New Perspectives on the Future of Computing Education: Teaching and Learning Explanatory Models

Lukas Höper
Computer Science, Paderborn University
Germany
lukas.hoeper@uni-paderborn.de

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

Abstract

This paper introduces the explanatory model approach to address challenges in computing education arising from rapid technological developments and paradigm shifts, particularly regarding artificial intelligence and machine learning. Traditional approaches in computing education aim to teach basic concepts derived from the computer science discipline as they are in order to support students' understanding of these concepts and digital technologies that implement these concepts. This approach is challenged in topics like machine learning, where the ground truth of the inner workings and the behaviors of these technologies is not so clear, making rethinking approaches in computing education necessary. The explanatory model approach suggests that students learn models about computational concepts and digital artifacts that help them understand, explain, and reason about digital technologies. While drawing on the notion of models in science and science education, this approach emphasizes learning and using explanatory models as a focal point in computing classes. Doing so may help students make use of these models as tools and enable them to reflect on and critique different models in various contexts. Additionally, this paper discusses how making explanatory models explicit in research can enrich computing education research and our discourses and describes avenues for researching explanatory models as different perspectives on computational concepts.

CCS Concepts

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

Keywords

K-12, computing education, explanatory models, artificial intelligence, computational concepts

ACM Reference Format:
Lukas Höper and Carsten Schulte. 2024. New Perspectives on the Future of Computing Education: Teaching and Learning Explanatory Models. In *24th Koli Calling International Conference on Computing Education Research (Koli Calling '24), November 12–17, 2024, Koli, Finland*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3699538.3699558

Koli Calling '24, November 12–17, 2024, Koli, Finland
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1038-4/24/11
https://doi.org/10.1145/3699538.3699558

1 Introduction

In this paper, we present and discuss the approach of *explanatory models* that serve as tools for educational diagnostics, teaching, and as an area of research in computing education. To introduce this idea, we use an example of a workshop concept for teacher education and professional development recently published [3]. In this workshop, participants were given several tasks and materials to engage in discussions about the question of what constitutes an algorithm. The authors observed that over the past decade, participants could only partially identify the elements of an algorithm agreed upon in computer science (CS). They noted being surprised that almost all student groups overlooked a specific perspective (i.e. that algorithms target a particular implementation device). The participants use different conceptions or notions of an algorithm and can, for example, intensely discuss the analogy of cooking recipes for algorithms. So, despite a consensus on the meaning of algorithm and given that this concept can be explained in terms of its ground truth, people often hold divergent perspectives and conceptions. The explanatory model approach proposed in this paper provides a theoretical framework as a foundation for describing different perspectives on such computational concepts or digital technologies.

From a computing education perspective, interventions about such topics (e.g., algorithms) often focus on the ultimate goal of teaching students the correct understanding of the computational concepts in line with the common understanding within the CS discipline. Traditional contents in computing classes are defined with the ground truth of the respective concepts. However, analogies and similar ideas are sometimes used when understanding the concepts is challenging. For example, in the context of programming, the idea of notional machines was introduced as a pedagogical vehicle to support students in understanding programs and their behavior during execution [see 17, 48]. Thus, analogies or notional machines are intended to scaffold and support students in developing a complete and correct understanding of the computational concepts as the intended learning outcomes. However, nowadays, computing education involves topics where this ground truth is not so clear; think, for example, about artificial intelligence (AI) and machine learning (ML) and what a correct understanding of large language models could be: it is discussed, for example, whether there are 'sparks of intelligence' [7] or whether they are rather like 'stochastic parrots' [5] [for detailed discussions, see 6, 31]. Such cases raise the question of whether a correct, ground truth understanding is achievable in computing education (at least in schools). With the approach presented here, we suggest taking *explanatory models as explicit content and learning them as an end goal in itself* instead of only as a scaffold during the learning process until the complete and correct understanding is achieved. Explanatory models provide explanations of computational concepts or technologies, such as how large language models work, keeping in mind that sometimes even different explanations exist. Thus, this approach may suggest rethinking what exactly it is that we teach computing students in K-12 (and maybe beyond).

The introductory example should also hint at another perspective on explanatory models, that is, *to make the explanatory models used in computing education research explicit*. This can facilitate comparisons across studies about the same or similar explanatory models (e.g., of the concept of algorithms) but different pedagogical approaches. For example, this approach allows us to say that people are using different explanatory models (e.g., of algorithms) and hence argue differently about questions like whether cooking recipes can be understood as algorithms. Additionally, it could allow us to relate two interventions or educational tools that aim to teach the same explanatory model but have entirely different pedagogical approaches. Being aware of the explanatory models used and explicitly communicating them could enhance discourses within the computing education community and contribute to cumulative knowledge and theory building. Furthermore, explicating explanatory models about computational concepts and digital technologies could also provide lenses on research opportunities, such as systematically examining the explanatory models students hold about specific computational concepts.

This idea of explicitly teaching and researching explanatory models leads to a third perspective on the approach presented here: *the notion of models should be considered*. In science and science education, using models has a long tradition, including research on teaching and learning about and with models. This research can support discussions on teaching explanatory models in computing education, even if the distinctions between science education and computing education should be taken into account. An essential aspect of this research is to be aware that models are not full images of reality but are selective views, covering specific aspects while leaving out others [see 12]. For example, an explanatory model for computational concepts could capture some essential features or dynamics, allowing students to test hypotheses and explore the behavior of digital artifacts. In doing so, an explanatory model is related to one specific purpose so that such a model's view on a computational concept or digital artifact could be suitable for a particular use case but not for other contexts.

Overall, the idea of explanatory models extends across different perspectives on computing education, its discourses, and research. In the following, we discuss the theoretical foundation before delving deeper into describing the explanatory model approach and the perspectives mentioned. After proposing the explanatory model approach, we reflect on promising avenues for further research and call for more discussions on explanatory models, which may be currently used implicitly in computing education research.

