WILEY

Journal of Computer Assisted Learning

ORIGINAL ARTICLE OPEN ACCESS

From Imitation Games to Robot-Teachers: A Review and Discussion of the Role of LLMs in Computing Education

Tobias Kohn

Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Correspondence: Tobias Kohn (tobias.kohn@kit.edu)

Received: 25 March 2024 | **Revised:** 6 March 2025 | **Accepted:** 31 March 2025

Keywords: ChatGPT | education | LLM

ABSTRACT

Background: The recent advent of powerful, exam-passing large language models (LLMs) in public awareness has led to concerns over students cheating, but has also given rise to calls for including or even focusing education on LLMs. There is a perceived urgency to react immediately, as well as claims that AI-based reforms of education will lead to a broadening of accessibility to high-quality education.

Objectives: We review and discuss three major themes that appear in the research literature on LLMs and computing education, namely that (i) LLMs exhibit human-like performance and can pass exams, (ii) LLMs are freely available and intuitive to use, and (iii) students use LLMs to cheat or accept the results without critical evaluation. Moreover, we highlight the importance of a more human-centric view on the topic.

Methods: The discussion is based on a review of the (research) literature in the fields related to computing education, picks up claims and statements from the literature, and compares them with research findings from the area. By making some of the rather tacit premises more explicit and putting them into context, we aim to base the discourse about AI in education on more solid grounds.

Results and Conclusion: We find that claims such as the broadening of accessibility to high-quality education or calls for urgent educational reforms are not supported by evidence. Furthermore, we argue that there is a central human element in education that cannot be automated or replaced by AI tools.

1 | Introduction

The arrival of large language models (LLMs) in the public consciousness has brought AI to the fore of public discourse. ChatGPT was immediately hailed as a game-changer with its capability to react to textual input with stunningly coherent answers (Becker et al. 2023; Bull and Kharrufa 2023; Harrer 2023; Herbold et al. 2023; Rajabi et al. 2023; Sallam 2023). LLMs were envisioned to both do and grade homework, give feedback, solve problems and make everyone a 'programmer'; but also to replace (white-collar) jobs and open up new opportunities for cheating (Bull and Kharrufa 2023; Christian 2023; Denny, Kumar, et al. 2023; Finnie-Ansley et al. 2022, 2023; Ibrahim et al. 2023; Savelka, Agarwal, Bogart, Song, et al. 2023; Savelka, Agarwal, An, et al. 2023; Vallance 2023). The one thing everyone seems to equivocally agree on is that LLMs will have a lasting impact on our world.

Soon after their arrival, LLMs have been shown to solve a wide variety of problems and achieve performances that would allow these tools to pass exams or excel at coding competitions, say (Prather, Denny, et al. 2023; Li et al. 2022). Evidently, this means that students might use LLMs to successfully cheat during assignments and exams. On the flip side, it puts into question the very methods of teaching and assessment; what value does an education or a test have, if it can be passed by a machine? Does it still make sense to teach the same contents

Summary

- What is currently known about this topic?
    - Large language models (LLMs) such as ChatGPT have had a huge impact on the public discourse and led to calls for immediate reaction and a revolution in education. Major concerns are new possibilities for students to cheat and how these LLMs will change professional work.
- What does this paper add?
    - We review and discuss three core themes in the research literature: the human-like performance of LLMs, their free availability, and whether dishonest use of LLMs is as big an issue as often suggested.
- Implications for practice/or policy
    - While the arrival of (generative) AI should certainly be considered in education, we must be careful not to blindly embrace it at the cost of our core values.

or should we rather instruct students on how to use LLMs instead? Conversely, do we have to make the questions 'harder' in the sense that LLMs should no longer reliably find the correct answers?

From the perspective of a teacher or an examiner, we are suddenly thrown into a real-life Turing test. We need to discern what is original work by a student and what might have been generated by AI systems. What was originally a thought experiment has suddenly become a reality: the notion that an artificial system can imitate a student well enough to fool the examiner.

Although the most famous in public discourse, ChatGPT is not the first LLM that has been released to the public recently. Codex and Copilot have already been the subject of research for some time, although generally with a clearer focus on the productivity of professional programmers or code analysis (Barke et al. 2023; Ernst and Bavota 2022; Finnie-Ansley et al. 2022; Sarsa et al. 2022; Vaithilingam et al. 2022). Still, research and debate in this area are just taking off with many yet unknowns and often a mixture of scientific facts, marketing, and folklore clouding the picture and debate.

Yet, one of the key questions has already emerged quite clearly: *how do we have to adapt our educational systems and practices in light of the new generative AI*? As outlined above, the arguments brought forward seem overwhelming, ranging from concerns over students' cheating to visions of completely novel skill sets needed to succeed in future careers. In fact, during a recent literature review (Prather, Denny, et al. (2023)) we noticed a small number of recurring themes that were used to motivate specific lines of research in this area or call for immediate action in reforming education:

- LLMs exhibit 'human-level' or 'human-like performance' and can pass scholarly tests and exams;
- LLMs are free of charge, easy to use and provide an accessible, convenient and ubiquitous 'natural-language' interface;
- LLMs allow students to cheat and/or cause them to become over-reliant on LLMs.

Closer inspection reveals that few papers actually discuss what they mean by 'human-like performance', leaving it as a vague but highly suggestive term. Moreover, neither the hailed accessibility nor the concerns over students' dishonesty really hold up to currently available evidence (cf. Sections 2.4 and 3.4). We therefore argue that we should not be too hasty or overzealous in our conclusions that education should be revolutionised in light of generative AI.

Moreover, an aspect that finds little attention in the discourse so far is the 'human element' in education. There seems to be general excitement about providing students with new educational tools that can provide customised explanations, assessments and support. In effect, however, this means replacing human tutors and teachers with machines. We question whether such a replacement is always appropriate and caution against a removal of the human element in education, particularly without due discussion and deliberation.

In four sections, we will try to investigate the above claims in the current debate (which is pretty much a moving target) and argue that many positions brought forward should be scrutinised more critically. In particular, in Section 2 we look at how the arrival of LLMs affects, in turn, students, teachers and professionals. In Section 3 we look at the skills required to effectively use LLMs and debunk the notion of a freely and intuitively accessible system. In Section 4 we focus on the issue of assessment before we aim to put it all into a larger context in Section 5. Perhaps the key message of the entire article is that we must not allow the 'human element' to be overlooked in the current debate. Education is not about the technology, but about the learner and the society we are shaping through education.

1.1 | Methodology

The sudden nascence of LLMs on the global stage and their immense potential impact on science, education and society means that, first, the most relevant literature is very recent and, second, new publications on the topic appear at a very fast pace. We therefore opted against a systematic literature review, but instead relied on a survey based on snowballing, published by an ITiCSE working group (Prather, Denny, et al. 2023), as our basis. We then complemented this initial list of around 70 papers by including 24 articles published at the ITiCSE 2023 conference and the SIGCSE TS 2024 as representatives of newer articles in the field that could not have been considered by the ITiCSE 2023 working group. We searched for the keywords 'GPT', 'LLM' and 'Generative AI' to find potentially relevant publications from these venues. The articles published at these later venues are noticeably more 'mature' in the sense that these articles spend less time on motivating the need for a discussion or assessment of generative AI, LLM or GPT models (similarly, Denny, Leinonen, et al. (2024) note that the discussion has shifted from assessing capabilities of LLMs to using them in classroom).

1.2 | Position and Limitations

As outlined above, we are critical of the narrative that LLMs are 'human-like' and highly accessible systems that could and should revolutionise education. We do not, however, question the performance LLMs have shown in various tasks and benchmarks, but rather whether descriptions as 'human-like' are conducive to a scholarly debate *without proper discussion* of what is meant by it. Likewise, it is evident that LLMs and ChatGPT in particular enjoy a huge number of users; we caution, however, against concluding that *everyone* has *equal access* to such systems and finds it equally intuitive to use *productively*.

In particular, we believe that calls to immediately integrate generative AI into education or to question the value of current education are premature. On the premise that education at its core is not about preparing students for the future, but to foster the personal growth of the students as human beings (see Section 5.1), we argue that the impact of new technologies such as LLMs on the core of education is often overestimated.

As a review with a targeted focus on a few selected themes, this paper does not aim to provide an overview of the entire research literature, but rather tries to highlight how some of the premises and conclusions surrounding the body of research deserve more scrutiny and in-depth discussion. Given how academic research and corporate interests are strongly intertwined, we must be mindful of how we frame our research in the public discourse and make sure that we critically assess our underlying assumptions and positions.

2 | The Arrival of Large Language Models

At least as far as computing education is concerned, LLMs have arrived in two waves. The first wave of research and exploration was caused by the introduction of Codex and Copilot in 2021 as models trained specifically on code with the aim of boosting programmer productivity (Barke et al. 2023; Becker et al. 2023; Bird et al. 2022; Ernst and Bavota 2022; Vaithilingam et al. 2022). While the release of these programming-oriented models was relatively contained to the computing community, the second wave, with the release of ChatGPT in late 2022, caused a much more widespread frenzy, leading to a sudden public awareness of LLMs (Fletcher and Nielsen 2024; Joshi et al. 2024; Vallance 2022).

Particularly from experimenting with ChatGPT, it was quickly found that LLMs could do your homework, pass exams, grade students and even solve problems (Becker et al. 2023; Finnie-Ansley et al. 2022). A crucial aspect that has been reported very frequently is the free and easy access to ChatGPT, making it readily available for everyone. Furthermore, the impressive natural-language output is usually described as achieving 'human-like performance' (Prather, Denny, et al. 2023).

In this section, we will look at the literature and debate from three different perspectives, focusing on the impact and ramifications of LLMs on (i) students and education in general, (ii) teachers and educators, and (iii) professional software developers; we acknowledge, however, that none of these three perspectives can be entirely isolated from the other two, leading to some thematic overlaps.

2.1 | The Immediate Revolution in Education

In light of ChatGPT's impressive performance, there seems to be general agreement that "changes to the educational system are inevitable" (Malinka et al. 2023) and that "it is already clear that artificial intelligence has the potential to revolutionise the way we teach and learn" (Rudolph et al. 2023). Moreover, these changes call for an immediate reaction (Malinka et al. 2023) before it is "too late for educational institutions" (Rudolph et al. 2023). Likewise, "With the escalating influence of LLMs, there is an urgent need to reshape the education of future software engineers" (Kirova et al. 2024). Although usually less dramatic, these sentiments are often shared in one form or another: "curricular changes needed to accommodate the new reality" (Savelka, Agarwal, An, et al. 2023), "an existential threat to the teaching and learning of introductory programming" (Finnie-Ansley et al. 2022), "It seems clear that the contemporary rise of AI tools has provoked some anxiety within the education community" (French et al. 2023) and "we envision that LLMs could have a substantial impact on the education sector" (Nunes et al. 2023). Finally, Becker et al. (2023) state that "Our view is that these tools stand to change how programming is taught and learned – potentially significantly – in the near-term, and that they present multiple opportunities and challenges that warrant immediate discussion as we adapt to the use of these tools proliferating", calling for that "we urgently need to review our educational practices in light of these new technologies".

