, too.

Secondly, computing education is on the move, as well as the discipline itself. It probably makes more sense thus to conceptualise a pedagogical approach to K-12 computing education somewhat more narrowly to be focused on some subset or view or tradition of the discipline. That is, it seems to be reasonable not to see K-12 computing education as either to teach problem solving or societal aspects, but probably to do both while discussing the proportions and the relationship between different approaches within different traditions. We believe this could impact how curricula are being designed.

Future work might draw on the framework to better understand the differences and commonalties across curricular approaches, or possibly discover novel ways to combine aspects from different traditions in order to develop new curricular approaches.

6.1 Towards a 'traditions framework'

We thus tentatively propose this framework for K-12 computing education as a tool that researchers and others can use to position their work. Each of the four traditions stems from different values and beliefs that an individual might hold. In the literature review conducted to answer RQ1, we noted that papers would often be coded against more than one curriculum approach and/or rationale. This indicates that hardly any of us will be driven by just one of these traditions, but by a combination, to a greater or lesser extent.

The framework can be used to give researchers a shared language around the values and beliefs that underpin their work in K-12 computing education research.

6.1.1 The algorithmic tradition. A researcher whose values align to the algorithmic tradition might positively respond to these statements:

- K-12 Computing education should focus primarily on developing students' ability to think algorithmically.
- Understanding algorithms and computational processes is more important than understanding the data those algorithms manipulate.
- Learning to program is essential because it teaches the fundamental principles of computation.
- Computing should be taught as a discipline that emphasizes problem-solving using algorithmic techniques.
- The most important aspect of K-12 computing education is teaching students how to write efficient and effective algorithms.

6.1.2 The scientific tradition. A researcher whose values align to the scientific tradition might positively respond to these statements:

- Using computational models and simulations is essential for understanding complex systems and phenomena.

Table 11: Four traditions in K-12 computing education.

Tradition / Aspect	Algorithmic Problem-Solving	Scientific	Design and Making	Societal
View of computing as a discipline	A formal practice that focuses on algorithms, computation, and transforming information.	A method of scientific inquiry aimed at understanding the world.	An engineering science emphasizing the design and construction of artefacts within constraints.	A social practice shaped by its community, producing tools that embody and influence societal values.
Why it is important	Highlights the unique ways of thinking and problem-solving inherent to the field.	Emphasizes learning about the natural and artificial world through computational approaches.	Stresses the significance of implementing computational solutions within real-world artefacts.	Demonstrates the need to reflect on the social context and moral implications of computing.
How it shapes computing education	Encourages teaching abstract concepts and context-independent thinking.	Promotes modelling, simulation, and computational tools for inquiry.	Anchors projects in learners' creativity, interests, and collaboration to solve complex problems.	Inspires learners to examine the interplay of computing and society while fostering a sense of responsibility.

- The primary goal of K-12 computing education should be to enable students to use computational tools for scientific inquiry.
- K-12 Computing education should integrate scientific methods such as hypothesis testing, data collection, and analysis.
- Computing should be seen as a means to understand and explore the world, both natural and artificial.
- Learning computing is valuable because it teaches how to think critically and analytically like a scientist.

6.1.3 The design-making tradition. A researcher whose values align to the design-making tradition might positively respond to these statements:

- The process of making and sharing digital artefacts is more valuable than the final product itself.
- Students learn best when they are actively engaged in designing and creating digital or physical artefacts.
- Learners should be empowered to see themselves as designers and innovators in computing.
- Building personally meaningful projects is the best way to learn computational concepts.
- K-12 Computing education should focus on creativity, collaboration, and real-world problem-solving.

6.1.4 The societal tradition. A researcher whose values align to the societal tradition might positively respond to these statements:

- It is important for students to understand that computing is not a neutral discipline but is value-laden.
- K-12 Computing education should emphasise the social and ethical implications of technology.
- K-12 Computing education should foster a sense of responsibility towards individuals, societies, and humanity.
- Students must learn how computing systems are shaped by societal values and power dynamics.
- A critical aspect of learning computing is understanding its impact on equity, justice, and social good.

How the four traditions might be implemented in the curriculum is shown in Table 12.

6.2 How a traditions framework might be used

Developing a framework for values and beliefs underpinning K-12 computing education research provides a common vocabulary and starting point to support the K-12 computing education community to review teaching activities, teacher training work, curricula, research studies, and computer science education policy. It can establish the philosophy that we—as K-12 computing education researchers—do not necessarily have to have the same values and beliefs underpinning our research. Framing different research endeavours more explicitly in relation to their primary educational tradition may be beneficial for future discourse in the K-12 computing education research community.

To illustrate the framework, we present five use cases for K-12 computing education stakeholders, namely for teachers, teacher trainers, resource developers, computing and associated curriculum designers, researchers and policymakers.

Researchers - to reflect on personal philosophy and positionality, develop a research study statement and to support peer review. Researchers can use the framework to reflect upon their personal philosophy and the context of the studies they are developing. Studies may also be constrained or influenced by funder requirements, the cultural context, political trends, and the institutional backdrop. An aim could be for the research team to collaboratively develop a statement describing the traditions and rationale for their study. Such a statement could be a useful component in funding bids, and a summary could be included in academic output. Such a summary could become a standardised item in the aim of the study description, and could be requested by conference and journal organisers. Such a narrative would help reviewers to better understand the motivation for the research as well as decisions made related to method, and standpoints in the discussion.

Table 12: Suggestions for the curriculum implementation for the traditions in the framework (drawn from Section 5.2)

Tradition **Curriculum Question**	**Algorithmic and problem solving**	**Scientific**	**Design and Making**	**Societal**
Towards which goals are they learning?	develop learners' cognitive abilities	core scientific competencies and content	design and make	to discover social biases and inequities in existing technologies
What are they learning?	computability, abstraction, logic, pattern recognition and decomposition	modelling, scales and limits, simulation, abstraction, automation, and interpretation of data	designing and making	understand how computing technologies are shaped by the social context in which they are designed
How are they learning?	activities to use algorithmic design techniques to constructively compute the elements of a solution	inquiry-based learning and simulation-based pedagogy	design and create digitally or materially embodied artefacts and projects that are often shared with others	classroom discussion and reflection
How is the teacher facilitating learning?	lessons may be teacher-led with activities designed by the teacher	teacher designs activities which stimulate curiosity, inquiry and use of tools	with powerful tools and social settings that cultivate imagination, creativity and play at the same time	facilitates discussion and reflection

The statement of tradition and rationale would include elements as exemplified in Section 6.1. It may be that researchers blend more than one of the suggested traditions (see Figure 10 in Section 7).