2 Background and Context

Current challenges for computing education posed by new topics like AI and ML are briefly discussed below, which leads to the idea of teaching 'explanatory models', providing explanations for computational concepts, digital artifacts, or socio-technical systems. We also discuss the relation to notional machines as a similar concept to explanatory models and then reflect on the notion of models and their use and conception in science and science education. We believe that respective experiences from science education can be useful for explanatory models for computing education.

2.1 Challenges and Paradigm Shifts in AI education

Recent discussions in computing education research demonstrate uncertainties about which ideas, concepts and perceptions should be taught regarding topics like ML [e.g., 22, 43, 49]. These challenges are further underscored as understanding these technologies in detail is problematic within CS discipline too, resulting in research areas trying to find explanations for ML models (e.g., see discussions about explainable AI). Below, we discuss these challenges as a foundation for the development of the explanatory model approach.

Increasing complexity and decreasing comprehensibility. The challenges in understanding AI systems and comprehending their specific outputs become clear when taking into account the fast-growing research field about making AI systems (particularly those using ML techniques) understandable to people: explainable AI (XAI) [for an overview, see 2]. XAI research reports that AI systems with higher accuracy are less likely to be comprehensible and explainable [see 1, 23]. The current technological developments make understanding digital technologies increasingly complex and less transparent, while their implementation in daily life makes comprehensibility increasingly necessary.

The algorithms and the code for designing AI systems may be simple and comprehensible, but the resulting ML models can still be very complex and are rather black-boxes [32, 42]. Notably, the algorithms and codes play an important but relatively small role in ML systems [46]. Thus, understanding the code does not sufficiently explain the behavior of such systems. It also requires considering the role of data (e.g., training data selection) and their impact on the system's behavior. Rahwan et al. [42] discuss different aspects of the complexity of such systems, including the high dimensionalities of ML models and the massive amounts of training data, imperfections in data, or the fact that the workings for generating output are learned by an ML system and not designed by the designer. Hence, AI systems and especially data-driven technologies are hard to understand. Knowing underlying conceptual ideas implemented in these technologies is probably insufficient for understanding them and their behavior. Thus, learning abstract concepts may not necessarily help students understand such technologies.

This demonstrates that the approaches we use to understand traditional algorithmic systems may not be sufficient for understanding data-driven systems, raising the question of what should be taught to students to effectively support their understanding of such technologies.

Paradigm shift in teaching about digital technologies. Similar discussions about differences in teaching about algorithmic systems and data-driven approaches can also be found in computing education research. For example, Tedre et al. [49] discuss differences in problem-solving and designing algorithmic solutions compared to data-driven problem-solving, highlighting paradigm shifts from traditional algorithms to ML. In recent years, many educational approaches for teaching students about AI concepts have been developed and discussed [e.g., 9, 22, 34, 43]. A review from Rizvi et al. [43] has shown that most of the materials published in articles about K-12 AI education do not cover concepts on the engines level (i.e. the concrete and formal underlying functionalities of AI algorithms). Instead, current materials seem to focus on teaching ML models (e.g., training and testing such models) and designing small AI-based applications [see 43]. This indicates that many educational approaches to AI education currently focus on explaining ML technologies and respective computational concepts instead of seeking to teach students about technical "ground truth" on an engine level. A similar observation can be made regarding one promising trend of using and developing educational tools, offering easy options for designing ML applications without requiring prior programming experiences [e.g., 24, 28, 55] [for an overview, see 21]. In doing so, students get insights on some aspects of the black-boxes of ML systems, while teaching focuses on a more abstract level, like training and testing ML models.

These debates on different paradigms and levels of teaching about AI and using educational tools seemingly focus on different contents and pedagogic approaches. However, there is probably an underlying question related to the idea of explanatory models: Which aspects of AI and ML should students learn? The different approaches, levels, and tools may target the same or different explanatory models. An idea argued in recent discussions is to build on our experiences from computing education about traditional, algorithmic systems, that is, adopting respective educational approaches to support students in developing mental models about AI systems [22]. For example, it is suggested to develop notional machines for AI systems [49]. Hence, we discuss this idea below as a foundation for developing the explanatory model approach, even if notional machines and explanatory models have significant differences, as we argue later.

2.2 Notional Machines as Vehicles to Understand Programs

The concept of notional machines is used in programming education and is to a certain extent similar to explanatory models [for overviews, see 17, 48]. A notional machine helps learners understand how programs and programming languages work by explaining what happens during program execution. It is a model for an idealized and conceptual computer [15] used as "a pedagogic device to assist the understanding of some aspect of programs or programming" [17]. It supports explaining programs, their behavior, and user-understandable semantics. For instance, notional machines can be analogies, such as boxes representing variables [for more examples, see 17]. Usually, a notional machine covers specific aspects (e.g., variables) but omits others. Based on mental model theory, Sorva [48] emphasizes explicitly teaching national machines rather than having them as implicit goals. Similarly, Munasinghe et al. [37] note that notional machines are often taught implicitly as vehicles or scaffolds rather than explicit models. This aligns with the mentioned perspective that often understanding the real ground truth is seen as the ultimate goal for computing education instead of learning about models: Even if notional machines are an example of teaching and learning models, they merely serve as pedagogical vehicles or didactic means to achieve a correct understanding, but not as the goal in itself. Understanding traditional algorithms led to grasping the ground truth (e.g., how variables work), making the need for models as end goals less critical, similar to the engine level of learning about AI described in the SEAME framework [47] [see also 43]. However, this seems to be changing with ML technologies (or even complex systems), where it is not so clear how they work in the smallest detail. Regarding AI education, developing notional machines for AI systems is suggested, although these may differ fundamentally from traditional notional machines [51]. For example, Munasinghe et al. [37] envision more abstract notional machines for AI systems.

