The Virtues and Vices of Life 'Close to the Machine'

Matthew Kuan Johnson and Lani Watson

This is an unpublished conference paper for the 9th Annual Jubilee Centre for Character and Virtues conference at Oriel College, Oxford University, Friday 10th – Sunday 12th September 2021.

These papers are works in progress and should not be cited without author's prior permission.

Jubilee Centre for Character and Virtues

University of Birmingham, Edgbaston, Birmingham, B15 2TT United Kingdom

T: +44 (0) 121 414 3602 F: +44 (0) 121 414 4875

E: jubileecentre@contacts.bham.ac.uk W: www.jubileecentre.ac.uk

Abstract

Much has been written, speculated and predicted about artificial intelligence (AI) and its capacity to replicate and simulate human intelligence. Much less has been said about causation in the other direction. While the advent of humanoid robots and machine learning algorithms modeled on human neurological systems exemplify AI and digital technologies formed in the image of their creators, it is also possible that working on these technologies can form creators in the image of their technological creations. In this paper, we examine the characterological effects of human-machine proximity from a virtue-theoretical perspective. We investigate both the virtues and vices that may be cultivated in those creating and working closely with technology and issue a call for more empirical work on the characterological effects of human-machine proximity, providing greater insight into the urgency and scope of this concern.

Introduction

Much has been written, speculated, and predicted about artificial intelligence (AI) and its capacity to replicate, simulate, or amalgamate human intelligence. Much less has been said, however, about causation in the other direction: while the advent of humanoid robots and machine learning algorithms modeled on human neurological systems exemplify AI and digital technologies formed in the image of their creators, it is also possible that working on these technologies can form creators – programmers, engineers, data analysts – in the image of their technological creations. Indeed, while there is currently a significant amount of interest in how the *use* of digital technologies may be forming the character of end users, we are interested in going further upstream and investigating how working on designing and developing these digital technologies themselves forms the character of the creators of these technologies. This upstream question is especially important, because any effects that working on such technologies might have on the creators, can influence how the digital platforms are developed (which will, in turn, affect the end users). Additionally, with the explosion of jobs in programming, software engineering, and data science (and with computer programming comprising increasingly larger portions of coursework in many educational programmes), it is critical to understand what the characterological benefits (and potential hazards) of working with these technologies might be.

In this paper, we examine the characterological effects of human-machine proximity from a virtue-theoretical perspective. We begin with some descriptions and anecdotes from programmers and data scientists about the characterological effects of life "close to the machine" (Section I). Next, we investigate both the virtues and vices that may be cultivated in those creating and working closely with AI and digital technologies, including increased rigour, creativity, and intellectual autonomy, alongside diminished capacity for moral perception and motivation (Sections II and III, respectively). After tracing some of these possible characterological effects, we lay out some reasons why this is an especially important and urgent question (Section IV). Ultimately, we conclude by issuing a call for more empirical work on the characterological effects of human-machine proximity (echoing a broader call by Scherer 2016), which would provide greater insight into the scope of this concern.

I. Previous Accounts of Characterological Effects of Life "Close to the Machine"

Computer science pioneer, Edsger W. Dijkstra, in his ACM Turing Lecture (his acceptance speech for the Turing Award) warned of "the influence of the tool we are trying to use upon our own thinking habits" (1972). He proposed that in comparing different programming languages, it is important to look at the effects that the languages have on the programmers who use them. To support this claim, he provides what he states is a well-known example among programmers: "the one-liners." Here, a particular programming language (he does not name which one) encourages writing in very short lines of code, and he points to what he claims is common knowledge in the programming community, which is that the programmers who start using this language begin to challenge one another to write increasingly shorter lines of code to run the same program. He claims that this program is "an open invitation for clever tricks," which he derides as "one of the most damning things that can be said about a programming language" (Dijkstra, 1972), expressing alarm that a programming language could, in this way, inculcate patterns of thought and interactions between programmers that replace generative, helpful, and collaborative interactions with "clever tricks" and one-upmanship.