The urgency of a necessary reaction was also reported by one of the first studies probing educators' views after the release of ChatGPT: "We found that in the short-term many planned to take immediate measures to discourage cheating" (Lau and Guo 2023). In this case, the immediacy is clearly connected to the fear that students might cheat, where Lau and Guo (2023) also note that "One specific concern on the forefront of many educators' minds is the fact that these tools can effectively solve homework assignments and exam problems across a wide variety of school subjects".

The cause for this widespread concern on the impact of LLMs on education can mostly be found in the ability of LLMs to solve assignments, write essays and even pass exams (Becker et al. 2023; Denny, Prather, et al. 2024; Finnie-Ansley et al. 2022; Herbold et al. 2023; Kiesler and Schiffner 2023; Mahon et al. 2023; Nunes et al. 2023; Prather, Denny, et al. 2023; Sarsa, Denny, et al. 2022; Yeadon et al. 2023), putting into question the sense of homework assignments, but also raising concerns over cheating and dishonesty. The result has been a large number of studies trying to assess the capabilities of LLMs for solving problems (Prather, Denny, et al. 2023), often with reference to the issue of cheating. In fact, while most papers report concerns about cheating and dishonesty, French et al. (2023) explicitly point out that "students might also choose to take responsibility for their actions". Equally, Zastudil et al. (2023) cite an interview participant suggesting that a "student who was never motivated to cheat in the first place will not do it now just because ChatGPT is available", which is also supported by Rogers et al. (2024)'s findings.

The perceived immediacy of revolution and thus urgency to react in education may be contrasted with a more cautious stance in medicine and healthcare. As Harrer (2023) points out:

"The often-heralded Silicon Valley paradigm of 'move fast and break things' does not apply to healthcare and medicine". With human lives at stake, errors weigh much heavier in the field of medicine and Harrer (2023) stresses the importance of trust (between doctor and patient), which may be quickly and irrevocably lost through faulty AI systems.

We might wonder, though, whether education, too, is—or should be—a field where trust is highly valued and considered a crucial good to be protected. The relationship between teacher and student is to a very large degree based on trust, where students need some authority to help them organise facts and learning (Kirschner and van Merriënboer 2013). In fact, Whitehead (1959) makes the point that students have to accept the authority of their teacher as a basis to trust in the validity and usefulness of the ideas presented before they can then go on and scrutinise these ideas for themselves. For instance, students cannot effectively assess the trustworthiness of an online resource on their own, nor effectively learn from experimenting without guidance (Kirschner et al. 2006; Kirschner and van Merriënboer 2013).

The discussion regarding trust in the context of AI is primarily concerned with the trust that students (or teachers) place in AI systems: "Trust is one of the main factors affecting the interaction between AI and its users" (Amoozadeh et al. 2024). Amoozadeh et al. (2024) found about half of the study participants expressing a lack of trust, while Zastudil et al. (2023) report that most students as well as most teachers in their study cited a lack of trustworthiness as a major concern. Tossell et al. (2024), too, report that students cited concerns about the trustworthiness of LLMs. Indeed, properly utilising LLMs does not necessarily come naturally, but requires some basic skills (Section 3). In particular, the output generated by LLMs is unreliable and requires critical evaluation (cf. Alves and Cipriano (2023); Dakhel et al. (2023); Harrer (2023)). In spite of impressive benchmark results, it is virtually impossible to prove the trustworthiness of LLMs due to their probabilistic nature. Even established and commercial AI systems may endanger children because of the AI's lack of understanding (BBC 2021a). Philbin (2023) warns that the students' belief that "the AI is always correct" may lead to "adverse learning outcomes".

It is important, though, to understand the difference between 'trust in a machine', which translates to a confidence that it works properly (we will come back to this issue in Section 3.2), and 'trust in a person', which is a complex social relationship. Placing your trust in a teacher is more akin to the assumption that the teacher will work towards your growth, success and prosperity. We would therefore argue that trust plays as crucial a role in education as in healthcare and that questions of human trust should be at the fore of our considerations.

Returning to the urgency of revolutionising education, we might note that similar arguments have been brought forward before. Actually, in the context of 'digital natives', for instance, Kirschner and van Merriënboer (2013) bring Cohen's concept of 'moral panic' to our attention, which we believe equally applies in the context of generative AI: "moral panic occurs when a 'condition, episode, person or group of persons emerge to become defined as a threat to social values and interests'" (Cohen 1973; Kirschner and van Merriënboer 2013). Moreover:

> Arguments are often couched in dramatic language, proclaim a profound change in the world, and pronounce stark generational differences...
>
> Such claims coupled with appeals to common sense and recognizable anecdotes are used to declare an emergency situation, and call for urgent and fundamental change. [...]
>
> Thus, the language of moral panic and the divides established by commentators serve to close down debate, and in doing so allow unevidenced claims to proliferate.

(Cohen 1973, 782–783)

As shown at the beginning of this section, we do indeed find the claims that (generative) AI and LLMs are profoundly changing our world and that we should urgently reform education and react immediately to these new challenges and opportunities (cf. also Balse et al. (2023)). For instance, Denny, Prather, et al. (2024) remark that "Educators need to adapt quickly" and "In short, we must adopt or perish". Some go even further and declare a fundamental reform of computer science itself (Welsh 2022). A frequently cited paper on early signs of 'artificial general intelligence' (Bubeck et al. 2023) readily admits that the results claimed in the paper are not reproducible (in fact, some claims have been experimentally refuted (Dean 2023)), which lacks in the fundamental principles of scientific methodology and discourse (Marcus 2023). Hence, Cohen's concept of 'moral panic' seems a surprisingly apt description of the current debate.

We feel that as a community we have to push back on this elements of 'moral panic' and strife for a more balanced discussion where we seek out reliable and durable evidence for the actual impact, educational necessity and benefits of generative AI (cf. Fletcher and Nielsen (2024)). In this, we should perhaps follow the example of healthcare to a more measured and contemplated approach.

2.2 | A Welcome Teaching Tool

Saving valuable time or human effort has been a trope in the field of technology since the dawn of public debate. It is not surprising that we find high hopes placed in LLMs to do just that—particularly on the part of the teacher or expert and usually combined with the idea of freeing up valuable resources for more meaningful tasks.

There are two (related) major themes that emerge from the literature. On the one hand, there is the problem of scaling education with an emphasis on scarce (human) teaching resources, particularly in light of growing student numbers (Al-Hossami et al. 2023, 2024; Denny, Sarsa, et al. 2022; Leinonen, Denny, et al. 2023; Liffiton et al. 2023; Liu and M'hiri 2024; Phung et al. 2023; Taylor et al. 2024; Zhang et al. 2022). On the other hand, teachers and experts seem to be bogged down with irrelevant problems, keeping them from utilising their skills more effectively, as noted by, e.g., Ahmed et al. (2022): "time that might otherwise be spent on useful pedagogy on problem-solving and logic is spent helping novices deal with such [minor] errors." This is also mirrored in healthcare: "LLMs show promise to change clinical practice by allowing doctors to spend more time with their patients." (Harrer 2023). In terms of professional software developers, this is usually framed as 'increased productivity'. In a very similar vein, we find that the production of 'high-quality' teaching materials is a time-consuming, laborious and costly task and hence both difficult to scale and taking away important teacher's time (Becker et al. 2023; Ishizue et al. 2024; Liffiton et al. 2023; Lu et al. 2023).

Education at scale is, of course, not a new phenomenon as evidenced by, e.g., the numerous MOOCs and online academies, which essentially continue the ideas of television-broadcasted educational programmes from the 20th century. In order to scale education, it must be automated to a very high degree, moving away from the classic notion of teacher-student relationships. The promise that AI brings to this endeavour is to make it truly interactive and personalised. Students can now generate new exercises, model solutions and explanations, but also get timely and useful feedback on their work (Becker et al. 2023; Dai et al. 2023; Jalil et al. 2023; MacNeil et al. 2023; Ouh et al. 2023; Phung et al. 2023; Prather, Denny, et al. 2023; Sarsa et al. 2022; Wermelinger 2023). On the flip side, Sarsa et al. (2022) note that automated assessment also has an impact on teaching in that it favours a multitude of small programmes that are amenable to automated assessment, grading and feedback generation. Thus the tools we use in order to scale education have reciprocally a direct influence on what and how we teach.

A consequence of the focus on scaling education is the necessary devaluation of teachers and seeing their role more in managing students' learning process. This is not only due to the issue of scaling, of course, but also coincides with other factors like the proliferation of self-directed learning such as Papert's constructionism. Kirschner and van Merriënboer (2013) observed the "demotion of teachers from someone whose job it was [...] to teach [...] to someone whose role is standing on the sidelines and guiding and/or coaxing a breed of self-educators". Moreover, they argue that students are not necessarily good or successful in controlling their own learning, "especially in computer-based learning environments"—something that has, unfortunately, only become too apparent during the recent pandemic.

So, while the literature motivates the use of LLMs for teaching by freeing teachers from mundane tasks and helping them focus on what is pedagogically more meaningful, we have to concede that the very notion of scaling education is at odds with this 'helping the teacher' sentiment—by scaling education the actual teacher is replaced by a mere course administrator. Moreover, we should equally be aware of the strong values induced by the very notion of freeing up teachers' time from ordinary or minor problems of learning to more substantial and important tasks. This is not to say that the use of LLMs or generative AI is necessarily problematic or not helpful for the purpose of teaching, but the picture is clearly more complex than how it is commonly drawn. The use of technology such as LLMs as well as the idea of education at scale comes with stark moral values and implications on what education actually is.

Finally, the notion of freeing up rote and meaningless human 'cognitive' labour is a cornerstone and defining characteristic of the field of artificial intelligence (and computing in general). Blackwell (2023), however, warns us that "if the purpose of an AI system is really to economically replicate or automate human actions, we need to ask when and why this is an appropriate thing to do". In other words, even if LLMs allow us to replace teachers and automate teaching tasks, we have to be careful and consider whether this results in a meaningful benefit for students and their learning process.

2.3 | Think and Act Like Professionals

In professional settings, generative AI is perceived both as a threat to job security and prospects, but also as a useful tool to boost productivity. The issue of employment prospects is explicitly raised by French et al. (2023): "An additional reason for advocating a critical approach to the software was to reduce students' fears around AI replacing them in the future workplace, a common anxiety that emerged during informal discussions." The flip side of this coin is the notion that students should learn to make effective use of LLMs as this skill will likely be required at future workplaces (Bull and Kharrufa 2023; Fernandez and Cornell 2024; Kirova et al. 2024; Malinka et al. 2023; Rogers et al. 2024). The two sides are also echoed by students, who "expressed mixed views on how GenAI tools might affect their future career prospects. While some believed job opportunities would decrease, others were optimistic that these tools would improve their productivity and give rise to new careers" (Prather, Denny, et al. 2023).