Notional machines could be understood as one particular type of explanatory models related to program executions and their behavior. However, teaching notional machines focuses on providing a scaffold rather than primarily teaching models, in contrast to the idea of teaching explanatory models. While arguing for teaching explanatory models, a foundation of a notion of models is needed to be aware of essential characteristics of explanatory *models*.

2.3 Notion of 'Model' and its Role in Science and Science Education

In science research, developing models is a fundamental scientific practice used for reasoning and sense-making [40]. For instance, think about well-known models in natural sciences, like the models of atoms by Thomson, Rutherford, and Bohr or the DNA helix model by Watson and Crick. According to Osborne [39], models help when considered phenomena are not directly accessible (e.g., when they are too big or small to be observable). Due to the integral role of models in natural sciences, teaching and learning with and about models is essential in science education to reflect the scientific disciplines authentically [e.g., 39, 40, 53]. Science education research provides insights regarding teaching and learning with and about models, including several challenges [12]. For example, understanding models as selective and idealized representations of something with a specific state of scientific knowledge (and maybe resulting in models showing to be limited or outdated) rather than true images of reality can be challenging for learners [see 12].

While a broad discussion of the term 'model' exceeds this paper's scope, we note some insightful perspectives involving different disciplines concerned with the notion and meaning of models [for historical overviews, see 35]. The mathematician and theoretical computer scientist Mahr [35] describes models as representations that can be simplified and abstracted but could include additional properties. He introduces the 'cargo' idea, describing that models convey information or knowledge about what they are meant to be used for. Regarding natural sciences, models are often characterized with two dimensions: they are representations *of* something and serving as tools *for* something [19, 40, 53]. They allow for illustrating, explaining, and communicating about phenomena. This scientific function of models is described with models as media (referring to the *of*-perspective) [53]. Additionally, models provide predictions for phenomena so that they can be used as a tool for generating ideas and knowledge about the considered phenomena (referring to the *for*-perspective) [53]. Thus, models have a representative perspective and an explanatory and predictive function. Notably, the development and use of a model is related to specific purposes and intentions [18]. Passmore, Gouvea, and Giere [40] argue that the of-for-distinction can support to underscore the functional perspective of using models, highlighting that science education is not limited to the fact that models are one more thing students need to learn and reproduce but instead being enabled to use models as tools for reasoning and sense-making. However, science education practice often focuses on the representative perspective and neglects functional perspectives [19]. From a computing education view, this functional perspective could relate to using models to explain computational concepts and to explore, explain, and reason about specific digital artifacts and their inner workings.

In science, various types of models are used, such as mathematical, theoretical, chemical, or analogical models [see 11]. As one type, Clement [11] describes explanatory models as explaining how and why something works. In addition, scientific, theoretical, and hypothesized models provide views on the world, but they should be distinct from just observational models (e.g., measurements or representative descriptions). Respective models used in science education are often developed, tested, and refined by scientists over time (e.g., involving experiments).

When relating the role of models in science (education) to computing education, crucial differences between these disciplines should be noted. For example, science education considers natural phenomena, while computing education is about digital artifacts, referring to objects invented and made by humans. We discuss this later in more detail.

3 Explanatory Models

In this section, we describe the *explanatory model* approach, explain the term, and discuss its notion. Figure 1 provides an overview of its characteristics. Then, we outline their potential role in teaching and learning processes and computing education research.

3.1 Characterization of Explanatory Models

What is an explanatory model? An explanatory model is a conceptual model that is an idealized representation of computing objects, such as computational concepts, digital artifacts, or socio-technical systems (i.e. a composition of human beings and digital artifacts). Such models should fulfill educational purposes of providing specific perspectives on and explanations of computational concepts and digital artifacts and their behavior. For example, an explanatory model for data-driven technologies can focus on the role of data [see 25, 26]. They allow students to explain computational concepts or explore and make sense of inner workings and contextual effects of specific digital artifacts.

Why do we use the term 'explanatory models'? While traditional concepts and algorithmic systems could be explained in full detail with the technical truth, this is challenged by the complexity of real-life digital artifacts, current technological developments, and paradigm shifts in the discipline (e.g., regarding AI and ML). Based on the notion of models discussed earlier, the following perspectives could be adopted (especially from the of-for-distinction): First, explanatory models represent computational concepts and digital artifacts, thereby focusing on specific aspects. Frameworks from computing education could assist in this regard, for example, to provide different levels of consideration on AI systems [e.g., 47] or a set of various aspects for programs [e.g., 33, 44]. This representational perspective of explanatory models can cover different degrees of complexity and technical precision. For example, an explanatory model of large language models could use the metaphor of stochastic parrots [see 5]. Second, explanatory models serve as tools for exploring and explaining digital artifacts, which involves making sense of their inner workings and reasoning about their behavior. From this perspective, an explanatory model highlights different aspects of digital artifacts and serves as a tool for uncovering inner workings of specific technologies. For example, a mentioned explanatory model of a large language model could then be used to explore and explain the outputs of a specific text generation tool.