In another example, he ran an "experiment" in which he asked people to write a programme, but in which none of the "participants" found the optimal (and most obvious) solution. This was because, "Upon closer analysis this turned out to have a common source: their notion of repetition was so tightly connected to the idea of an associated controlled variable to be stepped up, that they were mentally blocked from seeing the obvious. Their solutions were less efficient, needlessly hard to understand, and it took them a very long time to find them. It was a revealing, but also shocking experience for me" (Dijkstra, 1972). Here, he claims that being seeped in common conventions of programming had limited their ability to see the most obvious solution to the problem. In subsequent writing (Dijkstra, 1975/1985), he again suggests that, "The tools we use have a profound (and devious!) influence on our thinking habits, and, therefore, on our thinking abilities,"[1] and provides a series of examples of how using various programming languages changes one's cognitive development and thinking in particular ways.

Furthermore, the effects of life "close to the machine" can also be seen in the way that many programmers and data scientists have fallen prey to a "golden hammer" approach to solving problems (this is especially glaring in how large-scale societal issues are often approached as purely technical problems). This "golden hammer" approach is the phenomenon whereby spending so much time working with technical solutions has led to a mindset in which every societal problem is seen as having a technical solution – from attempting to solve large-scale societal loneliness with chatbots and care robots, to attempting to safeguard the integrity of elections by use of blockchain technologies, to endless apps intended to transform everything from the nature of work, to the nature of education, to the nature of money. Consequently, one data scientist describes the dominant mindset in Silicon Valley as follows, "I think the strangest thing about being out here in the Bay Area is that the

[1] While it goes beyond the scope of this paper, it is worth mentioning that there is some preliminary archaeological and cross-species evidence that the tools one adopts shapes the direction of cognitive evolution (Taylor & Gray, 2014; Overmann & Wynn, 2019). While this work concerns physical tools, it may well be the case that programming languages function as a kind of tool that similarly shapes cognitive development.

worldview has just completely saturated everything to the point that people think that everything is a technical problem that should be solved technologically…On the one hand, there's no better shepherd for the economy than an engineer; on the other hand, there's no worse shepherd for the economy than an engineer. Because that kind of machine thinking is very good at producing some things, and very, very bad at producing other things." (Tarnoff & Weigel, 2020, 127-128).

This mindset, whereby problems are approached via purely technical solutions, reveals a kind of restriction of focus, attention, and problem-solving ability brought about through life "close to the machine." Not all problems can be addressed (or are best addressed) through technical solutions, especially as some of the more thorny societal ills require complex and multifaceted solutions involving policy reform and deep structural change, and cannot simply be solved with a "disruptive" technical solution alone. The full range of necessary steps to a solution are, however, overlooked by the narrowing of mindset to focus only on technical solutions.

Numerous examples of the characterological effects of life "close to the machine" are also provided by Emily Ullman (1997), who coined the phrase "close to the machine" in her autobiography of that name. Her autobiography centers around her time as a programmer, describing how computer programmers who experience life "close to the machine" can experience a shift in their thinking and perception to more closely resemble the programs and systems they work on. She describes it as follows,

> "I'd like to think that computers are neutral, a tool like any other, a hammer that can build a house or smash a skull. But there is something in the system itself, in the formal logic of programs and data, that recreates the world in its own image. Like the rock-and-roll culture, it

> forms an irresistible horizontal country that obliterates the long, slow, old cultures of place and custom, law and social life. We think we are creating the system for our own purposes. We believe we are making it in our own image. We call the microprocessor the 'brain'; we say the machine has 'memory.' But the computer is not really like us. IT is a projection of a very slim part of ourselves: that portion devoted to logic, order, rule, and clarity. It is as if we took the game of chess and declared it the highest order of human existence" (p. 89).