During peer review processes, there is a recognised issue that reviewers or publication groups may penalise studies that have different theoretical viewpoints to the reviewers [84]. The framework may also help researchers when they act as peer-reviewers. Reviewers could point to the framework as a way to help researchers describe their tradition and rationale. A characterisation would help reviewers become aware of any potential clashes between their personal philosophy and that of academic work they are reviewing.

Educators – to review and plan teaching and learning. In order to select consistent foundations for their teaching, teachers need deep insight into the different areas of CS to decide which existing learning materials meet their expectations and which do not [99]. Teachers and other educators could use the overview of the K-12 computing education traditions and rationale to learn about and reflect upon their practice. Schools may choose to develop a statement about their computing tradition within their computing teaching and learning policy, for sharing with parents, students, school inspection and other interested groups.

To reflect on how others have put rationales into practice, teachers or other educators could reflect on the results from the manual scoping literature review of the most common coded answers to van den Akker's curriculum planning questions [117] (see Section 3.3 and Table 5). These exemplars could be used to review, challenge and inspire educators to change their curriculum approach.

Teacher educators – to review and plan pre-service and in-service education. Teacher educators can individually consider the framework to reflect upon their positionality. Then, within their education establishments they might review whether the teacher professional development curricula covers all the traditions and rationale and how they are presented and in what order. Some education establishments may emphasise certain traditions or rationales due to their historical or cultural contexts, with the important factor being that this framework enables reflection.

Curriculum designers – to design curricula. Curriculum designers may have their own perspectives on K-12 computing education and these are likely to influence what content is included and how it might be taught [49]. The underpinning views and theory available to designers are perhaps not agreed or stated [52, 84]. Therefore, curriculum designers and resource developers can use the framework to reflect upon their personal philosophy and also, as with researchers and other groups, to develop a shared understanding of what traditions will underpin the design of curricula and associated resources. What traditions are selected will, as with other groups, be constrained or informed by the sociocultural context, political and educational factors.

Policymakers – to review and create computing policy. Education is an essential element of government policy. Yet, it is likely policymakers will be unfamiliar with CS education. Policymakers will need to form their values, beliefs, and understanding of the discipline. Therefore, having the description of the traditions (Section 6.1) will help policymakers to better critically review CS education research and be more equipped to reflect on K-12 computing education in general. Hopefully, this will lead them to make more informed decisions on how CS can be included in their education policy. Helping policymakers better understand the CS education landscape is particularly important now, as in many countries, there may be no or limited CS provision [120], a lack of trained CS teachers [120], or unclear guidance for school leaders as to whether they should prioritise CS over other subjects [34].

6.3 Limitations

When organising a large group of researchers on a working group study, it is important to clearly define core terminology and the methodology to ensure that all researchers are on the same page and aware of the processes they should follow and how standards are maintained across researcher work. Similarly, on a multi-staged study, common core definitions must be defined and methods described to help manage the outputs and inputs across the boundaries of stages.

Core terminology was defined early, agreed upon in working group meetings, and documented for referral in shared documents and included in this study paper (Section 1.3). Similarly, methods were agreed upon early and documented and as processes were adapted, all working group members were kept informed through shared documents and regular meetings and updates.

In this working group study, there are three phases, each of which has a clearly defined method. To evidence that the methods have been followed, audit trails have been maintained through working documents in shared folders. Also, researchers have reviewed the work of their peers on a regular basis to confirm a common shared understanding and process adherence. There are, as with all studies, specific limitations, which should be considered by readers to better understand the generalisation of findings. We highlight here major limitations and how they have been mitigated to reduce impact.

- A set of six rationales (Section 2.1) and six approaches (Section 3.1) were initially defined based on working group consensus discussion; this was required to start the manual literature review process (Section 3.2). A different working group composition could select alternative initial rationales and approaches. Still, the objective of the starter items was to discover what alternatives would be found through subsequent phases of the study.
- The number of papers manually analysed for RQ1 was limited by time and the number of researchers available (Section 3). The paper analysis was complex and lengthy, with two searches carried out at two separate stages. It required a second researcher to check each set of categorisations. To mitigate the limited number of papers included in the process, sampling was purposive for some searches to select papers representing a variety of rationales and approaches. Despite this, the analysis revealed varied and overlapping combinations with limited clear patterns. Therefore, it was judged that increasing the number of papers reviewed would have added limited new knowledge on patterns.
- During the NLP analysis of papers for RQ2 (Section 4.2) a large number of papers were included, but there are limitations related to the use of NLP in paper analysis [76, 82]. Steps were taken within the automated paper analysis to reduce risks to validity and integrity, including maintaining an audit trail, discussion and agreement between researchers on technical choices made and manual intervention stages. There was also a specific validation of the RAG responses comparing automatically generated LLM analysis of papers to manually generated analysis (Section 4.2.4) with a 79.69% accuracy found.
- For the final development of the traditions to answer RQ3 (Section 5), researchers worked as a group to review the analysis done and reflect upon conclusions drawn.
- The study's design with its three phases, and the usage of different methods and tools for each research question, makes it complex to combine results across the phases. This complexity might lead to difficulties in keeping the overall conclusions consistent and coherent. Methodological differences between stages could have affected the final synthesis of results.

7 Conclusion

K-12 computing education research is a rapidly growing and dynamic discipline, gathering increased interest as more countries bring computing into their curriculum. This paper has described a project conducted by a large ITiCSE working group to investigate how we can surface and explicate some of the values and beliefs held by computing education researchers. As part of this research, we have used both manual methods and NLP to explore and analyse underpinning rationales, and used curriculum research to investigate how rationales might be implemented in a classroom context. Importantly, we then drew on theoretical perspectives and disciplinary traditions and developed them into educational traditions for our field, resulting in four traditions of K-12 computing education: the algorithmic and problem-solving tradition, the scientific tradition, the design and making tradition, and the societal tradition. The intention is that researchers, teachers, teacher educators and policy makers can all use this work to reflect on their own positionality within this academic field; we hope that we have made some small steps towards introducing a shared language to assist in professional discourse about our own and others' research.