Even if building on the notion of models from science, the representational view of models in computing education differs from that in natural sciences. For example, when considering a stone (as a natural object), the question of whether it is a good stone makes no sense as it is just a stone; in contrast, digital artifacts are based on human decisions and values so such a question makes sense. This example illustrates that explanatory models related to digital artifacts or socio-technical systems require other characteristics than models in science and science education. The dual nature theory supports this characteristic property of explanatory models, which stems from philosophical debates on the nature of technical artifacts and is used, for example, in computing or technology education [e.g., 10, 44]. While comparing explaining natural phenomena and digital artifacts, de Ridder [13] emphasizes that digital artifacts should be considered with the human agency, while this is different about natural phenomena. According to this theory, two views can be referred to when understanding or explaining digital artifacts [30, 44, 54]. While the perspective of the architecture or structure of digital artifacts refers to its inner workings (e.g., the algorithmic workings), the perspective of the relevance is related to an interpretation of the intentions, meanings, and social effects of the digital artifact (e.g., why it was designed as it is, or how can it be used). Notably, different aspects of a digital artifact (e.g., an algorithm) can be described from both perspectives. The theory states that digital artifacts are not neutral as different human purposes and intentions are included. The two perspectives are interrelated so that understanding the relevance is necessary to comprehend inner workings; conversely, understanding the architecture is a prerequisite for evaluating the relevance [see 30, 45]. Accordingly, it is argued that a comprehensive understanding of a digital artifact involves both perspectives. For example, digital artifacts often impact individuals and society. Rahwan et al. [42] describe several examples of potential influences of AI systems. For instance, systems choose information (or misinformation) people see in their news feeds, which can influence individuals' behaviors, emotions, and opinions [29, 52]. Accordingly, models about digital artifacts should not be representations of inner workings alone (the architecture perspective) but also - and this is important - should provide explanations for their functions, meanings, and impacts (the relevance perspective).

In summary, explanatory models always draw on architecture and relevance perspectives to highlight the interpretative and explanatory functions. This allows students to reason about digital artifacts, their behavior, and their influences on individuals and society.

Important difference between science and computing. Models in natural science and science education are developed and used to understand natural phenomena. In contrast, in computing education, models and modeling are used to understand and develop digital artifacts rather than primarily to understand given phenomena of the analog world. (Note that some digital artifacts are indeed designed to examine and understand natural phenomena, such as computational simulations, but this specific case deserves in-depth discussion [e.g., 41].) However, designing digital artifacts in computing also impacts the analog world, which is also considered in computing but always in relation to digital artifacts. Thus, models for computing education have other and additional functions than those known in science. Nevertheless, the experiences in teaching and learning with and about models in science education can be relevant to teaching and learning explanatory models, especially those related to technologies we struggle to understand and explain. This is similar to the argumentation of Rahwan et al. [42], who argue for adopting methods from natural sciences to investigate and understand the behavior of AI systems.

3.2 Explanatory Models in Teaching and Learning Processes

Based on the characteristics of explanatory models, we discuss use cases for teaching and learning practice and research in computing education.

Students learn explanatory models. Similarly to notional machines that are argued to be explicitly taught [see 17, 48], explanatory models are intended to be made explicit, that is, learning them becomes a concrete learning goal. However, while notional machines are meant as scaffolds or vehicles to support students in learning the ground truth of algorithmic programs, explanatory models are meant as a goal in themselves, including teaching students about these models and enabling them to work with them. Thereby, explanatory models aim to support students' metacognitive thinking processes, such as reflecting on different perspectives on computational concepts or the nature of computing; similarly, in science education, learning about and with models is argued as serving metacognitive tools [e.g., 12]. They provide a perspective or lens on computational concepts and digital artifacts. For example, this allows them to explain abstract concepts (e.g., algorithms or large language models) or to explore, explain, and reason about specific digital artifacts, their inner workings, and their behavior (e.g., particular chatbots). This could include using an explanatory model as a lens on large-language model applications, allowing them to find explanations for their behavior, such as reasoning about the generated text. Explanatory models can also be used for designing digital artifacts, such as applying an explanatory model of neural networks to design an ML application. Explicitly learning and using explanatory models also involves learning about their purposes, allowing students to critique and reflect on the contexts in which they can use a model or when it may not be applicable.

Educators use explanatory models. Following explanatory models as learning content, this approach can also help educators design teaching units and materials accordingly. Teaching about explanatory models is meant to support students in forming respective mental models, which are mental constructs (i.e. knowledge structure), for example, related to systems or algorithmic processes [27]. Mental model theory describes how people perceive, explain, and predict the world [20]. It is often used in computing education research, like in the approach of notional machines [e.g., 17] or when examining students' conceptions about topics like AI [e.g., 36]. While mental models are personal and internal representations, the external counterparts are conceptual models [20, 38, 48]; like explanatory models. Their representations could include analogies, artifacts, or visual explanations of systems [38, 48]. Conceptual models can be useful in teaching to explain structures and inner workings of digital artifacts [48]. For example, a study from Ben-Ari and Yeshno [4] indicates that learning a conceptual model about internal word-processing software supported school students in developing conceptual understanding and interacting with the software. Similarly, a study suggests that learning an explanatory model can help students understand everyday technologies [26]. However, based on experiences in science education, we should note that presenting conceptual models does not necessarily lead to adequate mental models as students often lack knowledge from the discipline or the domain to interpret the presented models [20].

This relation of explanatory models to intended mental models leads us to their potential support for diagnosing students' conceptions. Research on students' perspectives on computational concepts or their preconceptions can be understood as examining explanatory models students know, such as in the study about algorithms mentioned before [see 3]. This can help teachers choose explanatory models to introduce in a learning group. Additionally, explicating these explanatory models can help students reason about computational concepts. For example, using a recipe analogy as an explanatory model of algorithms could include discussing which aspects of algorithms this analogy could adequately represent (e.g., step-wise operations) and where it has its limitations (e.g., loops may then be problematic).

While using notional machines or analogies as vehicles to teach computational concepts is not new, the presented approach calls for making explanatory models explicit and taking learning and using them as one of the primary learning goals. Then, explanatory models serve as learning content but also provide diagnostic functions and can support developing interventions.