Toward the beginning of her memoir, she provides an account of a particular instance in which the effects of her life "close to the machine" particularly impacted her thinking and perception. She describes a meeting with a focus group of end-users, who are providing her with feedback for a platform for AIDS patients that she is developing. She begins by describing her difficulty in relating to them and their needs and desires for the system, "Before the meeting, the users existed only in my mind, projections, all mine. They were abstractions, the initiators of tasks that set off remote procedure calls; triggers to a set of logical and machine events that ended in an update to a relational database on a central server. Now I was confronted with their fleshly existence…The machine events already had more reality, had been with me longer, than the human beings at the conference table" (pp. 11-12). She then describes how her time spent programming affects her ability to communicate with them, finding herself naturally falling into a mode of discourse in which: "My brain whirred out a stream of logic-speak," (p. 12) which her audience finds difficult to follow. In the end, she finds that her loyalty lies not with the "real human need, to the impact I would have on the working lives of these people at the table, on the people living with AIDS," but that instead, "What I really thought was this: I must save the system" (p. 13).

Throughout the book, she provides other examples of how life "close to the machine" affects human relationality, in describing how she and her fellow co-workers perceive and relate to one another ("Our physical selves have been battered away. Now we know each other in one way and one way only: the code." [p. 4]) and to everyday people ("I recognize my virtual colleagues by their overattention to little interactions with waiters and cashiers, a supersensitivity that has come from too much time spent alone. We've been in a machine-mediated world – computers and e-mail, phones and faxes – and suddenly we're in a world where people lumber up and down the steps of buses, walk in and out of stores, have actual in-person conversations. *All this has been going on while I was in another universe*: that's what comes to us with a force like the too-bright sun or a stiff wind off the bay." [pp. 138-139]). She details how the kind of thinking and perception of life "close to the machine" can even affect approaches to, and understandings of, physical intimacy ("[It was] algorithmic. I once thought that love could not be programmed, but now I wondered. This sex was formulaic, had steps and positions and durations, all tried and perfected, like a martial arts kata or a well-debugged program. My own role in it was like a user-exit subroutine, an odd branch where anything might happen but from which we must return, tracing back to the mainline procedure" [p. 49]).

She also describes how, while humans typically operates with high levels of ambiguity and unclarity, machines cannot operate with any. Consequently, in order to get into the right headspace to program effectively, programmers must enter into a thought-space of rigid rules and crystal clear commands, and she explains the effects this can have: "Soon the programmer has no choice but to retreat into some private interior space, closer to the machine, where things can be accomplished. The machine begins to seem friendlier than the analysts, the users, the managers. The real-world reflection of the program – who cares anymore? (p. 23)." Similarly, she argues that the affectively-inflected parts of human thinking, such as hope and fear, have no place in life "close to the machine," either. She notes how, “Some part of me mourns, but I know there is no other way: human needs must cross the line into code. They must pass through this semipermeable membrane where urgency, fear, and hope are filtered out, and only reason travels across. There is no other way” (p. 15).

One consequence of trading out some of these characteristically human components of thinking and perception is that there will be certain failures of moral perception and moral reasoning. Ullman explains how one of her boyfriends was interested in creating an online platform for the anonymous and unregulated sharing of pornography, and had no interest in considering the ethics involved, “The whole complicated business of international pornography had devolved, in Brian’s thinking, to the level of a mathematical problem, some famously difficult proof, a challenge of the mind. He seemed neither attracted to nor repulsed by the content of the stuff he would be sending around. To him, it was just bits, stuff on the wire” (p. 62). This is just one example of a common pitfall she identifies, which involves failing to engage moral perception and moral reasoning about the implications of the technology one is developing, as one instead becomes solely preoccupied with finishing the programming task at hand. She describes how,

> “The goal now is not whatever all the analysts first set out to do; the goal becomes the creation of the system itself. Any ethics or morals or second thoughts, any questions or muddles or exceptions, all dissolve into a junky Nike-mind: Just do it. If I just sit here and code, you think, I can make something run. When the humans come back to talk changes, I can just run the program. Show them: Here. Look at this. See? This is not just talk. This runs. Whatever you might say, whatever the consequences, all you have are words and what I have is this, this thing I’ve built, this operational system. Talk all you want, but this thing here: it *works*” (p. 24).