Software engineers who were asked about their experience with LLMs "agree that there was no need to train developers specifically on the use of these tools", but emphasise that "people should have a grasp of programming fundamentals" (Bull and Kharrufa 2023). Indeed, the performance gap between novices and experienced programmers when using LLMs has been confirmed empirically (Nam et al. 2023), suggesting that a solid foundation in programming is not only a prerequisite for effective use of LLMs, but also the best preparation. This is also in line with findings regarding internet search behaviour: "Students with more prior knowledge have an advantage because they can easily link their prior knowledge [...] to information found on the web" (Kirschner and van Merriënboer 2013).

With their ability to suggest and complete program code, it is obvious that LLMs extend already existing code completion and augmentation technologies used in modern development environments (IDEs). Research has therefore looked into how LLMs are best integrated into existing development environments and workflows, and to what extent LLMs increase productivity (Bull and Kharrufa 2023; Nam et al. 2023; Vaithilingam et al. 2022).

In order to inform training and education, Bull and Kharrufa (2023) have interviewed five professional software developers and found that LLMs have "the potential to improve productivity and capacity". The participants reported that LLMs help automate the "boring stuff", speed up coding or simply provide ideas or a first draft as a starting point. In contrast, Vaithilingam et al. (2022) looked at the actual performance gains and found that although the participants reported a feeling of increased productivity, this was not necessarily backed up by numbers. The positive feeling towards LLMs and their usefulness is also reported by a different study, together with a significantly increased task completion rate (Nam et al. 2023). Finally, even novice programmers can benefit from using LLMs, although not as much as more experienced programmers (Kazemitabaar et al. 2023). There are, however, concerns regarding the students' abilities to critically assess the output of LLMs (Cipriano and Alves 2023; Dakhel et al. 2023; Finnie-Ansley et al. 2022; Prather, Reeves, et al. 2023; Sandoval et al. 2023).

2.4 | LLMs: For Adults Only

A noteworthy theme that emerges from the literature and general discussion on LLMs in education is the strong distrust of how students might use this technology. For instance: "Having Codex in the hands of students should warrant concern similar to having a power tool in the hands of an amateur" (Finnie-Ansley et al. 2022) or "[Copilot] can also become a liability if it is used by novices" (Dakhel et al. 2023). Hoq et al. (2024) observe that "While these tools might help professional programmers develop code more efficiently and can be used by instructors to create educational resources, programming educators have raised concerns around potential student over-reliance on these models". Indeed, the debate so far suggests that there are rather clear usage scenarios for the different types of users:

- Professionals use LLMs for increased productivity;
- Teachers use LLMs to reduce the workload;
- Students use LLMs unreflected or to cheat.

As a direct consequence of this sentiment there is a tension between 'banning LLMs to avoid cheating' and 'teaching the use of LLMs to prepare students for their careers'—a tension that has been explicitly picked up by the community in titles such as, e.g., "From 'ban it until we understand it' to 'resistance is futile'" (Lau and Guo 2023) or "RenAIssance or ApocAIypse?" (Denny, Becker, et al. 2023).

Moreover, it is striking how the discourse commonly frames the students using LLMs on their own initiative as a problem, whereas the same technology is hailed as a great tool in the hands of teachers and professionals. As French et al. (2023) and Zastudil et al. (2023) have pointed out, students themselves might actually be more worried about their learning than eager to cheat their way through school. This has been corroborated by Lee et al. (2024), who found that cheating behaviour remained relatively stable and by Rogers et al. (2024), who report that the majority of students are using LLMs "in ways that we as educators would appreciate", but also note that the unreliability of LLMs' output may act as a deterrence. Furthermore, there have already been anecdotes of teachers and researchers (i.e., not students) using LLMs in dishonest or inappropriate ways (cf. Hoover (2023); Klee (2023)). Hence, the picture of students cheating and 'adults' using LLMs responsibly has so far little evidence in reality.

In fact, students' motivation for using an LLM to complete an assignment might not differ that much from teachers and professionals. During interviews conducted by Zastudil et al. (2023), students indicated that they tend to use LLMs to avoid "busy work" or "meaningless" assignments. This differs little from the professionals using an LLM to automate the "boring stuff" (Bull and Kharrufa 2023) or the hope to free up teachers' time for meaningful tasks (cf. Section 2.2).

In addition to concerns about students' dishonesty, there are also concerns about students becoming over-reliant (Becker et al. 2023; Denny, Khosravi, et al. 2023; Fernandez and Cornell 2024; Finnie-Ansley et al. 2022; Kazemitabaar et al. 2023; Lau and Guo 2023; Reeves et al. 2023). or lacking the ability to critically reflect and understand the output generated by LLMs (Section 3.3). However, the concern of students using LLMs to cheat clearly permeates the literature (Becker et al. 2023; Bull and Kharrufa 2023; Cipriano and Alves 2023; Denny et al. 2022; Denny, Khosravi, et al. 2023; Denny, Leinonen, et al. 2023; Denny, Leinonen, et al. 2024; Dobslaw and Bergh 2023; Finnie-Ansley et al. 2022; Idialu et al. 2017; Jalil et al. 2023; Kazemitabaar et al. 2023; Kiesler and Schiffner 2023; Lau and Guo 2023; Nguyen and Allan 2024; Orenstrakh et al. 2023; Ouh et al. 2023; Philbin 2023; Prasad and Sane 2024; Prather, Denny, et al. 2023; Puryear and Sprint 2022; Rajabi et al. 2023; Savelka, Agarwal, Bogart, Song, et al. 2023; Savelka, Agarwal, Bogart, and Sakr 2023; Sheard et al. 2024; Wang et al. 2023; Wermelinger 2023).

3 | The Skills Required

The idea of interacting with a computer using natural language is at least as old as computers themselves (Turing 1950; Weizenbaum 1966; Dijkstra 1979; Guzdial 2015)—an idea that LLMs seem to finally fulfil. However, in spite of popular claims regarding 'natural conversations' or the LLMs' 'understanding of intent', there are issues regarding both input and output. Harrer (2023), for instance, emphasises that LLMs "should be treated as imperfect tools which [...] need strict human supervision and action at both operational interfaces, input and output". Indeed, as it turns out, there is skill involved in crafting a prompt for the input and the output in form of generated text must be critically evaluated.

3.1 | Prompt Engineering

The exact prompt (input) to an LLM is essential for the generation of high-quality output: "the AI code generator produces high quality results, but this depends on the quality of the prompt message" (Kazemitabaar et al. 2023). Even small details such as putting a colon at the end of a prompt can make a difference (Wermelinger 2023) and he notes that "Copilot most often does not understand our instructions to fix or improve the code it generates unless we formulate them in a very specific way". In contrast Reeves et al. (2023) found that variations in prompts did not make as much of a difference as expected. In any case, the task of ‘engineering a good prompt’ has certainly received some widespread attention.

The mere notion of ‘prompt engineering’ is already suggestive of the fact that LLMs do not really understand the prompt, nor do they provide a truly ‘natural’ interface. Jiang et al. (2022) argue that even LLMs have syntax and semantics that must be understood by the user. They highlight “a sense of needing to learn the system's ‘syntax’, despite model input consisting of natural language” and “challenges in forming an accurate mental model of the types of requests the model can reliably translate to code”. Moreover, Sarkar (2023) suggests that “the disappearance of formality may be an illusion; generative models still require high levels of craft expertise to use effectively, and the shift to ‘prompt engineering’ hasn't eliminated programming at all, but simply shifted it into a higher level of abstraction”.

Harel and Marron (2024) raise the question about the significance of knowing whether we converse with a human or a machine and point out that people might have different expectations when dealing with a human or a machine.

From a historical perspective, it is interesting to observe how the natural text interface is hailed as the future and expected to supplant programming, say. In contrast, Dijkstra (1979) pointed out that we should not consider formal symbols a burden, but rather a privilege. Indeed, the evolution of mathematical notation has arguably been a major driver for mathematical and scientific discoveries. Furthermore, Arawjo (2020) shows how programming notation has evolved from drawn diagrams to the modern text-based form, which is already—to a certain extent—a reversal of the long evolution of mathematical notation to fit a modern medium (i.e., the typewriter). Nonetheless, what certainly has changed since the publication of Dijkstra's article in the 1970s is the level of abstraction: it is not programming as a low-level algorithmic description that will necessarily be replaced by LLMs, but rather the idea that end-users can state intentions of what should be automated (Sarkar 2023). On the other hand, this works only ‘in the small’ as the complexity of ‘real-world’ software systems prohibits a specification in ‘natural language’ and therefore also highlights limitations of what LLMs can achieve (Yellin 2023).

3.2 | Using Generated Output

Despite the impressive performance of LLMs, their generated artefacts are not reliable and require human judgement (Denning and Arquilla 2022). LLMs are known to ‘hallucinate’, that is to generate linguistically correct, but factually incorrect text (Maynez et al. 2020; Ye et al. 2023). After all, these are language models made to mimic human language, not factual databases. In programming, LLMs likewise may generate programs that contain syntactic errors, bugs, do not solve the problem at hand or are vulnerable to cyberattacks (Becker et al. 2023; Sandoval et al. 2023; Wermelinger 2023). For instance, Wermelinger (2023) points out that “in spite of all the hype, using tools like Copilot can be a frustrating ‘hit and miss’ affair”. A critical assessment of the generated artefacts by the user is thus indispensable.

This has led to calls that programming education, say, should focus more on understanding and critically evaluating code as generated by LLMs, for instance (Becker et al. 2023; Lau and Guo 2023)—or as Wermelinger (2023) puts it: “algorithmic thinking, program comprehension, debugging and communication skills are as needed as ever”. It is not clear at all, however, how this educational shift towards more reading and understanding should be implemented, even though the need for better code comprehension has been known for some time now (Lister et al. 2004).

Even in terms of creativity, we find that generative AI is limited: “ChatGPT needs a guiding hand to produce something truly creative and that it works better as a sounding board for ideas or a conversational partner than as a competent writer on its own” (French et al. 2023). Similarly, Barke et al. (2023) found that study participants would use LLMs either in ‘acceleration mode’ or ‘exploration mode’, where the latter corresponds to the ‘sounding board idea’. This idea is further echoed by users who report that LLMs provide a great means to obtain a first draft to then work on and improve (Vaithilingam et al. 2022) or that LLMs are great for ‘ideation and brainstorming’ (Zastudil et al. 2023). In all these cases, the output of generative AI does not provide an end product, but helps the users develop their ideas.

Interestingly, a lack of trust in the LLM's output does not exclude finding it helpful (Amoozadeh et al. 2024). What seems paradoxical at first might on reflection express the fact that a critical stance towards the generated output is a prerequisite for successful utilisation, whereas those who are less critical might find the output much less helpful than expected.

In conclusion, a key to proper and beneficial usage of LLMs is the ability to critically assess and gauge the generated output, which requires an expertise novices do not possess.

3.3 | From Novice to Expert

The craft of prompt engineering and the need for critically assessing the output of LLMs already indicate that successful utilisation of LLMs requires some basic skills. More precisely, in order to successfully use an LLM to generate program code, say, the user must be able to understand the generated code and possibly modify that code or refine the prompt. In one study, professional software developers therefore all agree that “to benefit from these tools, people should have a grasp of programming fundamentals” (Bull and Kharrufa 2023). Similarly, Philbin (2023) remarks that using an LLM for education “only works if the programs returned are simplistic enough for novices to engage with and understand”. Finally, Becker et al. (2023) ask whether we could reasonably expect novice students to differentiate between code suggestions.