3.3 Explanatory Models in Computing Education Research

In addition to using explanatory models in computing education practice, we envision further potential use in research practice. The diagnostic function of explanatory models mentioned above also applies to research practice, such as studies on students' conceptions of computational concepts. Thus, explanatory models can serve as lenses for empirical research, inspiring the development of respective research instruments. Explicating explanatory models

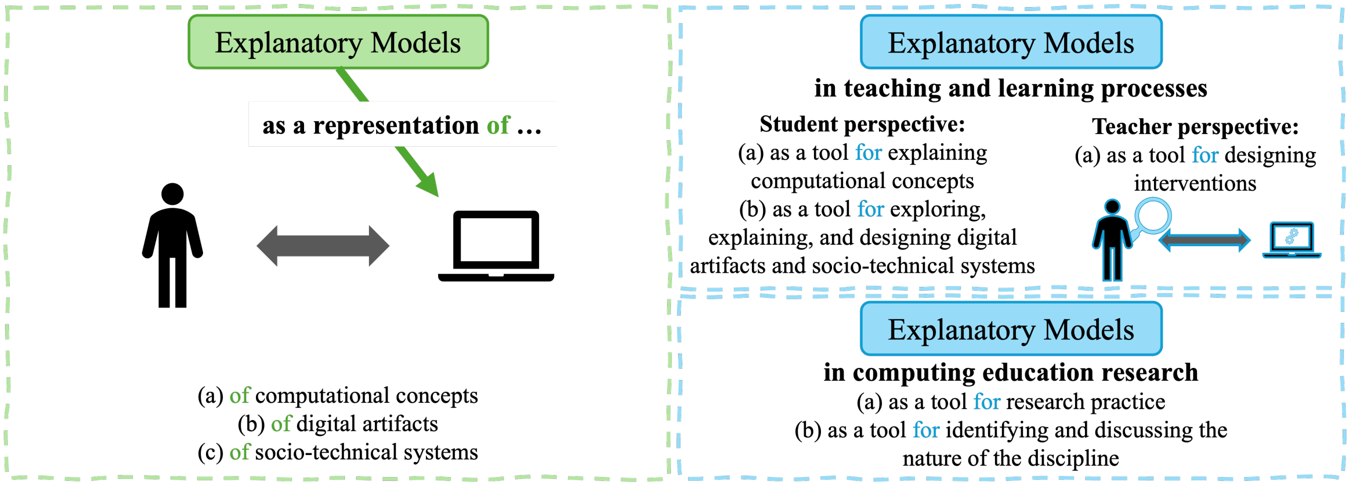
Explanatory Models
as a representation of ...
(a) of computational concepts
(b) of digital artifacts
(c) of socio-technical systems
Explanatory Models
in teaching and learning processes
Student perspective:
(a) as a tool for explaining computational concepts
(b) as a tool for exploring, explaining, and designing digital artifacts and socio-technical systems
Teacher perspective:
(a) as a tool for designing interventions
Explanatory Models
in computing education research
(a) as a tool for research practice
(b) as a tool for identifying and discussing the nature of the discipline

Figure 1: Overview of the perspectives of explanatory models based on the of-for-distinction discussed regarding the notion of models. Explanatory models (1) are representations of computational concepts, digital artifacts, or socio-technical systems while covering architecture and relevance perspectives, and (2) serve as tools for (a) teaching and learning processes and (b) research.

used in intervention studies could benefit communication about rationales and pedagogical ideas and, hence, cumulative knowledge building. For example, clarifying the explanatory model taught in a study's intervention could help researchers compare the results with other studies on the same explanatory model but with other pedagogical approaches.

Discussions on explanatory models are also related to discussing the nature of the CS discipline and computing education (e.g., caused by the fundamental changes regarding new technologies). The approach challenges the idea that we can teach and explain outcomes of the discipline based on reductions and by breaking them down to the smallest algorithmic details. We may need to rethink whether this approach holds for new computing topics. Practices of working with explanatory models could imply that computing education may need to focus more on scientific traditions in computing education [see 50], also in conveying a coherent and authentic image of the discipline.

4 Reflection and Conclusion

This paper introduced the explanatory model approach and characterized what constitutes an explanatory model, targeting challenges in computing education arising from rapid technological changes and paradigm shifts, particularly in areas like ML, where behaviors of digital artifacts cannot be fully understood by algorithms alone. Explanatory models serve as tools for understanding and explaining computational concepts and exploring, explaining, designing, reasoning about, and reflecting on specific digital artifacts. While analogies and abstractions are familiar in computing education as pedagogical vehicles, we propose that teaching explanatory models as an end goal in itself has unique benefits. We argue that systematically developing and making these models explicit can enhance both teaching and research in computing education (at least at school level). However, this approach also raises open questions that require further discussion and research. Below, we outline and explore some of them in more detail.

Model competencies and model thinking. This paper advocates for explicitly teaching explanatory models and communicating those used in computing education research (e.g., empirical studies). If we pursue this direction, students learn about and with explanatory models and use them as tools for different activities. This raises the question of whether we then need to teach model competencies and model thinking. Research in science education may provide valuable insights about teaching and learning models. In this context, it is criticized that models often are not adequately taught, such as the purposes of the models are often not sufficiently explained, which hinders students from reflecting on and critiquing models even if these are essential skills in working with models [e.g., 12, 19, 53]. Respective frameworks in science education addressing such skills of working with models could be considered, such as the framework for modeling competence [53] that comprises three levels: The first is about abilities to replicate or illustrate phenomena. The second includes abilities to use models in representation functions, that is, using models to describe, explain, and communicate about phenomena. The last adds functional perspectives of using models for something, such as a tool to derive predictions or knowledge about phenomena.

Further research is needed to explore to what extent such a framework from science education could be adopted to computing-specific explanatory models, taking into account the critical differences between computing and science education discussed earlier.