Finally, and relatedly, Ullman describes how the system can "infect its owner," such that the capability to acquire others' personal data increases one's desire to do so to increasing degrees, without any interest in considering the ethical questions involved. "You own the system, it's your data, you have power over it; and, once the system gives you this power, you suddenly can't help yourself from wanting more" (p. 88). She provides the example of a client who was interested in having his system set up to log his employees' keystrokes, not for any particular reason, but simply because he was "curious" and wanted to have the power to know what his employees were up to every day. Ullman asks him to provide any kind of substantive reason for the keystroke logging software (he cannot provide any) and then exhorts him to consider the ethical implications (he refuses). She concludes the anecdote by observing that, "Many years and clients later, this greed for more data, and more again, had become a commonplace. It had become institutionalized as a good feature of computer systems: you can link them up, you can cross-check, you can find out all sorts of things you didn't set out to know" (p. 89).

II. Virtues of human-machine proximity

Humans have been growing ever closer to their machines over the past 50 years, as computers have become smaller, faster, more powerful and, correspondingly, indispensable to daily life. The impacts of this have not gone unnoticed. Diverse responses to increased human-machine proximity range from the introduction of extensive new data and privacy laws, to increased resources for the expanding field of tech and AI ethics, in universities and tech companies alike. These responses focus predominantly on the wide range of ethical questions and challenges that human-machine proximity poses for individuals and society: biased algorithms, wearable technology, weaponized big data, and the ethical and legal treatment of AI, to name just a few.

A less prominent (but we think no less important) question to date, concerns the impact of human-machine proximity on character and character formation. Much of the theoretical and empirical work that has been done to assess the impact of human-machine proximity on humans themselves has been conducted in the classroom, with a focus on learning and skill development (Barr & Stephenson, 2011; Fessakis et al, 2013; Grover & Pea, 2013; Lye & Koh 2014). This work has covered diverse ground, including the acquisition and use of computational concepts (Hui and Umar, 2011; Kahn et al, 2011; Denner et al, 2012; Fessakis, Gouli and Mavroudi, 2013), alongside the development of skills such as problem-solving (Fessakis et al, 2013; Esteves, Fonseca, Morgado, and Martins 2011), language learning and composition (Miller, 2009; Lee, 2010; Burke, 2012), and information-processing (Garner 2009; Ismail et al, 2010; Ma et al., 2011; Urquiza-Fuentes and Velazquez-Iturbide, 2013).

One of the most widespread concepts to be introduced to the discourse has been that of 'computational thinking' (CT) (Wing, 2006). According to Wing (2006), who offered an influential definition of the concept, "computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (p.33). As Grover and Pea (2013) put it, "CT's essence is thinking like a computer scientist when confronted with a problem" (p.39). Wing (2006) argues that computational thinking "represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" (p.33). Significant research has focused on the development of computational thinking in students.

From this, we can gain valuable insights into the effects of human-machine proximity on thinking and learning, particularly among young people in educational settings. Many of these effects are reportedly positive. Lye and Koh (2014), for example, report that tools designed to teach computational thinking "can also engage students in the building of multi-media digital products, thereby enabling programming activities to be used as a means for students to express their ideas" (p.53). Likewise, these tools "allow students to acquire computational problem-solving practices more easily…and can possibly help K-12 students to extend these computational practices towards enhancing their general problem-solving ability" (Lye and Koh, 2014, p.53).

Such effects are an encouraging indication of the positive impact that human-machine proximity can have in the classroom. Nonetheless, we believe significantly more work is required in order to understand how and whether similar positive effects are seen outside of educational settings, particularly for those working closely with machines on a daily basis. This includes those building new technologies and writing intricate code, in the intensive and immersive environments of many of the world's fastest growing start-ups and largest tech companies. These workplace settings differ in important respects from structured classrooms, aimed at formal learning. For one, human-machine proximity is likely to be even closer for those immersed in the development and use of technology throughout the course of their day. Does increased proximity equate to further benefits, or is there a limit to the positive effects it can have? In addition, the primary motivations and incentives present in the tech sector are undoubtedly different from those of a school or university. How do such incentives affect the development of individuals?