This gap between novice and expert programmers in terms of utilising LLMs is confirmed by user studies. Both Kazemitabaar et al. (2023) and Nam et al. (2023) report that those with more prior programming competency benefit more from these tools. However, even students who lack the expertise to evaluate the generated code seek out the assistance of LLMs (Amoozadeh et al. 2024). Amoozadeh et al. (2024) thus raise the question of whether novice programmers can actually "calibrate their trust in GenAI tools, and if so at what level and with what prerequisite knowledge?" Zastudil et al. (2023) speak of "calibrating a healthy scepticism in generative AI", thus using almost the same words.

The notion of 'calibrating trust in tools' refers to a sweet spot between blindly trusting and accepting any output, and entirely refraining from using any of these tools. It essentially means to develop the ability to critically assess or evaluate output from generative AI, which sounds suspiciously like an overarching learning objective with regards to generative AI tools and their usage. Moreover, this formulation makes clear, once again, how much the two notions of trust with respect to machines and people, respectively, differ.

In the context of programming education, LLMs can not only be used for generating code, but also for explaining code. This use case is probably more suited for supporting novices as the prompt is fairly standardised and the natural-language output requires less existing programming skills. However, MacNeil et al. (2023) found that such code explanations did not always live up to expectations in that they tend to go off-topic, focus on mundane aspects, are overly detailed, or give line-by-line descriptions rather than conceptual explanations, say. Students still rated the code explanation by LLMs as useful.

These findings strongly suggest that LLMs are particularly good at boosting the performance or productivity of experienced and skilled users. Translated to a classroom setting, this means that the best students benefit most from LLMs and hence any gap between stronger and weaker students will likely be exacerbated.

There is, however, another side to this that we should keep in mind. Novices and experts do not only differ quantitatively but also qualitatively. For instance, while novices tend to reason 'backwards' by testing each possible hypothesis on the data, experts rather reason 'forwards' by using the data to narrow down the possible hypotheses (Kirschner 2009). This is indicative of entirely different ways of approaching problems and organising the 'world' (i.e., the field of study in this case). In fact, "experts have acquired extensive knowledge that affects what they notice and how they organize, represent, and interpret information in their environment. This, in turn, affects their abilities to remember, reason, and solve problems" (Bransford et al. 2000). Kirschner, for instance, therefore argues for a strict separation between epistemology and pedagogy: "the way an expert works in his or her domain (epistemology) is not equivalent to the way one learns in that area (pedagogy)" (Kirschner et al. 2006).

It seems this understanding of how novices and experts fundamentally differ is frustratingly often absent in computing education. It has been argued time and again, for instance, that beginners should use professional languages and IDEs as the 'real deal' instead of pedagogically motivated programming environments (e.g., Chen and Marx (2005); Mannila et al. (2006); Vihavainen et al. (2014)). Similarly, we now see cases where the rationale for introducing LLMs into (computing) education is that professionals use them. The professionals' rationale, in turn, is mostly for productivity gains, which is entirely at odds with the needs of a novice. A novice does not need to be efficient or productive, but rather to have experiences and learn. Little surprise then that one student explained that relying on LLMs "ruins the purpose of learning in the first place" (Zastudil et al. 2023) and other students equally voiced concerns about the quality of their education if they relied too much on LLMs (over-reliance is in fact frequently brought up in the literature, see Section 2.4).

3.4 | Socio-Economics

A number of papers on LLMs in education emphasise that LLMs such as ChatGPT or GitHub Copilot are freely available—at least to students and teachers (Becker et al. 2023; Denny, Kumar, et al. 2023; Dobslaw and Bergh 2023; Finnie-Ansley et al. 2023; Kiesler and Schiffner 2023; Lau and Guo 2023; Prather, Reeves et al. 2023; Prather, Denny, et al. 2023; Raman and Kumar 2022; Savelka, Agarwal, An, et al. 2023; Savelka, Agarwal, Bogart, and Sakr 2023; Wang et al. 2023; Wermelinger 2023). This leads to suggestions that generative AI has the power to "democratise access to help" such as "high-quality tutoring" (Zastudil et al. 2023) or provide opportunities to "learners without access to formal education" (Denny, Khosravi, et al. 2023). More powerful versions such as GPT-4, however, are usually available through paid service only (Fernandez and Cornell 2024). The free availability of these models to students is seen both as a reason for quick uptake by students, but also as an opportunity for more accessible education.

Citing the free availability of LLMs ignores two major socio-economic factors that should be taken into account. First, even basic internet access is not necessarily available to everyone. The recent pandemic has highlighted the 'digital divide' in the UK, for instance, where families with low socio-economic status had limited or no access to the internet (BBC 2021b; Holmes and Burgess 2020). This is, of course, a global phenomenon and not limited to any single country (Ma 2021). Second, the observation that only LLMs with limited capabilities are available in free tiers should give pause as it means that students of higher socio-economic status can afford to use more powerful tools and therefore gain a further advantage.

This issue of the digital divide is not a new phenomenon related exclusively to LLMs. It has been found that socio-economic background has a much larger influence on 'digital literacy' or reading habits than, e.g., age (De Bruyckere et al. 2016; Kirschner and De Bruyckere 2017; Hargittai 2010). In other words, it is not a question of a generation of 'digital natives' that has better computer-skills, but a question of socio-economic background.

Even if we disregard issues of privacy and data collection, we should be wary of the idea that LLMs are a freely and easily available technology to everyone. Rather, LLMs are readily available to an elite who can afford access and LLMs are thus likely to contribute to a further widening of the digital divide. French et al. (2023) draws a similar picture: "A deeper worry is that reliance on AI will ultimately erode human higher-order cognitive skills, placing knowledge and power in the circuitry of a few supercomputers managed by an elite."

Finally, M. Jordan et al. (2024) draw our attention to the fact that LLMs may "introduce additional barriers for non-native English speakers” and find that “the abilities of GPT-3.5 for problem generation are not the same across language” (cf. Liang et al. (2023)).

3.5 | The ‘Natural’ Interface

To summarise: a powerful natural-language interface that may be used free of charge—this sounds like a perfect opportunity to widen access to today's crucial computing resources. Closer inspection, however, reveals that many of those promises do not hold. The LLMs' lack of understanding of intent as well as reliability means that both the input and output require high levels of expertise for successful usage. Moreover, LLMs are ‘free’ only within a very narrow interpretation of the word and mainly for those who already have access to good computing infrastructure. In combination, these issues not only highlight that the usually drawn picture of LLMs as ‘free tools for everyone’ is heavily skewed and biased, but it also points to LLMs being poised to increase gaps in access, rather than narrowing them.

In extension, we should be particularly careful with any claims of ‘democratisation’. The development of LLMs takes place in a highly competitive multi-billion market where only the biggest tech companies command the means to drive LLM development (Whittaker 2021). That this has little to do with any thoughts of democracy is evidenced by the various lawsuits from artists, authors and open-source programmers whose work was used for training generative AI without consent, but also by the ‘cheap’ labour force used for part of ChatGPT's training (Perrigo 2023). Blackwell (2023) draws our attention to the fact that the “modern exhibition of AI, demonstrated by wealthy AI companies and research institutions” is “underpinned by the most exploitative kinds of anonymized and alienated labour” where “intelligent human work is [...] simply stolen”.

Nonetheless, given the huge impact LLMs have on current politics, economy and society, we should certainly have a discussion about the place that LLMs shall play in education. However, we must be aware that this technology comes at massive costs and is far from granting universal and equalised access to high-quality education.

4 | When Every Test Is A Turing Test

The discussion about student dishonesty and cheating quickly leads us to the question of whether we might be able to (reliably) discover such cheating attempts. In other words, given any written material we want to decide whether it has been authored by a human or a machine. This problem has, of course, a very long history in computer science and is known as the ‘Turing test’ (Turing 1950; Epstein et al. 2009). While originally intended as a thought experiment about the question of ‘machine intelligence’, we suddenly find ourselves in a situation where every grading and assessment of homework is also a Turing test.

4.1 | LLM Performance Evaluation

A common characteristic of the literature discussing LLMs is their assessment of recent LLM performance as ‘human-level’ or ‘human-like’ (Bellettini et al. 2023; Bubeck et al. 2023; Cipriano and Alves 2023; Denny, Prather, et al. 2024; Denny, Kumar, et al. 2023; Druga and Otero 2023; Finnie-Ansley et al. 2022; Joshi et al. 2024; Leinonen, Hellas, et al. 2023; Li et al. 2022; MacNeil et al. 2023; Malinka et al. 2023; Nguyen and Allan 2024; Orenstrakh et al. 2023; Ouh et al. 2023; Shen et al. 2024)—a rather vague term that is seldom fully explained or defined. Inferring from context, however, we find that it refers to two distinct characteristics of an LLM's performance:

1. Given a set of problems, the LLM generates a ‘solution’ to each of the problems. These solutions are then assessed according to a predefined metric such as, e.g., a unit test. The LLM achieves ‘human-level’ performance if its success rate in generating correct solutions is within the common bounds of humans' success rate.
2. Given a set of tasks, the LLM generates an output text for each task that is then compared to a text written by a human as a response to the same task. The LLM then achieves ‘human-level’ performance if experts either rate the quality of its output at least as high as the quality of the human-produced counterpart, or if they fail to reliably distinguish between the two versions.

The first notion of ‘human-level performance’ is important in the context of automated assessments, now fairly common in computing education for dealing with large student cohorts. It is therefore no wonder that we find a number of computing education papers looking at LLM performance in the context of automated assessments (Becker et al. 2023; Denny, Kumar, et al. 2023; Finnie-Ansley et al. 2022; Prather, Denny, et al. 2023; Savelka, Agarwal, Bogart, Song, et al. 2023; Savelka, Agarwal, An, et al. 2023).

In the context of essays—or written work in general—human evaluators are much more common (Mahon et al. 2023; Malinka et al. 2023; Wermelinger 2023; Yeadon et al. 2023). For example, Herbold et al. (2023) asked a pool of experts to compare argumentative essays written by ChatGPT with those written by students, and Steiss et al. (2024) asked a number of human raters to compare a feedback from ChatGPT with one from a human expert for each essay.

There is, of course, some grey area and overlap between these two categories. For instance, even though Finnie-Ansley et al. (2022) evaluated the code generated by the LLM through unit tests, they remark that LLMs “return often-correct, well-structured code that could pass as human-written”. Other papers are even more precise, such as, e.g., Harrer (2023): “a response to the prompt that [...] seems indistinguishable from the output a human counterpart might have produced”.

Interestingly, Herbold et al. (2023) explicitly note that while such approaches do “not directly assess the quality of the output, it serves as a Turing test designed to evaluate whether humans can distinguish between human- and AI-produced output”. Indeed, closer inspection reveals that both approaches of evaluation attempt to measure whether either a machine or a human can distinguish between LLM output and text produced by a human. In other words, the Turing test is the template with which LLM performance is typically measured and evaluated—at least in the area of education.