Methods and approaches for teaching and learning about models. Explicitly teaching explanatory models in schools also leads to the question of which methods and approaches are suitable and effective to teach models (e.g., based on research on notional machines) or can be beneficial to import and adapt from other disciplines. Research that explores computing-specific methods and approaches for effectively teaching students about explanatory models to support meta-cognitive processes and enable them to work with such models is needed. In this vein, the computing-specific functions for which students could use explanatory models should be explored. This may involve research on using explanatory models for (1) explaining, reconstructing, and reflecting on digital artifacts, and (2) modeling and designing digital artifacts. In other words, it would be fruitful to examine the use of explanatory models for different activities in computing education [see 26], such as described in approaches like computational empowerment [e.g., 14].

Development of explanatory models. Another question concerns thinking about where the explanatory models might come from. An intuitive approach would be to look for useful models from the CS discipline. However, explanatory models may be designed from an educational perspective, involving interpretative views in addition to objectively describing the inner workings of digital technologies. Thus, we believe that educational frameworks could be useful for designing explanatory models, like following the idea of educational reconstruction [see 16]. Nevertheless, related work is also done in the CS discipline, such as in the context of XAI, which involves developing techniques for explaining ML systems. Additionally, it could also involve other disciplines, such as social sciences, which also seek explanations for AI technologies [e.g., 8]. Similarly, Rahwan et al. [42] discuss using interdisciplinary approaches and methods from other disciplines to examine and explain AI systems behaviors.

Concluding remarks. Computer science is rapidly changing, leading to significant implications for computing education that we aim to address with the explanatory model approach. Traditional algorithmic systems may be explained in 'complete' detail with ground truth, but complex real-world systems and AI technologies challenge this clarity. Understanding and explaining such technologies is problematic in detail (e.g., see discussions around XAI). This highlights the need for explanatory models. In this vein, the presented approach is intended to encourage rethinking the nature of the computing education discipline, as past methods and approaches may not suffice for new technologies and concepts, especially thinking about challenges and paradigm shifts brought by AI. We propose focusing on explanatory models as concrete goals in computing education and a field in computing education research. While models are not new in computing education at all, they are often not treated as central learning content or even made explicit. We advocate for making explanatory models (a) explicit as teaching content and goals and (b) explicit in research, encompassing materials and tools, approaches, and empirical studies to clarify discussions on what is being measured and examined and understanding of various perspectives on computational concepts. As a consequence of developing explanatory models, it is necessary to decide which aspects should be covered by the explanatory models we teach (e.g., selecting aspects that are easy to comprehend for students [see 24]). According to the characterization of explanatory models, the question arises which purposes, aims, values, and norms should be included in these models, which requires a respective discourse in our discipline.

References

[1] Amina Adadi and Mohammed Berrada. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

[2] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Information Fusion* 58 (2020), 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

[3] Carlo Bellettini, Violetta Lonati, Mattia Monga, and Anna Morpurgo. 2024. To Be Or Not To Be ... An Algorithm: The Notion According to Students and Teachers. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 102–108. https://doi.org/10.1145/3626252.3630950

[4] Mordechai Ben-Ari and Tzippora Yeshno. 2006. Conceptual Models of Software Artifacts. *Interacting with Computers* 18, 6 (2006), 1336–1350. https://doi.org/10.1016/j.intcom.2006.03.005

[5] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. ACM, Virtual Event Canada, 610–623. https://doi.org/10.1145/3442188.3445922

[6] Ali Borji. 2023. Stochastic Parrots or Intelligent Systems? A Perspective on True Depth of Understanding in LLMs. https://doi.org/10.2139/ssrn.4507038

[7] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of Artificial General Intelligence: Early Experiments with GPT-4. https://doi.org/10.48550/ARXIV.2303.12712

[8] Jenna Burrell. 2016. How the Machine 'Thinks': Understanding Opacity in Machine Learning Algorithms. *Big Data & Society* 3, 1 (2016), 1–12. https://doi.org/10.1177/2053951715622512

[9] Lorena Casal-Otero, Alejandro Catala, Carmen Fernández-Morante, Maria Taboada, Beatriz Cebreiro, and Senén Barro. 2023. AI Literacy in K-12: A Systematic Literature Review. *International Journal of STEM Education* 10, 1 (2023), 29. https://doi.org/10.1186/s40594-023-00418-7

[10] Anne-Marie Cederqvist. 2020. Pupils' Ways of Understanding Programmed Technological Solutions When Analysing Structure and Function. *Education and Information Technologies* 25, 2 (2020), 1039–1065. https://doi.org/10.1007/s10639-019-10006-4

[11] John J. Clement. 2013. Roles for Explanatory Models and Analogies in Conceptual Change. In *International Handbook of Research on Conceptual Change* (2. ed ed.), Stella Vosniadou (Ed.). Routledge, New York, 414–446.

[12] Richard K. Coll, Bev France, and Ian Taylor. 2005. The Role of Models/and Analogies in Science Education: Implications from Research. *International Journal of Science Education* 27, 2 (2005), 183–198. https://doi.org/10.1080/0950069042000276712

[13] Jeroen de Ridder. 2006. Mechanistic Artefact Explanation. *Studies in History and Philosophy of Science Part A* 37, 1 (2006), 81–96. https://doi.org/10.1016/j.shpsa.2005.12.009

[14] Christian Dindler, Ole Sejer Iversen, Michael E. Caspersen, and Rachel Charlotte Smith. 2022. Computational Empowerment. In *Computational Thinking Education in K–12*, Siu-Cheung Kong and Harold Abelson (Eds.). The MIT Press, Cambridge, Massachusetts; London, England, 121–140. https://doi.org/10.7551/mitpress/13375.003.0009