We believe the situation is similar to the situation in chemistry, where teachers have different curricular emphasis, but often not only one pure emphasis [119]. Over time, the number of emphases in chemistry has been significantly reduced, leading to three primary focuses: 'fundamental chemistry', 'chemistry, technology and society', and 'knowledge development in chemistry' [119]. To gain a more abstract understanding of the overarching emphases in K-12 computing education, we examined the discipline-specific traditions. Initially, we identified three core traditions: traditional, engineering, and scientific. Building on this framework, we expanded our analysis to include a fourth educational tradition, the societal tradition, which considers the impact of computing on society and its ethical implications. These four educational traditions offer a more nuanced lens for analysing the rationales and approaches in computer science education.

In summary, in this report we have analysed the debate in K-12 computing education research in terms of implicit and explicit common understandings. Explicitly, this has been in terms of stated

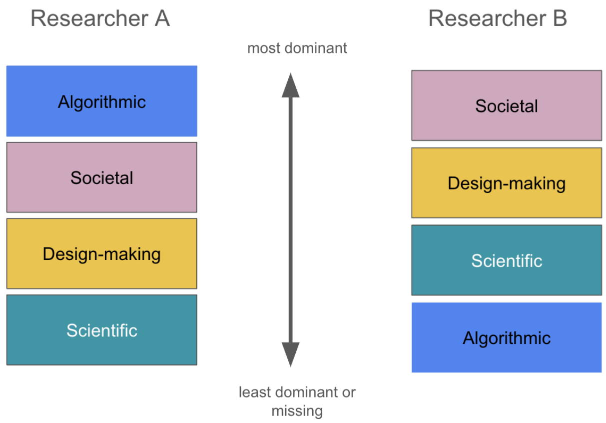

Figure 10: Researchers may differ in the traditions that underpin their perspectives on K-12 computing education

reasons for why people should learn computing in K-12, and implicitly, in terms of commonalities in underlying values and beliefs. Between the four *traditions* are what we call *approaches* to K-12 computing education. These approaches can be seen as a living market, where new approaches are being developed, existing approaches are being refined, and classic approaches still exist and probably dominate the market. New approaches are also likely to be blends of existing ideas. In particular, the prominence of particular rationales fluctuates over time and varies across different conferences and sub-communities within computer science education research. This temporal and contextual variability contributes to the lack of a clear, overarching pattern. It was therefore not really useful to make clean distinctions between approaches, for example, by attempting to assign rationales exclusively to particular approaches. There is much overlap and, of course, different advocates in the debate may have different views on the same approach.

7.1 Further Work

K-12 computing education researchers may draw on one or more of the traditions in their work, to a greater or lesser extent. One tradition may be dominant for them, and some less so, as shown in Figure 10. Their values may also change over time. However, this paper highlights the lack of a common language to support dialogues about the nature of the field. We propose that K-12 computing education researchers might welcome a tool that help them reflect on some of the statements in Section 6.1. As an example, the working group devised a sample quiz[4], at this point offered quite lightheartedly, to illustrate a potential way of considering one's influences and values relating to K-12 computing education in school. Further work would involve surveying and interviewing researchers as to their perspectives on K-12 computing education, and their views with respect to the four traditions, which may highlight whether a tool would be worth further exploration.

We have drawn on the expertise of the working group and analysed a snapshot of existing literature from the last ten years to address the difficult questions we posed ourselves; in further work we would like to broaden the discussion to include many other perspectives and a wider body of literature, which will result in a deepening of our analysis. We are aware that the 'we' in 'what we talk about when we talk about K-12 computing education' is limited to the computing education research community, and we could extend the discussion to others, including science education researchers, the maker community, general education, early childhood researchers, etc. We, and other researchers following on from this project, could also look at implemented curricula around the world and engage with practising educators, thus extending the analysis beyond published research.

[4]https://form.jotform.com/242535028788263

References

[1] Edith Ackermann. 2001. Piaget's Constructivism, Papert's Constructionism: What's the Difference. *Future of Learning Group Publication* 5, 3 (2001), 438.

[2] Alfred Vaino Aho. 2012. Computation and Computational Thinking. *Comput. J.* 55, 7 (2012), 832–835. https://doi.org/10.1093/comjnl/bxs074

[3] AI@Meta. 2024. Llama 3 model card. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[4] Annette Thijs and Jan van den Akker (Eds.). 2009. *Curriculum in development.* SLO - Netherlands Institute for curriculum development, Enschede. Backup Publisher: SLO • Netherlands Institute for Curriculum Development tex.ids: annettethijsCurriculumDevelopment2009, slonetherlandsinstituteforcurriculumdevelopmentCurriculumDevelopment2009.

[5] Claus Boye Asmussen and Charles Møller. 2019. Smart literature review: a practical topic modelling approach to exploratory literature review. *Journal of Big Data* 6, 1 (Dec. 2019), 93. https://doi.org/10.1186/s40537-019-0255-7

[6] Amanda Barany, Nidhi Nasiar, Chelsea Porter, Andres Felipe Zambrano, Alexandra L. Andres, Dara Bright, Mamta Shah, Xiner Liu, Sabrina Gao, Jiayi Zhang, Shruti Mehta, Jaeyoon Choi, Camille Giordano, and Ryan S. Baker. 2024. ChatGPT for Education Research: Exploring the Potential of Large Language Models for Qualitative Codebook Development. In *Artificial Intelligence in Education*, Andrew M. Olney, Irene-Angelica Chounta, Zitao Liu, Olga C. Santos, and Ig Ibert Bittencourt (Eds.). Springer Nature Switzerland, Cham, 134–149.

[7] Ali Battal, Gülgün Afacan Adanır, and Yasemin Gülbahar. 2021. Computer Science Unplugged: A Systematic Literature Review. *Journal of Educational Technology Systems* 50, 1 (Sept. 2021), 24–47. https://doi.org/10.1177/00472395211018801 Publisher: SAGE Publications Inc.

[8] Rolf Biehler. 2019. Allgemeinbildung, Mathematical Literacy, and Competence Orientation. In *Traditions in German-Speaking Mathematics Education Research*, Hans Niels Jahnke and Lisa Hefendehl-Hebeker (Eds.). Springer International Publishing, Cham, 141–170. https://doi.org/10.1007/978-3-030-11069-7_6

[9] Gert Biesta. 2009. Good Education in an Age of Measurement: on the Need to Reconnect with the Question of Purpose in Education. *Educational Assessment, Evaluation and Accountability* 21, 1 (Feb. 2009), 33–46. https://doi.org/10.1007/s11092-008-9064-9

[10] Gert Biesta. 2010. Why 'What Works' Still Won't Work: From Evidence-Based Education to Value-Based Education. *Studies in Philosophy and Education* 29, 5 (2010), 491–503. https://doi.org/10.1007/s11217-010-9191-x

[11] Gert Biesta. 2015. What is Education For? On Good Education, Teacher Judgement, and Educational Professionalism. *European Journal of Education* 50, 1 (2015), 75–87. https://doi.org/10.1111/ejed.12109

[12] Karl-Emil Kjær Bilstrup, Magnus H. Kaspersen, and Marianne Graves Petersen. 2020. Staging Reflections on Ethical Dilemmas in Machine Learning: A Card-Based Design Workshop for High School Students. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference* (Eindhoven, Netherlands) *(DIS '20)*. Association for Computing Machinery, New York, NY, USA, 1211–1222. https://doi.org/10.1145/3357236.3395558

[13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3, Jan (2003), 993–1022.