It is important to note that a 'human-like performance' does not make the system itself 'human-like'. Floridi (2023) points out that the term 'artificial intelligence' refers to a behaviour that would require intelligence if exhibited by a human, but this "does not mean that the machine *is* intelligent". In fact, "AI is not about reproducing [...] intelligence. It is about doing without it" (Floridi 2023). This distinction is crucial when assessing the performance of LLMs in educational settings. LLMs do not 'solve' the problems in the same way a human would, but merely generate a solution. Trying to measure or attribute 'problem-solving skills' of LLMs is thus highly misleading.

4.2 | Automated Detection

The major concern of students cheating is in part fuelled by the probabilistic nature of LLMs, which means that the generated output is quasi-unique. Classic tools for detecting plagiarism are therefore rendered useless. Moreover, people have been found to exhibit difficulties discerning between LLM-generated and human-written text (Köbis and Mossink 2021). This begs the question of whether other tools might be able to reliably discern whether a text (or program) was generated by AI or written by a human.

Orenstrakh et al. (2023) compared a number of different LLM detection tools and found that they "are not yet ready to be trusted blindly for academic integrity purposes". Moreover, they explicitly warn that despite seemingly great success statistics, "it is apparent that the number of potential false positives can lead to a wide array of issues, especially if being trusted for plagiarism detection at educational institutions." Likewise, Liang et al. (2023) also warn of misleading accuracy statistics as they found that LLM detectors are biased against non-native English writers. Part of the reason for misclassification is the limited vocabulary and diversity in language use of non-native English writers. Interestingly, ChatGPT can be used to improve the writing style, which lowers the risk of being classified as AI. So, "paradoxically, GPT detectors might compel non-native writers to use GPT more to evade detection" (Liang et al. 2023).

The difficulties of identifying texts written by an LLM have led to repeated calls for watermarking such texts (e.g., Kirchenbauer et al. (2023)). Unfortunately, it has proven extremely challenging to devise a robust watermark that would not significantly lower the quality of the generated text. In virtually all instances, a watermark can be simply removed by asking a different LLM to paraphrase the output of the first LLM.

4.3 | Certificates and Diplomas

A crucial question arises from the entire discussion about cheating, academic integrity and whether the output of LLMs can pass exams: why does it matter? From a naïve point of view, we would expect that our students are intrinsically motivated to learn. Stajano (2023) recently commented that the aim of learning is not to pass a test, but to achieve a level of competence where the test can be passed. Although most educators might agree with this sentiment, it contrasts starkly with the discussion surrounding LLMs and the machines' ability to pass tests.

One possible explanation might be offered by the chain of trust and the issuing of certificates and diplomas. By issuing a diploma to a student, an educational institution vouches for the student's abilities and skills according to the standard set by the respective course or programme. If students start to fall short of the expectations linked to a specific diploma or certificate, the reputation of and trust in the institution is jeopardised—and with this its very existence as an educational institution. From this point of view, it is not necessarily the learning process itself that is at stake, but the survival of institutionalised education in its current form.

4.4 | AI-Proof Assignments

It seems fairly obvious that a Turing test should be approached by asking questions that a machine could not possibly answer 'correctly' (i.e., in a convincing way). A well-crafted question would thus reveal whether the conversational partner is a mere machine or an actual human being (cf. Floridi and Chiriatti (2020); Harel and Marron (2024)). The idea of asking a question only humans could answer correctly is also at the basis of 'Completely Automated Public Turing test to tell Computers and Humans Apart' (CAPTCHA).

In the spirit of such revealing questions or 'AI-proof assignments' (Cipriano and Alves 2023; Lau and Guo 2023), should we adapt our homework assignments or assessments to be based on questions and problems LLMs cannot really solve?

On the one hand, coming up with AI-proof assignments runs the danger of entering into an 'arms race' with LLM development, where teachers and educators have to constantly revise their pool of assignments and check against available LLMs—which will increase the likelihood that LLMs will learn to solve these assignments. For instance, Prather, Denny, et al. (2023) report that LLMs have significantly improved in terms of solving programming exercises within a remarkably short time, not to mention the differences between the earlier Codex/Copilot models and the latest release of ChatGPT, say. On the other hand, there is a real danger that changing the wording or making the questions 'harder' (e.g., problem-solving on a 'higher level') will introduce barriers first and foremost for the students and to a much lesser extent to the LLMs.

Let us abstract away from the question of whether LLMs exhibit any true kind of intelligence for a moment and assume that the LLMs' abilities to solve most exercises correctly and pass exams may stem from similar examples in the training data. This then implies that our exercises follow common patterns and adhere to (probably invisible and unconscious) common standards. This lack of excessive variation might actually be a good thing, as it points towards a shared culture and an agreement on what are good exercises. After all, there are only so many ways you can ask your students to solve the 'rainfall problem' (Finnie-Ansley et al. 2022; Fisler 2014), say.

At this point, we might also want to recall two important pedagogical principles. First, students should focus on the exercise or question at hand without having to spend too much time trying to figure out what is actually asked of them. Second, the purpose of these exercises is never to solve the problem as such, but rather that solving the problem teaches you something and supports your learning process. In other words, LLMs might be good at solving our assignments and exercises because there is both a shared body of knowledge and an (emergent) understanding of how to train our novices.

So, rather than trying to devise 'AI-proof assignments', we could understand an LLM's ability to find a solution as an indication that the question is clearly asked and understandable—although probably more in the sense that if an LLM cannot solve an exercise it might be because it is ambiguous or an ill-posed problem.

5 | The Aims of Education

Education is arguably one of the most important, most complex, and most expensive enterprises any society can embark on. It is nothing short of equipping and enabling our children to shape, become, and live as tomorrow's society. This is often reduced to the question of how to prepare our children for an unknown future, particularly now when technology seems to evolve at a staggering speed. However, this sees our students as mere passive recipients, rather than envisioning them as active creators of their future.

5.1 | Humanistic and Utilitarian Education

While a full discussion of the aims of education is clearly beyond the scope of this article, we do want to highlight two important stances towards the aim of education (in a clearly simplified version): the aim of a *humanistic education* is to help each individual to unfold, evolve, grow and fulfil their full potential as humans. In contrast, the aim of a *utilitarian education* is to prepare an individual for the labour market and workforce, enabling them to be productive and prosper economically.

The ideas of a humanistic education are nicely laid out by Whitehead (1959), where he stresses that "we have to remember that the valuable intellectual development is self-development". Interestingly, Whitehead puts the question of utility at the core of his discussion on the aims of education. We need to be aware, though, that his use and understanding of 'utility' differs greatly from a utilitarian standpoint as he puts it into the context of 'utility for the individual learner'.

If we take this focus on the learner as a human seriously, we find that education is governed by technology and trends only to a very little extent. An education that has value—in that it helps develop the cognitive faculties, for instance—has value irrespective of any tools that might be used. Hence, the arrival of LLMs will have only a marginal effect on any humanistic education.

Yet, at least in computing education, the utilitarian stance on education is clearly prevalent. Guzdial (2015), for instance, writes that "today, most of the arguments that I read for computing in schools are based on jobs" and "one argument to teach everyone about computing is that we need more workers who can program." Bull and Kharrufa (2023) are equally clear by highlighting "an intention within computing education practice to have content and methods informed by industry". Similar notions have been discussed in Section 2.3.

However, basing computing education too much on industry comes with the danger of a vicious circle as pointed out by Wirth (2002). If we teach our students what is used in industry and industry relies on what students have learned, we will stagnate and never be able to significantly progress. In our view, it is, moreover, remarkable that computer science is one of the only 'scientific' study programmes that is expected to prepare students directly for the labour market.

It has been evident for some time now that generative AI tools like LLMs have a direct impact on professionals; in particular, they are meant to increase the productivity of the workforce. It is from this impact that many authors deduce that LLMs must become part of (computing) education (Alves and Cipriano 2023; Becker et al. 2023; Brennan and Lesage 2023; Bull and Kharrufa 2023; Ernst and Bavota 2022; Puryear and Sprint 2022), which is entirely in line with the utilitarian view on education. In addition to the fallacy of not understanding the difference between novice and expert (Section 3.3), we should be aware that this argument for the inclusion of LLMs is only valid within this frame of mind.

The utilitarian standpoint is not the only one adopted in computing education, as it is to some degree at odds with the 'Computational Thinking' movement, emphasising the utility of computer science far beyond work-related issues (Tedre and Denning 2016; Wing 2006). This grew out of establishing computer science as a rigorous academic discipline, which was not easy (Tedre 2018). To this day, we still fight to establish computing education not as a mere learning of tools like office suites, but as a foundational and cognitive discipline of problem-solving and strategies. With the arrival of LLMs, we have to be very careful not to succumb to yet another tool that seems to render all other computing skills moot. Reviving the rallying cry that 'computer science is no more about using a machine than astronomy is about using telescopes'[1], we might want to emphasise that computing education is never about using a tool, not even one as seemingly sophisticated as LLMs.

Naturally, this does not mean that we argue for a ban on LLMs from educational practice. However, it cannot be an objective of serious computing education to learn how to use a specific tool, that is computing education must not be 'about using LLMs'.

5.2 | The Human Element

With generative AI reaching 'human-level performance' to the point where examiners face Turing tests (Section 4.1), we might want to consider the importance and the role of humans in education. In a way, automation of education has now reached a point where a machine can design the curriculum, ask the questions, do the assessment, but also answer the questions to successfully pass the assessment, or as Rudolph et al. (2023) put it: "A first AI circumvents a second AI and is assessed by a third AI. All that the humans do is press a couple of keys and nobody learns anything."

It is fairly obvious that we cannot replace the learner or student, as they are the nucleus of all education. A large part of the concerns around LLMs in education revolve around exactly this point: that students could now be imitated by machines, which would immediately render the entire enterprise meaningless and moot. However, it is not quite as obvious whether (human) teachers are as necessary for education as the students with various authors suggesting ideas such as that LLMs might offer "cheaper access to high-quality tutoring" (Zastudil et al. 2023).

Studies on learning have invariably pointed to the importance of the teacher and the teacher-student relationships (e.g., Hattie (2003); Hattie and Yates (2013)). Accordingly, De Bruyckere et al. (2016) warn that "the crucial factor for learning improvement is to make sure that you do not replace the teacher as the instrument of instruction, allowing computers to do what teachers would normally do, but instead use computers to supplement and amplify what the teacher does." Moreover, students have indicated that they prefer human teachers for assessing and grading their work (Tossell et al. 2024).

On a more student-oriented note, the human element is just as important. Guzdial (2015) argues that a key driver for student motivation is human relationships, such as peers, being part of a community, and following role models, including teachers. That students preferred being assessed by a human (Tossell et al. 2024) might also be part of the picture of motivation: after all, what does it matter to strive or excel if nobody notices? Finally, relying on automated teaching could have serious repercussions on motivation, as evidenced by the low completion rates of MOOCs (K. Jordan 2015).

In our view, the student-teacher relationship is absolutely crucial for any meaningful education. The more we devalue and replace human teachers with machines, the more we jeopardise the quality of education. While LLMs can be great tools to enhance, they must never replace humans in positions where the human element really matters: "The role of AI systems is to augment human intelligence and to assist, not replace human decision making and knowledge retrieval" (Harrer 2023).