[15] Benedict Du Boulay, Tim O'Shea, and John Monk. 1999. The Black Box inside the Glass Box: Presenting Computing Concepts to Novices. *International Journal of Human-Computer Studies* 51, 2 (1999), 265–277. https://doi.org/10.1006/ijhc.1981.0309

[16] Reinders Duit, Harald Gropengießer, Ulrich Kattmann, Michael Komorek, and Ilka Parchmann. 2012. The Model of Educational Reconstruction – a Framework for Improving Teaching and Learning Science1. In *Science Education Research and Practice in Europe*, Doris Jorde and Justin Dillon (Eds.). SensePublishers, Rotterdam, 13–37. https://doi.org/10.1007/978-94-6091-900-8_2

[17] Sally Fincher, Johan Jeuring, Craig S. Miller, Peter Donaldson, Benedict Du Boulay, Matthias Hauswirth, Arto Hellas, Felienne Hermans, Colleen Lewis, Andreas Mühling, Janice L. Pearce, and Andrew Petersen. 2020. Notional Machines in Computing Education: The Education of Attention. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*. ACM, Trondheim Norway, 21–50. https://doi.org/10.1145/3437800.3439202

[18] Ronald N. Giere. 2009. An Agent-Based Conception of Models and Scientific Representation. *Synthese* 172, 2 (2009), 269. https://doi.org/10.1007/s11229-009-9506-z

[19] Julia Gouvea and Cynthia Passmore. 2017. 'Models of' versus 'Models for': Toward an Agent-Based Conception of Modeling in the Science Classroom. *Science & Education* 26, 1 (2017), 49–63. https://doi.org/10.1007/s11191-017-9884-4

[20] Ileana Maria Greca and Marco Antonio Moreira. 2000. Mental Models, Conceptual Models, and Modelling. *International Journal of Science Education* 22, 1 (2000), 1–11. https://doi.org/10.1080/095006900289976

[21] Christiane Gresse Von Wangenheim, Jean C. R. Hauck, Fernando S. Pacheco, and Matheus F. Bertonceli Bueno. 2021. Visual Tools for Teaching Machine Learning in K-12: A Ten-Year Systematic Mapping. *Education and Information Technologies* 26, 5 (2021), 5733–5778. https://doi.org/10.1007/s10639-021-10570-8

[22] Shuchi Grover. 2024. Teaching AI to K-12 Learners: Lessons, Issues, and Guidance. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1.* ACM, Portland OR USA, 422–428. https://doi.org/10.1145/3626252.3630937

[23] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. 2019. XAI—Explainable Artificial Intelligence. *Science Robotics* 4, 37 (2019), eaay7120. https://doi.org/10.1126/scirobotics.aay7120

[24] Tom Hitron, Yoav Orlev, Iddo Wald, Ariel Shamir, Hadas Erel, and Oren Zuckerman. 2019. Can Children Understand Machine Learning Concepts?: The Effect of Uncovering Black Boxes. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.* ACM, Glasgow Scotland Uk, 1–11. https://doi.org/10.1145/3290605.3300645

[25] Lukas Höper and Carsten Schulte. 2024. Empowering Students for the Data-Driven World: A Qualitative Study of the Relevance of Learning about Data-Driven Technologies. *Informatics in Education* 23, 3 (2024), 593–624. https://doi.org/10.15388/infedu.2024.19

[26] Lukas Höper, Carsten Schulte, and Andreas Mühling. 2024. Learning an Explanatory Model of Data-Driven Technologies Can Lead to Empowered Behavior: A Mixed-Methods Study in K-12 Computing Education. In *Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 1.* ACM, Melbourne VIC Australia, 326–342. https://doi.org/10.1145/3632620.3671118

[27] Philip Nicholas Johnson-Laird. 1995. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness* (6. print ed.). Number 6 in Cognitive Science Series. Harvard Univ. Press, Cambridge, Mass.

[28] 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.* ACM, St. Gallen Switzerland, 1–10. https://doi.org/10.1145/3466725.3466728

[29] A. D. I. Kramer, J. E. Guillory, and J. T. Hancock. 2014. Experimental Evidence of Massive-Scale Emotional Contagion through Social Networks. *Proceedings of the National Academy of Sciences* 111, 24 (2014), 8788–8790. https://doi.org/10.1073/pnas.1412583111

[30] Peter Kroes. 1998. Technological Explanations: The Relation between Structure and Function of Technological Objects. *Society for Philosophy and Technology Quarterly Electronic Journal* 3, 3 (1998), 124–134. https://doi.org/10.5840/techne19983325

[31] Shalom Lappin. 2024. Assessing the Strengths and Weaknesses of Large Language Models. *Journal of Logic, Language and Information* 33, 1 (2024), 9–20. https://doi.org/10.1007/s10849-023-09409-x

[32] Zachary C. Lipton. 2018. The Mythos of Model Interpretability. *Commun. ACM* 61, 10 (2018), 36–43. https://doi.org/10.1145/3233231

[33] Violetta Lonati, Andrej Brodnik, Tim Bell, Andrew Paul Csizmadia, Liesbeth De Mol, Henry Hickman, Therese Keane, Claudio Mirolo, and Mattia Monga. 2022. What We Talk About When We Talk About Programs. In *ITiCSE 2022: Innovation and Technology in Computer Science Education.* ACM, Dublin Ireland, 117–164. https://doi.org/10.1145/3571785.3574125

[34] 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.* ACM, Honolulu HI USA, 1–16. https://doi.org/10.1145/3313831.3376727

[35] Bernd Mahr. 2009. Information Science and the Logic of Models. *Software & Systems Modeling* 8, 3 (2009), 365–383. https://doi.org/10.1007/s10270-009-0119-2

[36] Erik Marx, Thiemo Leonhardt, and Nadine Bergner. 2023. Secondary School Students' Mental Models and Attitudes Regarding Artificial Intelligence - A Scoping Review. *Computers and Education: Artificial Intelligence* 5 (2023), 100169. https://doi.org/10.1016/j.caeai.2023.100169

[37] Bhagya Munasinghe, Tim Bell, and Anthony Robins. 2023. Computational Thinking and Notional Machines: The Missing Link. *ACM Transactions on Computing Education* 23, 4 (2023), 1–27. https://doi.org/10.1145/3627829

[38] Donald A. Norman. 1983. Some Observations on Mental Models. In *Mental Models* (1 ed.), Dedre Gentner and Albert L. Stevens (Eds.). Psychology Press, New York, 7–14.