[14] Paulo Blikstein. 2013. Gears of our Childhood: Constructionist Toolkits, Robotics, and Physical Computing, Past and Future. In *Proceedings of the 12th International Conference on Interaction Design and Children (IDC '13)*. Association for Computing Machinery, New York, NY, USA, 173–182. https://doi.org/10.1145/2485760.2485786

[15] Paulo Blikstein and Sepi Hejazi Moghadam. 2019. Computing Education: Literature Review and Voices from the Field. In *The Cambridge Handbook of Computing Education Research*, Sally A. Fincher and Anthony V.Editors Robins (Eds.). Cambridge University Press, UK, 56–78.

[16] Francisco Bolaños, Angelo Salatino, Francesco Osborne, and Enrico Motta. 2024. Artificial Intelligence for Literature Reviews: Opportunities and Challenges. *Artificial Intelligence Review* 57, 10 (Aug. 2024), 259. https://doi.org/10.1007/s10462-024-10902-3

[17] John Seely Brown, Allan Collins, and Paul Duguid. 1989. Situated Cognition and the Culture of Learning. *Educational Researcher* 18, 1 (1989), 32. https://doi.org/10.2307/1176008 Publisher: SAGE Publications.

[18] Noelle Brown, Benjamin Xie, Ella Sarder, Casey Fiesler, and Eliane S. Wiese. 2024. Teaching Ethics in Computing: A Systematic Literature Review of ACM Computer Science Education Publications. *ACM Trans. Comput. Educ.* 24, 1 (Jan. 2024), 6:1–6:36. https://doi.org/10.1145/3634685

[19] Caelin Bryant, Yesheng Chen, Zhen Chen, Jonathan Gilmour, Shyamala Gumidyala, Beatriz Herce-Hagiwara, Annabella Koures, Seoyeon Lee, James Msekela, Anh Thu Pham, Halle Remash, Marli Remash, Nolan Schoenle, Jonah Zimmerman, Sarah Dahlby Albright, and Samuel A. Rebelsky. 2019. A Middle-School Camp Emphasizing Data Science and Computing for Social Good. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 358–364. https://doi.org/10.1145/3287324.3287510

[20] Randy Connolly. 2020. Why Computing Belongs within the Social Sciences. *Commun. ACM* 63, 8 (July 2020), 54–59. https://doi.org/10.1145/3383444

[21] National Research Council. 2010. *Report of a Workshop on The Scope and Nature of Computational Thinking.* National Academies Press, Washington, D.C. https://doi.org/10.17226/12840 Pages: 12840.

[22] Paul Curzon, Bell, Tim, Dorling, Mark, and Jane Waite. 2019. Computational Thinking. In *The Cambridge Handbook of Computing Education.* Cambridge University Press, UK. https://core.ac.uk/reader/201006880

[23] Julia de Miguel and J. Ángel Velázquez-Iturbide. 2025. What is an Algorithm?: Traditional vs. Intelligent Algorithms. In *Encyclopedia of Information Science and Technology* (6th ed.). IGI Global, Hershey, PA, 111–140. https://doi.org/10.4018/978-1-6684-8656-6.ch006

[24] Peter J. Denning. 2009. The Profession of IT: Beyond Computational Thinking. *Commun. ACM* 52, 6 (June 2009), 28–30. https://doi.org/10.1145/1516046.1516054

[25] Peter J. Denning, Douglas E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, and Paul R. Young. 1989. Computing as a Discipline. *Commun. ACM* 32, 1 (Jan. 1989), 9–23. https://doi.org/10.1145/63238.63239

[26] Peter J. Denning and Matti Tedre. 2021. Computational Thinking for Professionals. *Commun. ACM* 64, 12 (Nov 2021), 30–33. https://doi.org/10.1145/3491268

[27] Fabio Dennstädt, Johannes Zink, Paul Martin Putora, Janna Hastings, and Nikola Cihoric. 2024. Title and Abstract Screening for Literature Reviews using Large Language Models: an Exploratory Study in the Biomedical Domain. *Systematic Reviews* 13, 1 (2024), 158. Publisher: Springer.

[28] Edsger W. Dijkstra. 2008. In Reply to Comments (EWD 1058). https://www.cs.utexas.edu/~EWD/transcriptions/EWD10xx/EWD1058.html.

[29] Wil Dijkstra. 1987. Interviewing Style and Respondent Behavior: An Experimental Study of the Survey-Interview. *Sociological Methods & Research* 16, 2 (1987), 309–334.

[30] Christian Dindler, Ole Sejer Iversen, Michael E. Caspersen, and Rachel Charlotte Smith. 2022. Computational Empowerment. In *Computational Thinking Education in K–12: Artificial Intelligence Literacy and Physical Computing.* The MIT Press, New York, USA. https://doi.org/10.7551/mitpress/13375.003.0009

[31] Christian Dindler, Rachel Smith, and Ole Sejer Iversen. 2020. Computational Empowerment: Participatory Design in Education. *CoDesign* 16, 1 (Jan. 2020), 66–80. https://doi.org/10.1080/15710882.2020.1722173

[32] Daniella DiPaola, Blakeley H. Payne, and Cynthia Breazeal. 2020. Decoding Design Agendas: An Ethical Design Activity for Middle School Students. In *Proceedings of the Interaction Design and Children Conference* (London, United Kingdom) *(IDC '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3392063.3394396

[33] Andrea A. diSessa. 2018. Computational Literacy and "The Big Picture" Concerning Computers in Mathematics Education. *Mathematical Thinking and Learning* 20, 1 (Jan. 2018), 3–31. https://doi.org/10.1080/10986065.2018.1403544

[34] Cheri Fancsali, Linda Tigani, Paulina Toro Isaza, and Rachel Cole. 2018. A Landscape Study of Computer Science Education in NYC: Early Findings and Implications for Policy and Practice. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. Association for Computing Machinery, New York, NY, USA, 44–49. https://doi.org/10.1145/3159450.3159467

[35] Paulo Freire. 1974. *Pedagogy of the Oppressed.* Seabury Press, New York, USA.