6 | Conclusion

The arrival of generative AI and LLMs, in particular, has led to a 'moral panic' with calls to immediately react and transform education. There is a perceived danger that students might use these tools to cheat and render any existing assessments useless. On the flip side, LLMs are seen as determining the shape of future workplaces, from which it is argued that education needs to better prepare students for that new reality. Moreover, LLMs are hailed as a means to provide equal access to and even 'democratise' high-quality education.

The fear of students cheating reveals a rather negative picture of our student population, their work ethics, and stands in contrast to surveys that found students using these tools to primarily automate the 'boring' stuff. The argument of LLMs shaping future workplaces is problematic in two ways: on the one hand, it denies our ability to also influence and have a say in the shaping of the future. On the other hand, it reduces education to a utilitarian exercise to prepare humans for labour. Finally, the opportunity for wider access to high-quality education stands in stark contrast to past experience with technology-driven education as well as our understanding of how learning works.

It is without doubt that generative AI can be a powerful tool to improve education and we are far away from suggesting to completely ban it. However, we also caution against a blind embrace of this new technology, with a strong focus on refuting the narrative of the urgency and immediacy of the issue. Education is far too valuable, important, and complex that we could allow ourselves to rush into something without due consideration and debate. The debate, however, should not be so much on what these tools can do, but rather on what the true aims of education are.

Finally, we would like to also caution against the inflationary use of comparing LLMs to 'human-level performance'. As the philosopher McGilchrist (2022) noted, the comparison is a two-way street: "the belief that machines could become sentient is really just the obverse face of the view that we, sentient beings, are really just machines."

Author Contributions

Tobias Kohn: conceptualization, writing – review and editing, writing – original draft, investigation, formal analysis.

Acknowledgements

Open Access funding enabled and organized by Projekt DEAL.

Conflicts of Interest

Prof. Dr. Natalie Kiesler (co-authored paper in 2023).

Data Availability Statement

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Endnotes

[1] This is often attributed to E. Dijkstra, but the true origin remains unclear.

References

Ahmed, T., N. R. Ledesma, and P. Devanbu. 2022. "Synshine: Improved Fixing of Syntax Errors." *IEEE Transactions on Software Engineering* 49, no. 4: 2169–2181.

Al-Hossami, E., R. Bunescu, J. Smith, and R. Teehan. 2024. "Can Language Models Employ the Socratic Method? Experiments With Code Debugging." In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (53–59).

Al-Hossami, E., R. Bunescu, R. Teehan, L. Powell, K. Mahajan, and M. Dorodchi. 2023. "Socratic Questioning of Novice Debuggers: A Benchmark Dataset and Preliminary Evaluations." In Proceedings of the 18th Workshop on Innovative Use of Nlp for Building Educational Applications (Bea 2023) (709–726).

Alves, P., and B. P. Cipriano. 2023. "The Centaur Programmer–How Kasparov's Advanced Chess Spans Over to the Software Development of the Future." arXiv preprint arXiv:2304.11172.

Amoozadeh, M., D. Daniels, D. Nam, et al. 2024. “Trust in Generative Ai Among Students: An Exploratory Study.” In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (67–73).

Arawjo, I. 2020. “To Write Code: The Cultural Fabrication of Programming Notation and Practice.” In Proceedings of the 2020 Chi Conference on Human Factors in Computing Systems (1–15).

Balse, R., B. Valaboju, S. Singhal, J. M. Warriem, and P. Prasad. 2023. “Investigating the Potential of Gpt-3 in Providing Feedback for Programming Assessments.” In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education v. 1 (292–298). Association for Computing Machinery. https://doi.org/10.1145/3587102.3588852.

Barke, S., M. B. James, and N. Polikarpova. 2023. “Grounded Copilot: How Programmers Interact With Code-Generating Models.” *Proceedings of the ACM on Programming Languages* 7, no. OOPSLA1: 85–111. https://doi.org/10.1145/3586030.

BBC. 2021a. “Alexa Tells 10-Year-Old Girl to Touch Live Plug With Penny.” https://www.bbc.com/news/technology-59810383.

BBC. 2021b. “Digital Divide ‘Locking Children Out of Education’.” https://www.bbc.com/news/uk-england-55816686.

Becker, B. A., P. Denny, J. Finnie-Ansley, A. Luxton-Reilly, J. Prather, and E. A. Santos. 2023. “Programming Is Hard – Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation.” In Proceedings of the 54th Acm Technical Symposium on Computer Science Education *v. 1* (500–506).

Bellettini, C., M. Lodi, V. Lonati, M. Monga, and A. Morpurgo. 2023. “Davinci Goes to Bebras: A Study on the Problem Solving Ability of Gpt-3.” In Csedu 2023-15th International Conference on Computer Supported Education (2, 59–69).

Bird, C., D. Ford, T. Zimmermann, et al. 2022. “Taking Flight With Copilot: Early Insights and Opportunities of Ai-Powered Pair-Programming Tools.” *Queue* 20, no. 6: 35–57.

Blackwell, A. F. 2023. “Article Commentary: The Two Kinds of Artificial Intelligence, or How Not to Confuse Objects and Subjects.” *Interdisciplinary Science Reviews* 48, no. 1: 5–14.

Bransford, J. D., A. L. Brown, R. R. Cocking, et al. 2000. *How People Learn*. Vol. 11. National academy press.

Brennan, R. W., and J. Lesage. 2023. *Exploring the Implications of Openai Codex on Education for Industry 4.0*, edited by T. Borangiu, D. Trentesaux, and P. Leitão, 254–266. Springer International Publishing.

Bubeck, S., V. Chandrasekaran, R. Eldan, et al. 2023. “Sparks of Artificial General Intelligence: Early Experiments With Gpt-4.” arXiv preprint arXiv:2303.12712.

Bull, C., and A. Kharrufa. 2023. “Generative Ai Assistants in Software Development Education: A Vision for Integrating Generative Ai Into Educational Practice, Not Instinctively Defending Against It.” *IEEE Software* 41: 1–9. https://doi.org/10.1109/MS.2023.3300574.

Chen, Z., and D. Marx. 2005. “Experiences With Eclipse Ide in Programming Courses.” *Journal of Computing Sciences in Colleges* 21, no. 2: 104–112.

Christian, A. 2023. “Panic and Possibility: What Workers Learned About AI in 2023. BBC.” https://www.bbc.com/worklife/article/20231219-panic-and-possibility-what-workers-learned-about-ai-in-2023.

Cipriano, B. P., and P. Alves. 2023. “Gpt-3 vs Object Oriented Programming Assignments: An Experience Report.” In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education *v. 1* (61–67).

Cohen, S. 1973. *Folk Devils and Moral Panics*. Paladin.

Dai, W., J. Lin, H. Jin, et al. 2023. “Can Large Language Models Provide Feedback to Students? A Case Study on Chatgpt.” In 2023 Ieee International Conference on Advanced Learning Technologies (Icalt) (323–325).

Dakhel, A. M., V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, and Z. M. J. Jiang. 2023. “Github Copilot Ai Pair Programmer: Asset or Liability?” *Journal of Systems and Software* 203: 111734.

De Bruyckere, P., P. A. Kirschner, and C. D. Hulshof. 2016. “Technology in Education: What Teachers Should Know.” *American Educator* 40, no. 1: 12.

Dean, A. K. 2023. “GPT Unicorn: A Daily Exploration of Gtp-4's Image Generation Capabilities.” https://adamkdean.co.uk/posts/gpt-unicorn-a-daily-exploration-of-gpt-4s-image-generation-capabilities.

Denning, P. J., and J. Arquilla. 2022. “The Context Problem in Artificial Intelligence.” *Communications of the ACM* 65, no. 12: 18–21.

Denny, P., B. A. Becker, J. Leinonen, and J. Prather. 2023. “Chat Overflow: Artificially Intelligent Models for Computing Education - RenAIssance or ApocAIypse?” In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education v. 1 (3–4). Association for Computing Machinery. https://doi.org/10.1145/3587102.3588773.

Denny, P., H. Khosravi, A. Hellas, J. Leinonen, and S. Sarsa. 2023. “Can We Trust Ai-Generated Educational Content? Comparative Analysis of Human and Ai-Generated Learning Resources.” arXiv preprint arXiv:2306.10509.

Denny, P., V. Kumar, and N. Giacaman. 2023. “Conversing With Copilot: Exploring Prompt Engineering for Solving CS1 Problems Using Natural Language.” In Proceedings of the 54th Acm Technical Symposium on Computer Science Education v. 1 (1136–1142). Association for Computing Machinery. https://doi.org/10.1145/3545945.3569823.

Denny, P., J. Leinonen, J. Prather, et al. 2023. “Promptly: Using Prompt Problems to Teach Learners How to Effectively Utilize Ai Code Generators.” arXiv preprint arXiv:2307.16364.

Denny, P., J. Leinonen, J. Prather, et al. 2024. “Prompt Problems: A New Programming Exercise for the Generative Ai Era.” In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* 296 302.

Denny, P., J. Prather, B. A. Becker, et al. 2024. “Computing Education in the Era of Generative Ai.” *Communications of the ACM* 67, no. 2: 56–67. https://doi.org/10.1145/3624720.

Denny, P., S. Sarsa, A. Hellas, and J. Leinonen. 2022. “Robosourcing Educational Resources–Leveraging Large Language Models for Learnersourcing.” arXiv preprint arXiv:2211.04715.

Dijkstra, E. 1979. “On the Foolishness of ‘Natural Language Programming’.” In Program Construction (International Summer School, Marktoberdorf, Germany, July 26-August 6, 1978) (51–53). Spriger.

Dobslaw, F., and P. Bergh. 2023. “Experiences With Remote Examination Formats in Light of Gpt-4.” In Proceedings of the 5th European Conference on Software Engineering Education (220–225).

Druga, S., and N. Otero. 2023. “Scratch Copilot Evaluation: Assessing Ai-Assisted Creative Coding for Families.” arXiv preprint arXiv:2305.10417.

Epstein, R., G. Roberts, and G. Beber. 2009. *Parsing the Turing Test*. Springer.

Ernst, N. A., and G. Bavota. 2022. “Ai-Driven Development Is Here: Should You Worry?” *IEEE Software* 39, no. 2: 106–110.

Fernandez, A. S., and K. A. Cornell. 2024. “Cs1 With a Side of Ai: Teaching Software Verification for Secure Code in the Era of Generative Ai.” In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (345–351).

Finnie-Ansley, J., P. Denny, B. A. Becker, A. Luxton-Reilly, and J. Prather. 2022. “The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming." In Proceedings of the 24th Australasian Computing Education Conference (10–19). Association for Computing Machinery. https://doi.org/10.1145/3511861.3511863.

Finnie-Ansley, J., P. Denny, A. Luxton-Reilly, E. A. Santos, J. Prather, and B. A. Becker. 2023. "My AI Wants to Know if This Will Be on the Exam: Testing Openai's Codex on CS2 Programming Exercises." In Proceedings of the 25th Australasian Computing Education Conference (97–104). Association for Computing Machinery. https://doi.org/10.1145/3576123.3576134.