Furthermore, and significantly for our purposes, more work is needed in order to assess the impact of close human-machine proximity on *character*, rather than merely skills. Character comprises an individual's deep and stable dispositions to think and act in certain ways; their virtues and vices. Does the intense and immersive experience of working in close proximity with machines impact upon an individual's moral and intellectual character? The evidence from educational research suggests that it may. Skills are, after all, regarded by many to be either a component of character (Zagzebski, 1996; Battaly, 2006; Baehr, 2011) or, as many in Ancient Greece had it, as equivalent to the virtues (Bloomfield 2000; Stichter, 2018). In either case, if any significant relationship holds between skills and virtues, then evidence that demonstrates the impact of human-machine proximity on the development of skills may also be pertinent to the cultivation of character.

Three dimensions of intellectual character are of particular interest in this regard. In the first instance, computer programming and computational thinking are both intricately bound to logic and reasoning skills. Angeli and Giannakos (2020), for example, write that computational thinking "enables the development of qualities such as abstract thinking, problem solving, pattern recognition, and logical reasoning." In turn, these thinking skills and processes can be viewed as integral to the cultivation and exercise of intellectual virtues such as attentiveness, rigour and intellectual perseverance. These virtues require, at least, the ability to engage in basic reasoning and a degree of abstract thinking, in order to pursue intellectual goods. If increases in computational thinking as a result of human-machine proximity result in improved reasoning skills, it is plausible that this proximity may also play a role in the cultivation of these associated intellectual virtues.

Secondly, computer programming has been found to have a positive impact on originality, a facet of creative thinking that involves selective encoding and combining, leading to the cultivation of creativity, creative problem solving, and divergent thinking (Horton and Ryba, 1986; Clements and Merriman, 1988; Clements, 1995). As Clements observed in a review of educational research on this topic in the mid-nineties:

> "If the creative act is viewed as the act of solving an ill-defined problem (Carey and Flower, 1989) or as a full expression of intelligence…there is strong support for the computer's facilitation of creativity." (Clements, 1995, p.157)

Both creativity and creative problem-solving can be viewed as aspects of intellectual character, or indeed virtues, in and of themselves (Gaut, 2014; Kieran, 2014; Baehr, 2018). As such, increased creativity as a result of human-machine proximity arguably has a direct impact on intellectual character, in this regard. Somewhat more speculatively, such an impact may plausibly be tied to the development of other intellectual virtues, such as curiosity and open-mindedness. The creative thinker is plausibly one who is characteristically disposed to seek out information, ask questions, and consider a range of alternative perspectives and ideas. At the very least, creativity is likely to feature in the expression of intellectual virtues such as these.

Lastly, several authors indicate the positive effects of increases in computational thinking on active learning. As Lye and Koh (2014) put it, "students are no longer passive consumers of the technology" (p.53). Similarly, Palma Rivas (2000) states that "computer-based learning gives students more control over their own learning" (p.76). These results suggest that human-machine proximity may have a positive impact in the domain of intellectual autonomy. Arguably, intellectual autonomy is a foundational aspect of intellectual character, allowing individuals to control their own intellectual pursuits, follow autonomous lines of inquiry, and critically engage with information, rather than passively absorb it. As such, in contrast to the familiar dystopian vision of humans enslaved by their machines, perhaps human-machine proximity has the capacity to cultivate greater levels of intellectual autonomy and control.

These are just three dimensions of intellectual character that may be positively impacted by increased human-machine proximity. As noted, much more empirical and theoretical work is required in order to investigate whether, in fact, such positive impacts on individual character are found, particularly among those working with technology in an intensive and immersive way in the tech sector. There are, moreover, reasons to be cautious in overstating the potential for positive effects from increased human-machine proximity in these settings. Several authors, for example, note the significance of engaging in reflective practice alongside the development of computational thinking:

> "It was found that the blogging experience could support the development of problem-solving practices such as being incremental and iterative for university students." (Lye and Koh, 2014, p.57, citing Robertson, 2011)

Unlike students in a formal learning environment, computer programmers and others working in fast-paced start-ups or large tech corporates are unlikely to be as readily incentivized to engage in this kind of valuable reflective practice throughout the day. Similarly, Clements notes that "most studies that report gains in creativity are based on what we call a 'mediated conceptual framework hypotheses' featuring teacher mediation and a sound theoretical foundation." (Clements, 1995, p.147). Such mediation is, again, less likely to feature in the day-to-day work and lives of those working in close human-machine proximity in the tech sector. Once again, further work is required in order to establish precisely if and how the positive effects of computer-based learning in classrooms translates to technology-based work. In addition, there are also reasons to consider the potentially negative impacts of human-machine proximity on character.