[36] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. https://doi.org/10.48550/arXiv.2312.10997 arXiv:2312.10997 [cs]

[37] Erin B. Godfrey and Luke J. Rapa (Eds.). 2023. *Developing Critical Consciousness in Youth: Context and Settings* (1 ed.). Cambridge University Press, UK. https://doi.org/10.1017/9781009153843.013

[38] Michael Goldweber, John Barr, Tony Clear, Renzo Davoli, Samuel Mann, Elizabeth Patitsas, and Scott Portnoff. 2013. A Framework for Enhancing the Social Good in Computing Education: A Values Approach. *ACM Inroads* 4, 1 (March 2013), 58–79. https://doi.org/10.1145/2432596.2432616

[39] Michael Goldweber, Lisa Kaczmarczyk, and Robert Blumenthal. 2019. Computing for the Social Good in Education. *ACM Inroads* 10, 4 (2019), 24–29. https://doi.org/10.1145/3368206

[40] Mareen Grillenberger. 2023. Why and How to Teach Physical Computing: Research and Practice in Computer Science Education at Secondary Schools. In *Teaching Coding in K-12 Schools: Research and Application*, Therese Keane and Andrew E. Fluck (Eds.). Springer International Publishing, Cham, 225–243. https://doi.org/10.1007/978-3-031-21970-2_15

[41] Maarten Grootendorst. 2022. BERTopic: Neural Topic Modeling with a Class-based TF-IDF Procedure. https://doi.org/10.48550/arXiv.2203.05794

[42] Shuchi Grover and Roy Pea. 2013. Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher* 42, 1 (2013), 38–43.

[43] Shuchi Grover and Roy Pea. 2023. Computational Thinking: A Competency Whose Time Has Come. In *Computer Science Education: Perspectives on Teaching and Learning in School* (2 ed.). Bloombury, UK.

[44] Teresa Guarda, Filipe Portela, and Jose Maria Diaz-Nafria (Eds.). 2024. *Advanced Research in Technologies, Information, Innovation and Sustainability: Third International Conference, ARTIIS 2023, Madrid, Spain, October 18–20, 2023, Proceedings, Part I.* Communications in Computer and Information Science, Vol. 1935. Springer Nature Switzerland, Cham. https://doi.org/10.1007/978-3-031-48858-0

[45] Mark Guzdial. 2022. Providing Students with Computational Literacy for Learning About Everything. In *Computational Thinking Education in K–12: Artificial Intelligence Literacy and Physical Computing*, Siu-Cheung Kong and Harold Abelson (Eds.). The MIT Press, USA, 29–48. https://doi.org/10.7551/mitpress/13375.003.0005

[46] D. Harel. 1987. *Algorithmics-the Spirit of Computing.* Addison Wesley, MA, USA.

[47] Jan Hauke and Tomasz Kossowski. 2011. Comparison of Values of Pearson's and Spearman's Correlation Coefficients on the Same Sets of Data. *Quaestiones Geographicae* 30, 2 (2011), 87–93.

[48] Sven Hüsing, Carsten Schulte, and Felix Winkelnkemper. 2023. Epistemic Programming. In *Computer Science Education: Perspectives on Teaching and Learning in School*, Sue Sentance, Erik Barendsen, and Carsten Schulte (Eds.). Vol. 2. Bloomsbury Academic, London, UK.

[49] Charles L. Isbell, Lynn Andrea Stein, Robb Cutler, Jeffrey Forbes, Linda Fraser, John Impagliazzo, Viera Proulx, Steve Russ, Richard Thomas, and Yan Xu. 2010. (Re)Defining Computing Curricula by (Re)Defining Computing. *SIGCSE Bull.* 41, 4 (Jan. 2010), 195–207. https://doi.org/10.1145/1709424.1709462

[50] Ole Sejer Iversen, Rachel Charlotte Smith, and Christian Dindler. 2018. From Computational Thinking to Computational Empowerment: a 21st Century PD Agenda. In *Proceedings of the 15th Participatory Design Conference: Full Papers - Volume 1 (PDC '18)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3210586.3210592

[51] Shiyan Jiang, Amato Nocera, Cansu Tatar, Michael Miller Yoder, Jie Chao, Kenia Wiedemann, William Finzer, and Carolyn P. Rosé. 2022. An Empirical Analysis of High School Students' Practices of Modelling with Unstructured Data. *British Journal of Educational Technology* 53, 5 (2022), 1114–1133. https://doi.org/10.1111/bjet.13253

[52] Yasmin Kafai, Cindi Proctor, and Denise Lui. 2019. From Theory Bias to Theory Dialogue: Embracing Cognitive, Situated, and Critical Framings of Computational Thinking in K-12 CS Education. In *Proceedings of the 2019 ACM Conference on International Computing Education Research.* ACM, USA, 101–109. https://doi.org/10.1145/3291279.3339400

[53] Yasmin B Kafai and Chris Proctor. 2022. A Revaluation of Computational Thinking in K–12 Education: Moving Toward Computational Literacies. *Educational Researcher* 51, 2 (2022), 146–151. Publisher: SAGE Publications Sage CA: Los Angeles, CA.

[54] Magnus Hoeholt Kaspersen, Karl-Emil Kjaer Bilstrup, Maarten Van Mechelen, Arthur Hjorth, Niels Olof Bouvin, and Marianne Graves Petersen. 2021. VotestratesML: A High School Learning Tool for Exploring Machine Learning and its Societal Implications. In *FabLearn Europe / MakeEd 2021 - An International Conference on Computing, Design and Making in Education* (St. Gallen, Switzerland) *(FabLearn Europe / MakeEd 2021)*. Association for Computing Machinery, New York, NY, USA, Article 3, 10 pages. https://doi.org/10.1145/3466725.3466728

[55] Nazish Zaman Khan and Andrew Luxton-Reilly. 2016. Is Computing for Social Good the Solution to Closing the Gender Gap in Computer Science?. In *Proceedings of the Australasian Computer Science Week Multiconference* (Canberra, Australia) *(ACSW '16)*. Association for Computing Machinery, New York, NY, USA, Article 17, 5 pages. https://doi.org/10.1145/2843043.2843069

[56] Barbara Kitchenham and Stuart Charters. 2007. *Guidelines for Performing Systematic Literature Reviews in Software Engineering.* Technical Report EBSE-2007-01. EBSE Technical Report, Software Engineering Group, School of Computer Science and Mathematics, Keele University and Department of Computer Science, University of Durham, Keele, Staffs, ST5 5BG, UK and Durham, UK.