Fisler, K. 2014. "The Recurring Rainfall Problem." In Proceedings of the Tenth Annual Conference on International Computing Education Research (35–42).

Fletcher, R., and R. K. Nielsen. 2024. "What Does the Public in Six Countries Think of Generative Ai in News?" Reuters Institute for the Study of Journalism.

Floridi, L. 2023. *The Ethics of Artificial Intelligence: Principles, Challenges, and Opportunities*. Oxford University Press.

Floridi, L., and M. Chiriatti. 2020. "Gpt-3: Its Nature, Scope, Limits, and Consequences." *Minds and Machines* 30: 681–694.

French, F., D. Levi, C. Maczo, A. Simonaityte, S. Triantafyllidis, and G. Varda. 2023. "Creative Use of OpenAI in Education: Case Studies From Game Development." *Multimodal Technologies and Interaction* 7, no. 8: 81.

Guzdial, M. 2015. *Learner-Centered Design of Computing Education: Research on Computing for Everyone*. Morgan & Claypool Publishers.

Harel, D., and A. Marron. 2024. "The Human-Or-Machine Issue: Turing-Inspired Reflections on an Everyday Matter." *Communications of the ACM* 67, no. 6: 62–69.

Hargittai, E. 2010. "Digital Na (t) Ives? Variation in Internet Skills and Uses Among Members of the "Net Generation"." *Sociological Inquiry* 80, no. 1: 92–113.

Harrer, S. 2023. "Attention Is Not all You Need: The Complicated Case of Ethically Using Large Language Models in Healthcare and Medicine." *eBioMedicine* 90: 104512.

Hattie, J. 2003. "Teachers Make a Difference, What Is the Research Evidence?" Australian Council for Educational Research.

Hattie, J., and G. C. Yates. 2013. *Visible Learning and the Science of How We Learn*. Routledge.

Herbold, S., A. Hautli-Janisz, U. Heuer, Z. Kikteva, and A. Trautsch. 2023. "A Large-Scale Comparison of Human-Written Versus Chatgpt-Generated Essays." *Scientific Reports* 13, no. 1: 18617.

Holmes, H., and G. Burgess. 2020. "Coronavirus has Intensified the uk's Digital Divide." https://www.cam.ac.uk/stories/digitaldivide.

Hoover, A. 2023. "Use of AI is Seeping Into Academic Journals—and it's Proving Difficult to Detect." https://www.wired.com/story/use-of-ai-is-seeping-into-academic-journals-and-its-proving-difficult-to-detect/.

Hoq, M., Y. Shi, J. Leinonen, et al. 2024. "Detecting Chatgpt-Generated Code Submissions in a cs1 Course Using Machine Learning Models." In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (526–532).

Ibrahim, H., R. Asim, F. Zaffar, T. Rahwan, and Y. Zaki. 2023. "Rethinking Homework in the Age of Artificial Intelligence." *IEEE Intelligent Systems* 38, no. 2: 24–27.

Idialu, J., D. Etsenake, and N. Abbas. 2017. "Whodunnit: Human or Ai?" University of Waterloo.

Ishizue, R., K. Sakamoto, H. Washizaki, and Y. Fukazawa. 2024. "Improved Program Repair Methods Using Refactoring With Gpt Models." In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (569–575).

Jalil, S., S. Rafi, T. D. LaToza, K. Moran, and W. Lam. 2023. "Chatgpt and Software Testing Education: Promises & Perils." In 2023 Ieee International Conference on Software Testing, Verification and Validation Workshops (Icstw) (pp. 4130–4137).

Jiang, E., E. Toh, A. Molina, et al. 2022. "Discovering the Syntax and Strategies of Natural Language Programming With Generative Language Models." In Proceedings of the 2022 Chi Conference on Human Factors in Computing Systems (1–19).

Jordan, K. 2015. "Massive Open Online Course Completion Rates Revisited: Assessment, Length and Attrition." *International Review of Research in Open and Distance Learning* 16, no. 3: 341–358. https://doi.org/10.19173/irrodl.v16i3.2112.

Jordan, M., K. Ly, and A. G. Soosai Raj. 2024. "Need a Programming Exercise Generated in Your Native Language? Chatgpt's Got Your Back: Automatic Generation of Nonenglish Programming Exercises Using Openai Gpt-3.5." In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (pp. 618–624).

Joshi, I., R. Budhiraja, H. Dev, et al. 2024. "Chatgpt in the Classroom: An Analysis of Its Strengths and Weaknesses for Solving Undergraduate Computer Science Questions." In Proceedings of the 55th Acm Technical Symposium on Computer Science Education v. 1 (625–631).

Kazemitabaar, M., J. Chow, C. K. T. Ma, B. J. Ericson, D. Weintrop, and T. Grossman. 2023. "Studying the Effect of AI Code Generators on Supporting Novice Learners in Introductory Programming." In Proceedings of the 2023 Chi Conference on Human Factors in Computing Systems. Association for Computing Machinery. https://doi.org/10.1145/3544548.3580919.

Kiesler, N., and D. Schiffner. 2023. "Large Language Models in Introductory Programming Education: Chatgpt's Performance and Implications for Assessments." arXiv preprint arXiv:2308.08572.

Kirchenbauer, J., J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein. 2023. "A Watermark for Large Language Models." In International Conference on Machine Learning (17061–17084).

Kirova, V. D., C. S. Ku, J. R. Laracy, and T. J. Marlowe. 2024. "Software Engineering Education Must Adapt and Evolve for an Llm Environment." In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (pp. 666–672).

Kirschner, P. A. 2009. "Epistemology or Pedagogy, That Is the Question." In *Constructivist Instruction: Success or Failure*, 144–157. Routledge/Taylor & Francis Group.

Kirschner, P. A., and P. De Bruyckere. 2017. "The Myths of the Digital Native and the Multitasker." *Teaching and Teacher Education* 67: 135–142.

Kirschner, P. A., J. Sweller, and R. E. Clark. 2006. "Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching." *Educational Psychologist* 41, no. 2: 75–86.

Kirschner, P. A., and J. J. van Merriënboer. 2013. "Do Learners Really Know Best? Urban Legends in Education." *Educational Psychologist* 48, no. 3: 169–183. https://doi.org/10.1080/00461520.2013.804395.

Klee, M. 2023. "Professor Flunks All His Students After Chatgpt Falsely Claims it Wrote Their Papers." https://www.rollingstone.com/culture/culture-features/texas-am-chatgpt-ai-professor-flunks-students-false-claims-1234736601/.

Köbis, N., and L. D. Mossink. 2021. "Artificial Intelligence Versus Maya Angelou: Experimental Evidence That People Cannot Differentiate Ai-Generated From Human-Written Poetry." *Computers in Human Behavior* 114: 106553. https://www.sciencedirect.com/science/article/pii/S0747563220303034. https://doi.org/10.1016/j.chb.2020.106553.

Lau, S., and P. Guo. 2023. "From 'Ban It Till We Understand It' to 'Resistance Is Futile': How University Programming Instructors Plan to Adapt as More Students Use Ai Code Generation and Explanation Tools

Such as Chatgpt and Github Copilot." In Proceedings of the 2023 Acm Conference on International Computing Education Research-*volume 1* (pp. 106–121).

Lee, V. R., D. Pope, S. Miles, and R. C. Zárate. 2024. "Cheating in the Age of Generative Ai: A High School Survey Study of Cheating Behaviors Before and After the Release of Chatgpt." *Computers and Education: Artificial Intelligence* 7: 100253. https://doi.org/10.1016/j.caeai.2024.100253. https://www.sciencedirect.com/science/article/pii/S2666920X24000560.

Leinonen, J., P. Denny, S. MacNeil, et al. 2023. "Comparing Code Explanations Created by Students and Large Language Models." In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education v. 1 (p. 124–130). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3587102.3588785.

Leinonen, J., A. Hellas, S. Sarsa, et al. 2023. "Using Large Language Models to Enhance Programming Error Messages." In Proceedings of the 54th Acm Technical Symposium on Computer Science Education v. 1 (563–569). Association for Computing Machinery. https://doi.org/10.1145/3545945.3569770.

Li, Y., D. Choi, J. Chung, et al. 2022. "Competition-Level Code Generation With Alphacode." *Science* 378, no. 6624: 1092–1097.

Liang, W., M. Yuksekgonul, Y. Mao, E. Wu, and J. Zou. 2023. "Gpt Detectors Are Biased Against Non-Native English Writers." *Patterns* 4, no. 7: 100779.

Liffiton, M., B. E. Sheese, J. Savelka, and P. Denny. 2023. "Codehelp: Using Large Language Models With Guardrails for Scalable Support in Programming Classes." In Proceedings of the 23rd Koli Calling International Conference on Computing Education Research (1–11).

Lister, R., E. S. Adams, S. Fitzgerald, et al. 2004. "A Multi-National Study of Reading and Tracing Skills in Novice Programmers." In Working Group Reports From Iticse on Innovation and Technology in Computer Science Education (119–150). Association for Computing Machinery. https://doi.org/10.1145/1044550.1041673.

Liu, M., and F. M'hiri. 2024. "Beyond Traditional Teaching: Large Language Models as Simulated Teaching Assistants in Computer Science." In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (743–749).

Lu, X., S. Fan, J. Houghton, L. Wang, and X. Wang. 2023. "Readingquizmaker: A Human-Nlp Collaborative System That Supports Instructors to Design High-Quality Reading Quiz Questions." In Proceedings of the 2023 Chi Conference on Human Factors in Computing Systems (pp. 1–18).

Ma, J. K.-H. 2021. "The Digital Divide at School and at Home: A Comparison Between Schools by Socioeconomic Level Across 47 Countries." *International Journal of Comparative Sociology* 62, no. 2: 115–140.

MacNeil, S., A. Tran, A. Hellas, et al. 2023. "Experiences From Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book." In Proceedings of the 54th Acm Technical Symposium on Computer Science Education v. 1 (931–937). Association for Computing Machinery. https://doi.org/10.1145/3545945.3569785.

Mahon, J., B. Mac Namee, and B. A. Becker. 2023. "No More Pencils no More Books: Capabilities of Generative Ai on Irish and UK Computer Science School Leaving Examinations." In Proceedings of the 2023 Conference on United Kingdom & Ireland Computing Education Research (1–7).

Malinka, K., M. Peresíni, A. Firc, O. Hujnák, and F. Janus. 2023. "On the Educational Impact of Chatgpt: Is Artificial Intelligence Ready to Obtain a University Degree?" In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education v. 1 (47–53).

Mannila, L., M. Peltomäki, and T. Salakoski. 2006. "What About a Simple Language? Analyzing the Difficulties in Learning to Program." *Computer Science Education* 16, no. 3: 211–227.

Marcus, G. 2023. "The Sparks of Agi? or the End of Science?" https://garymarcus.substack.com/p/the-sparks-of-agi-or-the-end-of-science.

Maynez, J., S. Narayan, B. Bohnet, and R. McDonald. 2020. "On Faithfulness and Factuality in Abstractive Summarization." arXiv preprint arXiv:2005.00661.