[39] Jonathan Osborne. 2014. Teaching Scientific Practices: Meeting the Challenge of Change. *Journal of Science Teacher Education* 25, 2 (2014), 177–196. https://doi.org/10.1007/s10972-014-9384-1

[40] Cynthia Passmore, Julia Svoboda Gouvea, and Ronald Giere. 2014. Models in Science and in Learning Science: Focusing Scientific Practice on Sense-making. In *International Handbook of Research in History, Philosophy and Science Teaching*, Michael R. Matthews (Ed.). Springer Netherlands, Dordrecht, 1171–1202. https://doi.org/10.1007/978-94-007-7654-8_36

[41] Simon Portegies Zwart. 2018. Computational Astrophysics for the Future. *Science* 361, 6406 (2018), 979–980. https://doi.org/10.1126/science.aau3206

[42] Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W. Crandall, Nicholas A. Christakis, Iain D. Couzin, Matthew O. Jackson, Nicholas R. Jennings, Ece Kamar, Isabel M. Kloumann, Hugo Larochelle, David Lazer, Richard McElreath, Alan Mislove, David C. Parkes, Alex 'Sandy' Pentland, Margaret E. Roberts, Azim Shariff, Joshua B. Tenenbaum, and Michael Wellman. 2019. Machine Behaviour. *Nature* 568, 7753 (2019), 477–486. https://doi.org/10.1038/s41586-019-1138-y

[43] Saman Rizvi, Jane Waite, and Sue Sentence. 2023. Artificial Intelligence Teaching and Learning in K-12 from 2019 to 2022: A Systematic Literature Review. *Computers and Education: Artificial Intelligence* 4 (2023), 100145. https://doi.org/10.1016/j.caeai.2023.100145

[44] Carsten Schulte. 2008. Block Model: An Educational Model of Program Comprehension as a Tool for a Scholarly Approach to Teaching. In *Proceeding of the Fourth International Workshop on Computing Education Research - ICER '08.* ACM Press, Sydney, Australia, 149–160. https://doi.org/10.1145/1404520.1404535

[45] Carsten Schulte and Lea Budde. 2018. A Framework for Computing Education: Hybrid Interaction System: The Need for a Bigger Picture in Computing Education. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research.* ACM, Koli Finland, 1–10. https://doi.org/10.1145/3279720.3279733

[46] D Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. 2015. Hidden Technical Debt in Machine Learning Systems. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc., Montreal Canada, 2503–2511.

[47] Sue Sentance and Jane Waite. 2022. Perspectives on AI and Data Science Education. In *AI, Data Science, and Young People. Understanding Computing Education (Vol 3).* Raspberry Pi Foundation, Cambridge, UK, 1–9.

[48] Juha Sorva. 2013. Notional Machines and Introductory Programming Education. *ACM Transactions on Computing Education* 13, 2 (2013), 1–31. https://doi.org/10.1145/2483710.2483713

[49] Matti Tedre, Peter Denning, and Tapani Toivonen. 2021. CT 2.0. In *21st Koli Calling International Conference on Computing Education Research.* ACM, Joensuu Finland, 1–8. https://doi.org/10.1145/3488042.3488053

[50] Matti Tedre and Erkki Sutinen. 2008. Three Traditions of Computing: What Educators Should Know. *Computer Science Education* 18, 3 (2008), 153–170. https://doi.org/10.1080/08993400802332332

[51] Matti Tedre, Tapani Toivonen, Juho Kahila, Henriikka Vartiainen, Teemu Valtonen, Ilkka Jormanainen, and Arnold Pears. 2021. Teaching Machine Learning in K–12 Classroom: Pedagogical and Technological Trajectories for Artificial Intelligence Education. *IEEE Access* 9 (2021), 110558–110572. https://doi.org/10.1109/ACCESS.2021.3097962

[52] Zeynep Tufekci. 2015. Algorithmic Harms beyond Facebook and Google: Emergent Challenges of Computational Agency. *Colorado Technology Law Journal* 13, 2 (2015), 203–218.

[53] Annette Upmeier Zu Belzen, Jan Van Driel, and Dirk Krüger. 2019. Introducing a Framework for Modeling Competence. In *Towards a Competence-Based View on Models and Modeling in Science Education*, Annette Upmeier Zu Belzen, Dirk Krüger, and Jan Van Driel (Eds.). Vol. 12. Springer International Publishing, Cham, 3–19. https://doi.org/10.1007/978-3-030-30255-9_1

[54] Pieter E. Vermaas and Wybo Houkes. 2006. Technical Functions: A Drawbridge between the Intentional and Structural Natures of Technical Artefacts. *Studies in History and Philosophy of Science Part A* 37, 1 (2006), 5–18. https://doi.org/10.1016/j.shpsa.2005.12.002

[55] Abigail Zimmermann-Niefield, Makenna Turner, Bridget Murphy, Shaun K. Kane, and R. Benjamin Shapiro. 2019. Youth Learning Machine Learning through Building Models of Athletic Moves. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children.* ACM, Boise ID USA, 121–132. https://doi.org/10.1145/3311927.3323139