[57] Clyde Kluckhohn. 1951. Values and Value-Orientations in the Theory of Action: An Exploration in Definition and Classification. In *Toward a General Theory of Action*, Talcott Parsons and Edward A. Shils (Eds.). Harvard University Press, USA, 388–433. https://doi.org/10.4159/harvard.9780674863507.c8

[58] Donald E. Knuth. 1974. Computer Programming as an Art. *Commun. ACM* 17, 12 (Dec. 1974), 667–673. https://doi.org/10.1145/361604.361612

[59] Donald E. Knuth. 1985. Algorithmic Thinking and Mathematical Thinking. *The American Mathematical Monthly* 92, 3 (March 1985), 170–181. https://doi.org/10.1080/00029890.1985.11971572

[60] Amy J. Ko, Alex Beitlers, Brooke Wortzman, Michael Davidson, Alyson Oleson, Ma Kirdani-Ryan, Stefania Druga, and James Everson. 2023. Critically Conscious Computing: Methods for Secondary Education. https://criticallyconsciouscomputing.org/

[61] Amy J. Ko, Alannah Oleson, Neil Ryan, Yim Register, Benjamin Xie, Mina Tari, Matthew Davidson, Stefania Druga, and Dastyni Loksa. 2020. It is Time for More Critical CS Education. *Commun. ACM* 63, 11 (Oct. 2020), 31–33. https://doi.org/10.1145/3424000

[62] Siu-Cheung Kong and Harold Abelson (Eds.). 2022. *Computational Thinking Education in K–12: Artificial Intelligence Literacy and Physical Computing*. The MIT Press, USA. https://doi.org/10.7551/mitpress/13375.001.0001

[63] Angelie Kraft and Eloïse Soulier. 2024. Knowledge-Enhanced Language Models Are Not Bias-Proof: Situated Knowledge and Epistemic Injustice in AI. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*. ACM, Rio de Janeiro Brazil, 1433–1445. https://doi.org/10.1145/3630106.3658981

[64] Kuzma Kukushkin, Yury Ryabov, and Alexey Borovkov. 2022. Digital Twins: A Systematic Literature Review Based on Data Analysis and Topic Modeling. *Data* 7, 12 (Nov. 2022), 173. https://doi.org/10.3390/data7120173

[65] Jean Lave, Etienne Wenger, and others. 1991. *Situated Learning: Legitimate Peripheral Participation*. Cambridge university press, UK.

[66] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and others. 2020. Retrieval-augmented Generation for Knowledge-intensive nlp Tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[67] B. Linck, L. Ohrndorf, S. Schubert, P. Stechert, J. Magenheim, W. Nelles, J. Neugebauer, and N. Schaper. 2013. Competence Model for Informatics Modelling and System Comprehension. In *2013 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, Berlin, Germany, 85–93. https://doi.org/10.1109/EduCon.2013.6530090

[68] Duri Long and Brian Magerko. 2020. What is AI Literacy? Competencies and Design Considerations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3313831.3376727

[69] Alejandro Lozano, Scott L. Fleming, Chia-Chun Chiang, and Nigam Shah. 2023. clinfo.ai: An Open-Source Retrieval-Augmented Large Language Model System for Answering Medical Questions using Scientific Literature. In *Biocomputing 2024*. WORLD SCIENTIFIC, Kohala Coast, Hawaii, USA, 8–23. https://doi.org/10.1142/9789811286421_0002

[70] Philip Machanick. 2003. Principles versus Artifacts in Computer Science Curriculum Design. *Computers & Education* 41, 2 (2003), 191–201. https://doi.org/10.1016/S0360-1315(03)00045-9

[71] Peter Madzík and Lukáš Falát. 2022. State-of-the-art on Analytic Hierarchy Process in the Last 40 years: Literature Review based on Latent Dirichlet Allocation Topic Modelling. *PLOS ONE* 17, 5 (May 2022), e0268777. https://doi.org/10.1371/journal.pone.0268777

[72] Johannes Magenheim and Carsten Schulte. 2006. Social, Ethical and Technical Issues in Informatics—An Integrated Approach. *Education and Information Technologies* 11, 3 (2006), 319–339. https://doi.org/10.1007/s10639-006-9012-6

[73] Johann S. Magenheim. 2001. *Deconstruction of Socio-technical Information Systems with Virtual Exploration Environments as a Method of Teaching Informatics*. Association for the Advancement of Computing in Education (AACE), Waynesville, NC. 00026.

[74] Michael S. Mahoney. 2005. The Histories of Computing(s). *Interdisciplinary Science Reviews* 30, 2 (June 2005), 119–135. https://doi.org/10.1179/030801805X25927

[75] Stefanie L Marshall and Ain A Grooms. 2022. Industry's Push for Computer Science Education: Is Computer Science Really for All? *Policy Futures in Education* 0, 0 (Jan. 2022), 14782103211045601. https://doi.org/10.1177/14782103211045601 Publisher: SAGE Publications.

[76] Toleen Mazloum. 2024. AI and Research: Benefits and Challenges. *Clinical Oral Science and Dentistry* 6, 3 (2024), 1–10.

[77] Oliver McGarr and Keith Johnston. 2020. Curricular Responses to Computer Science Provision in Schools: Current Provision and Alternative Possibilities. *Curric. j.* 31, 4 (Nov. 2020), 745–756. https://doi.org/10.1002/curj.40

[78] Michael S. Merry. 2005. Cultural Coherence and the Schooling for Identity Maintenance. *Journal of Philosophy of Education* 39, 3 (Aug. 2005), 477–497. https://doi.org/10.1111/j.1467-9752.2005.00449.x

[79] Pekka Mertala. 2021. The Pedagogy of Multiliteracies as a Code Breaker: A Suggestion for a Transversal Approach to Computing Education in Basic Education. *British Journal of Educational Technology* 52, 6 (2021), 2227–2241. https://doi.org/10.1111/bjet.13125

[80] Janet Morales-Chicas, Melina Castillo, Isabel Bernal, Paola Ramos, and Beatriz L. Guzman. 2019. Computing with Relevance and Purpose: A Review of Culturally Relevant Education in Computing. *International Journal of Multicultural Education* 21, 1 (2019), 125–155.