McGilchrist, I. 2022. "Artificial Intelligence and the Matter With Things. Keynote Talk. (AI World Summit 2022)."

Nam, D., A. Macvean, V. Hellendoorn, B. Vasilescu, and B. Myers. 2023. "In-Ide Generation-Based Information Support With a Large Language Model." arXiv preprint arXiv:2307.08177.

Nguyen, H., and V. Allan. 2024. "Using Gpt-4 to Provide Tiered, Formative Code Feedback." In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (958–964).

Nunes, D., R. Primi, R. Pires, R. Lotufo, and R. Nogueira. 2023. "Evaluating Gpt-3.5 and Gpt-4 Models on Brazilian University Admission Exams." arXiv preprint arXiv:2303.17003.

Orenstrakh, M. S., O. Karnalim, C. A. Suarez, and M. Liut. 2023. "Detecting LLM-Generated Text in Computing Education: A Comparative Study for Chatgpt Cases."

Ouh, E. L., B. K. S. Gan, K. Jin Shim, and S. Wlodkowski. 2023. "Chatgpt, Can You Generate Solutions for My Coding Exercises? An Evaluation on Its Effectiveness in an Undergraduate Java Programming Course." In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education v. 1 (p. 54–60). Association for Computing Machinery. https://doi.org/10.1145/3587102.3588794.

Perrigo, B. 2023. "Exclusive: OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic. Time Magazine." https://time.com/6247678/openai-chatgpt-kenya-workers/.

Philbin, C. A. 2023. "Exploring the Potential of Artificial Intelligence Program Generators in Computer Programming Education for Students." *ACM Inroads* 14, no. 3: 30–38.

Phung, T., J. Cambronero, S. Gulwani, et al. 2023. "Generating High-Precision Feedback for Programming Syntax Errors Using Large Language Models." arXiv preprint arXiv:2302.04662.

Prasad, P., and A. Sane. 2024. "A Self-Regulated Learning Framework Using Generative Ai and Its Application in Cs Educational Intervention Design." In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (1070–1076).

Prather, J., P. Denny, J. Leinonen, et al. 2023. "The Robots Are Here: Navigating the Generative Ai Revolution in Computing Education." In Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education (108–159).

Prather, J., B. N. Reeves, P. Denny, et al. 2023. "'It's Weird That It Knows What I Want': Usability and Interactions With Copilot for Novice Programmers." *ACM Transactions on Computer-Human Interaction* 31, no. 1: 1–31.

Puryear, B., and G. Sprint. 2022. "Github Copilot in the Classroom: Learning to Code With Ai Assistance." *Journal of Computing Sciences in Colleges* 38, no. 1: 37–47.

Rajabi, P., P. Taghipour, D. Cukierman, and T. Doleck. 2023. "Exploring Chatgpt's Impact on Post-Secondary Education: A Qualitative Study." In Proceedings of the 25th Western Canadian Conference on Computing Education (1–6).

Raman, A., and V. Kumar. 2022. "Programming Pedagogy and Assessment in the Era of AI/ML: A Position Paper." In Proceedings of the 15th Annual Acm India Compute Conference (29–34). Association for Computing Machinery. https://doi.org/10.1145/3561833.3561843.

Reeves, B., S. Sarsa, J. Prather, et al. 2023. “Evaluating the Performance of Code Generation Models for Solving Parsons Problems With Small Prompt Variations.” In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education *v. 1* (299–305).

Rogers, M. P., H. M. Hillberg, and C. L. Groves. 2024. “Attitudes Towards the Use (And Misuse) of Chatgpt: A Preliminary Study.” In *Proceedings of the 55th Acm Technical Symposium on Computer Science Education v. 1* (1147–1153).

Rudolph, J., S. Tan, and S. Tan. 2023. “Chatgpt: Bullshit Spewer or the End of Traditional Assessments in Higher Education?” *Journal of Applied Learning & Teaching* 6, no. 1: 342–363.

Sallam, M. 2023. “Chatgpt Utility in Healthcare Education, Research, and Practice: Systematic Review on the Promising Perspectives and Valid Concerns.” *Health* 11, no. 6: 887. https://doi.org/10.3390/healthcare11060887.

Sandoval, G., H. Pearce, T. Nys, R. Karri, S. Garg, and B. Dolan-Gavitt. 2023. “Lost at C: A User Study on the Security Implications of Large Language Model Code Assistants.” In 32nd Usenix Security Symposium (Usenix Security 23). (2205–2222).

Sarkar, A. 2023. “Will Code Remain a Relevant User Interface for End-User Programming With Generative Ai Models?” In Proceedings of the 2023 Acm Sigplan International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (153–167).

Sarsa, S., P. Denny, A. Hellas, and J. Leinonen. 2022. “Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models.” In Proceedings of the 2022 Acm Conference on International Computing Education Research1 (27–43): Association for Computing Machinery. https://doi.org/10.1145/3501385.3543957.

Savelka, J., A. Agarwal, M. An, C. Bogart, and M. Sakr. 2023. “Thrilled by Your Progress! Large Language Models (GPT-4) no Longer Struggle to Pass Assessments in Higher Education Programming Courses.” In Proceedings of the 2023 Acm Conference on International Computing Education Research - volume 1 (78–92). Association for Computing Machinery. https://doi.org/10.1145/3568813.3600142.

Savelka, J., A. Agarwal, C. Bogart, and M. Sakr. 2023. “Large Language Models (gpt) Struggle to Answer Multiple-Choice Questions About Code.” arXiv preprint arXiv:2303.08033.

Savelka, J., A. Agarwal, C. Bogart, Y. Song, and M. Sakr. 2023. “Can Generative Pre-Trained Transformers (GPT) Pass Assessments in Higher Education Programming Courses?” In Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education v. 1. Association for Computing Machinery. https://doi.org/10.1145/3587102.3588773.

Sheard, J., P. Denny, A. Hellas, J. Leinonen, L. Malmi, and Simon. 2024. “Instructor Perceptions of Ai Code Generation Tools-A Multi-Institutional Interview Study.” In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (1223–1229).

Shen, Y., X. Ai, A. G. Soosai Raj, R. J. Leo John, and M. Syamkumar. 2024. “Implications of Chatgpt for Data Science Education.” In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (1230–1236).

Stajano, F. 2023. “Frank Stajano Explains: I Failed this Important Exam. Why am I Smiling?” https://www.youtube.com/watch?v=n6_91ZH2iHg.

Steiss, J., T. Tate, S. Graham, et al. 2024. “Comparing the Quality of Human and Chatgpt Feedback of Students' Writing.” *Learning and Instruction* 91: 101894.

Taylor, A., A. Vassar, J. Renzella, and H. Pearce. 2024. “Dcc–Help: Transforming the Role of the Compiler by Generating Context-Aware Error Explanations With Large Language Models.” In Proceedings of the 55th Acm Technical Symposium on Computer Science Education *v. 1* (1314–1320).

Tedre, M. 2018. “The Nature of Computing as a Discipline.” In *Computer Science Education: Perspectives on Teaching and Learning in School*, edited by S. Sentance, E. Barendsen, and C. Schulte, 5–18. Bloomsbury Publishing.

Tedre, M., and P. J. Denning. 2016. “The Long Quest for Computational Thinking.” In Proceedings of the 16th Koli Calling International Conference on Computing Education Research (120–129), USA: Association for Computing Machinery. https://doi.org/10.1145/2999541.2999542.

Tossell, C. C., N. L. Tenhundfeld, A. Momen, K. Cooley, and E. J. de Visser. 2024. “Student Perceptions of Chatgpt Use in a College Essay Assignment: Implications for Learning, Grading, and Trust in Artificial Intelligence.” *IEEE Transactions on Learning Technologies* 99: 1–15.

Turing, A. M. 1950. “Computing Machinery and Intelligence.” *Mind* 49: 433–460. https://doi.org/10.1093/mind/LIX.236.433.

Vaithilingam, P., T. Zhang, and E. L. Glassman. 2022. “Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models.” In Extended Abstracts of the 2022 Chi Conference on Human Factors in Computing Systems. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3491101.3519665.

Vallance, C. 2022. “ChatGPT: New AI Chatbot has Everyone Talking to it. BBC.” https://www.bbc.com/news/technology-63861322.

Vallance, C. 2023. “AI Could Replace Equivalent of 300 Million Jobs – Report. BBC.” https://www.bbc.com/news/technology-65102150.

Vihavainen, A., J. Helminen, and P. Ihantola. 2014. “How Novices Tackle Their First Lines of Code in an Ide: Analysis of Programming Session Traces.” In Proceedings of the 14th Koli Calling International Conference on Computing Education Research (109–116). Association for Computing Machinery. https://doi.org/10.1145/2674683.2674692.

Wang, T., D. V. Díaz, C. Brown, and Y. Chen. 2023. “Exploring the Role of Ai Assistants in Computer Science Education: Methods, Implications, and Instructor Perspectives.” In 2023 Ieee Symposium on Visual Languages and Human-Centric Computing (Vl/Hcc) (92–102).

Weizenbaum, J. 1966. “Eliza—A Computer Program for the Study of Natural Language Communication Between Man and Machine.” *Communications of the ACM* 9, no. 1: 36–45.

Welsh, M. 2022. “The End of Programming.” *Communications of the ACM* 66, no. 1: 34–35. https://doi.org/10.1145/3570220.

Wermelinger, M. 2023. “Using Github Copilot to Solve Simple Programming Problems.” In Proceedings of the 54th Acm Technical Symposium on Computer Science Education v. 1 (172–178).

Whitehead, A. N. 1959. “The Aims of Education.” *Daedalus* 88, no. 1: 192–205.

Whittaker, M. 2021. “The Steep Cost of Capture.” *Interactions* 28, no. 6: 50–55.

Wing, J. M. 2006. “Computational Thinking.” *Communications of the ACM* 49, no. 3: 33–35. https://doi.org/10.1145/1118178.1118215.

Wirth, N. 2002. “Computing Science Education: The Road Not Taken.” *ACM SIGCSE Bulletin* 34, no. 3: 1–3. https://doi.org/10.1145/637610.544415.

Ye, H., T. Liu, A. Zhang, W. Hua, and W. Jia. 2023. “Cognitive Mirage: A Review of Hallucinations in Large Language Models.” arXiv preprint arXiv:2309.06794.

Yeadon, W., O.-O. Inyang, A. Mizouri, A. Peach, and C. P. Testrow. 2023. “The Death of the Short-Form Physics Essay in the Coming Ai Revolution.” *Physics Education* 58, no. 3: 035027.

Yellin, D. M. 2023. “The Premature Obituary of Programming.” *Communications of the ACM* 66, no. 2: 41–44.

Zastudil, C., M. Rogalska, C. Kapp, J. Vaughn, and S. MacNeil. 2023. "Generative Ai in Computing Education: Perspectives of Students and Instructors." In 2023 Ieee Frontiers in Education Conference (Fie) (1–9).

Zhang, J., J. Cambronero, S. Gulwani, et al. 2022. "Repairing Bugs in Python Assignments Using Large Language Models." arXiv preprint arXiv:2209.14876.