[81] Liliana Morales-Navarro and Yasmin B. Kafai. 2023. Conceptualizing Approaches to Critical Computing Education: Inquiry, Design, and Reimagination. In *Past, Present and Future of Computing Education Research*, Maarit Apiola, Susana López-Pernas, and Mohamed Saqr (Eds.). Springer, Cham. https://doi.org/10.1007/978-3-031-25336-2_21

[82] Fatemeh Mosaiyebzadeh, Seyedamin Pouriyeh, Reza Parizi, Nasrin Dehbozorgi, Mohsen Dorodchi, and Daniel Macêdo Batista. 2023. Exploring the Role of ChatGPT in Education: Applications and Challenges. In *The 24th Annual Conference on Information Technology Education*. ACM, Marietta GA USA, 84–89. https://doi.org/10.1145/3585059.3611445

[83] Zachary Munn, Micah Peters, Cindy Stern, Catalin Tufanaru, Alexa McArthur, and Edoardo Aromataris. 2018. Systematic Review or Scoping Review? Guidance for Authors when choosing between a Systematic or Scoping Review Approach. *BMC Medical Research Methodology* 18 (2018), 1–7. Publisher: Springer.

[84] Greg L. Nelson and Amy J. Ko. 2018. On Use of Theory in Computing Education Research. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (Espoo, Finland) *(ICER '18)*. Association for Computing Machinery, New York, NY, USA, 31–39. https://doi.org/10.1145/3230977.3230992

[85] Davy Tsz Kit Ng, Jac Ka Lok Leung, Samuel Kai Wah Chu, and Maggie Shen Qiao. 2021. Conceptualizing AI Literacy: An Exploratory Review. *Computers and Education: Artificial Intelligence* 2 (Jan. 2021), 100041. https://doi.org/10.1016/j.caeai.2021.100041

[86] Li Ni, Gabriella Bausch, Emily Thomas-Cappello, Fred Martin, and Brooke Feliciano. 2024. Creating Apps for Community and Social Good: Preliminary Learning Outcomes from a Middle School Computer Science Curriculum. *ACM Transactions on Computing Education* 24, 3 (2024), 32:1–32:26. https://doi.org/10.1145/3658674

[87] National Academy of Sciences. 1999. *Being Fluent with Information Technology*. Technical Report. National Academy Press.

[88] Geir Ottestad and Greta Björk Gudmundsdottir. 2018. Information and Communication Technology Policy in Primary and Secondary Education in Europe. In *Second Handbook of Information Technology in Primary and Secondary Education*, Joke Voogt, Gerald Knezek, Rhonda Christensen, and Kwok-Wing Lai (Eds.). Springer International Publishing, Cham, 1–21. https://doi.org/10.1007/978-3-319-53803-7_92-1

[89] Seymour Papert. 1980. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Inc., USA.

[90] Seymour Papert and Idit Harel. 1991. Situating Constructionism. In *Constructionism*. Ablex Publishing Corporation, USA. http://www.papert.org/articles/SituatingConstructionism.html

[91] Blikstein Paulo and Moghadam Sepi Hejazi. 2019. Computing Education: Literature Review and Voices from the Field. In *The Cambridge Handbook of Computing Education Research*. Cambridge University Press, Cambridge, 56–78.

[92] Mitchel Resnick. 1998. Technologies for Lifelong Kindergarten. *Educational Technology Research & Development* 46, 4 (1998), 43–55.

[93] Mitchel Resnick. 2017. *Lifelong Kindergarten: Cultivate Creativity Through Projects, Passion, Peers, and Play*. MIT Press, USA.

[94] Richard M Ryan and Edward L Deci. 2017. *Self-Determination Theory: Basic Psychological Needs in Motivation, Development and Wellness*. The Guilford Press, New York and London.

[95] Jean J. Ryoo and Takeria Blunt. 2024. "Show Them the Playbook That These Companies Are Using": Youth Voices about Why Computer Science Education Must Center Discussions of Power, Ethics, and Culturally Responsive Computing. *ACM Trans. Comput. Educ.* 24, 3, Article 34 (Aug 2024), 21 pages. https://doi.org/10.1145/3660645

[96] Jean J. Ryoo, Alicia Morris, and Jane Margolis. 2021. "What Happens to the *Raspado* man in a Cash-free Society?": Teaching and Learning Socially Responsible Computing. *ACM Transactions on Computing Education* 21, 4 (Dec. 2021), 1–28. https://doi.org/10.1145/3453653

[97] Rafi Santo, Leigh Ann DeLyser, June Ahn, Anthony Pellicone, Julia Aguiar, and Stephanie Wortel-London. 2019. Equity in the Who, How and What of Computer Science Education: K12 School District Conceptualizations of Equity in 'CS for All' Initiatives. In *2019 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT)*. IEEE, USA, 1–8. https://doi.org/10.1109/RESPECT46404.2019.8985901

[98] Michael Schad and W. Monty Jones. 2020. The Maker Movement and Education: A Systematic Review of the Literature. *Journal of Research on Technology in Education* 52, 1 (Jan. 2020), 65–78. https://doi.org/10.1080/15391523.2019.1688739

[99] Sigrid Schubert and Andreas Schwill. 2011. *Didaktik der Informatik* (2. aufl ed.). Spektrum, Akad. Verl, Heidelberg.

[100] Carsten Schulte, Malte Hornung, Sue Sentance, Valentina Dagiene, Tatjana Jevsikova, Neena Thota, Anna Eckerdal, and Anne-Kathrin Peters. 2012. Computer science at school/CS teacher education: Koli working-group report on CS at school. In *Proceedings of the 12th Koli Calling International Conference on Computing Education Research (Koli Calling '12)*. Association for Computing Machinery, New York, NY, USA, 29–38. https://doi.org/10.1145/2401796.2401800

event-place: Koli, Finland.

[101] Richard J Shavelson, Lisa Towne, National Research Council (U.S.), and Committee on Scientific Principles for Education Research. 2002. *Scientific Research in Education - Carsten comm.* National Academy Press, Washington, DC. http://public.eblib.com/choice/publicfullrecord.aspx?p=3375187

[102] Valerie J. Shute, Chen Sun, and Jodi Asbell-Clarke. 2017. Demystifying Computational Thinking. *Educational Research Review* 22 (2017), 142–158.

[103] Rachel Charlotte Smith, Marie-Monique Schaper, Mariana Aki Tamashiro, Maarten Van Mechelen, Marianne Graves Petersen, and Ole Sejer Iversen. 2023. A Research Agenda for Computational Empowerment for Emerging Technology Education. *International Journal of Child-Computer Interaction* 38 (Dec. 2023), 100616. https://doi.org/10.1016/j.ijcci.2023.100616

[104] Hannah Snyder. 2019. Literature Review as a Research Methodology: An Overview and Guidelines. *Journal of Business Research* 104 (Nov. 2019), 333–339. https://doi.org/10.1016/j.jbusres.2019.07.039

[105] Charles Spearman. 1987. The Proof and Measurement of Association Between Two Things. *The American Journal of Psychology* 100, 3/4 (1987), 441–471.

[106] Xiaodan Tang, Yue Yin, Qiao Lin, Roxana Hadad, and Xiaoming Zhai. 2020. Assessing computational thinking: A Systematic Review of Empirical Studies. *Computers & Education* 148 (April 2020), 103798. https://doi.org/10.1016/j.compedu.2019.103798

[107] Matti Tedre. 2006. The development of computer science: a sociocultural perspective. In *Proceedings of the 6th Baltic Sea Conference on Computing Education Research: Koli Calling 2006* (Uppsala, Sweden) *(Baltic Sea '06)*. Association for Computing Machinery, New York, NY, USA, 21–24. https://doi.org/10.1145/1315803.1315808

[108] Matt Tedre. 2014. *The Science of Computing: Shaping a Discipline.* CRC Press / Taylor & Francis, New York, NY, USA.

[109] Matti Tedre and Mikko Apiola. 2013. Three Computing Traditions in School Computing Education. In *Improving Computer Science Education*, D. M. Kadijevich, C. Angeli, and C. Schulte (Eds.). Routledge, New York, NY and London, 100–116. 00013.

[110] Matti Tedre and John Pajunen. 2022. Grand Theories or Design Guidelines? Perspectives on the Role of Theory in Computing Education Research. *ACM Trans. Comput. Educ.* 23, 1, Article 4 (Dec. 2022), 20 pages. https://doi.org/10.1145/3487049

[111] Matti Tedre, Simon, and Lauri Malmi. 2018. Changing Aims of Computing Education: A Historical Survey. *Computer Science Education* 28, 2 (2018), 158–186.

[112] Matti Tedre and Henriikka Vartiainen. 2023. K-12 Computing Education for the AI Era: From Data Literacy to Data Agency. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1* (Turku, Finland) *(ITiCSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1–2. https://doi.org/10.1145/3587102.3593796

[113] Jakita O Thomas, Quincy K Brown, and Jamika Burge. 2024. Introduction to the Special Issue on Situating the Intersectional Experiences of Black Girls and Women in Computing & Technology. , 5 pages.

[114] Mike Tissenbaum, Josh Sheldon, and Hal Abelson. 2019. From Computational Thinking to Computational Action. *Communincations of the ACM* 62, 3 (Feb. 2019), 34–36. https://doi.org/10.1145/3265747

[115] Lisa Towne and Richard J. Shavelson (Eds.). 2002. *Scientific Research in Education.* National Academies Press, Washington, D.C.

[116] Allen B. Tucker. 1991. A Summary of the ACM/ IEEE-CS Joint Curriculum Task Force Report: Computing Curricula. *Commun. ACM* 34, 6 (Jun 1991), 68–84. https://doi.org/10.1145/103701.103710

[117] Jan van den Akker. 2003. Curriculum Perspectives: An Introduction. In *Curriculum Landscapes and Trends*, Jan van den Akker, Wilmad Kuiper, and Uwe Hameyer (Eds.). Springer Netherlands, Dordrecht, 1–10. https://doi.org/10.1007/978-94-017-1205-7_1

[118] Jan van den Akker. 2013. *2. Curricular Development Research as a Specimen of Educational Design Research.* Netherlands Institute for Curriculum Development (SLO), Enschede, Enschede, Nederlands, 53.

[119] Jan H. van Driel, Astrid M. W. Bulte, and Nico Verloop. 2008. Using the curriculum emphasis concept to investigate teachers' curricular beliefs in the context of educational reform. *Journal of Curriculum Studies* 40, 1 (Feb. 2008), 107–122. https://doi.org/10.1080/00220270601078259 00097.

[120] Emiliana Vegas, Michael Hansen, and Brian Fowler. 2021. *Building Skills for Life (Brookings report).* Technical Report. The Brookings Institution.

[121] Emiliana Vegas, Michael Hansen, and Brian Fowler. 2021. *Building Skills for Life: how to expand and improve computer science education around the world.* Policy Report. Brookings Institution. https://www.brookings.edu/essay/building-skills-for-life-how-to-expand-and-improve-computer-science-education-around-the-world/

[122] J. Ángel Velázquez-Iturbide. 2013. An Experimental Method for the Active Learning of Greedy Algorithms. *ACM Transactions on Computing Education* 13, 4 (Nov. 2013), article 18. https://doi.org/10.1145/2534972

[123] Sara Vogel, Rafi Santo, and Dixie Ching. 2017. Visions of Computer Science Education: Unpacking Arguments for and Projected Impacts of CS4All Initiatives. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education.* ACM, Seattle Washington USA, 609–614. https://doi.org/10.1145/3017680.3017755

[124] Lev S. Vygotsky. 1978. *Mind in society.* Harvard University Press, USA. https://doi.org/10.2307/j.ctvjf9vz4 tex.editorbtype: redactor.

[125] Cameron Wilson, Leigh Ann Sudol, Chris Stephenson, and Mark Stehlik. 2010. *Running on Empty: the Failure to Teach K–12 Computer Science in the Digital Age.* ACM, New York, NY, USA. https://doi.org/10.1145/3414583

[126] Jeannette M. Wing. 2006. Computational Thinking. *CACM* 49, 3 (March 2006), 33–35. https://doi.org/10.1145/1118178.1118215

[127] Felix Winkelnkemper, Lukas Höper, and Carsten Schulte. 2024. ARIadne – An Explanation Model for Digital Artefacts. *Informatics in Education* 23, 2 (2024), 479–505. https://doi.org/10.15388/infedu.2024.09

[128] Benjamin S Wohl, Sophie Beck, and Lynne Blair. 2017. The Future of the Computing Curriculum: How the Computing Curriculum Instills Values and Subjectivity in Young People. *International Journal of Computer Science Education in Schools* 1, 1 (2017), 21–29.

[129] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. https://doi.org/10.48550/arXiv.2309.07597

[130] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. https://doi.org/10.48550/arXiv.2309.01219 arXiv:2309.01219 [cs]