III. Vices of human-machine proximity

Despite these promising correlations, there are also possibilities for moral and intellectual deterioration from life "close to the machine." Programmers, for example, may develop cognitive and perceptual patterns that are more localized and constrained, and more algorithmic and less affective. These shifts may, in turn, inhibit the cultivation of certain virtues. To take the issues over localized and constrained thinking first, this issue was identified by all three example sources in Section I: recall that Dijkstra argues that his "experiment" revealed that certain programming languages can narrow one's attention to fail to see certain obvious solutions, Tarnoff & Weigel write about a data scientist who notes that the dominant mindset among those who live "close to the machine" is to look solely to technical solutions for addressing all manner of problems, and Ullman describes various cases in which the kind of localized thinking induced by spending long hours programing narrowed an individual's moral perceptual and social landscape, affecting how one perceives (or fails to perceive) others, and the moral dimension to the work that one is doing (e.g. seeing only the technical issues in the work at hand, and not the ethical implications). These examples perhaps point to how life "close to the machine" can foster mindsets of localized and constrained thinking that hinder the development of virtues of attention, broad-based problem solving, and moral perception.

The implications for the potential hindering of virtues of attention (where they relate to social intelligence) and of moral perception (i.e. decreased sensitivity to the moral landscape) are particularly troubling in light of the fact that this may affect how these technologies are developed – engineers and programmers with insufficient sensitivity to stakeholders and to ethical consequences and considerations, and who have a narrow focus upon the technical questions alone, may well create ethically dubious and damaging technologies. In fact, a recent account of life as an engineer at one of the world's largest tech companies details the reality and urgency of this risk. He describes how there had been a push from AI ethicists within the company to train engineers in cultivating virtues of social intelligence and moral attention, so that engineers would no longer simply constrain their focus to the technical problems at hand, with no attention paid to the ethical questions or the stakeholders involved (Lemoine 2021). He details how the efforts were unsuccessful, and argues that truly fair and ethical AI will only be possible once engineers are not "given a pass" to narrow their focus to only the technical questions, but are required to develop virtues of social intelligence and moral attention that will operate as they live "close to the machine," informing and integrated into how they go about developing these technologies at every step.

Second, we turn to the issue of how life "close to the machine" may potentially result in what we might call "affective blunting," where one's patterns of thought gradually take on less of an affective dimension. Recall that Ullman describes how a common feature of programming involves long periods in which affective thinking and reactions are muted or suppressed. Such muting and suppression may well lead to affective blunting. This, however, would impede virtues of moral perception, as affect is a key component to moral perception. To see why this is so, consider how the knowledge of a red object is not exhausted by knowledge of its primary qualities (its shape, surface reflectance properties, etc.). Instead, secondary qualities that arise from these primary qualities (e.g. the phenomenal experience of redness from seeing the object) are also important to knowledge of the object. If one only had knowledge of the primary qualities, without knowledge of the secondary qualities (the experience of redness), we would take them to be missing some crucial bit of knowledge about the red object. For full knowledge, the primary qualities must be experienced through what McDowell refers to as "the whirl of organism" (1998, 60-63), that is, in a characteristically human way, which will involve secondary qualities (e.g. the phenomenal experience of redness).

Similarly, just as knowledge of the "redness" of an object is bound up in experiencing the primary qualities through "the whirl of organism," so too must we experience the primary qualities of various moral situations through the "whirl of organism," which will involve secondary qualities that are bound up with affect (Johnson 2020, 29). For example, if someone witnessed an instance of workplace discrimination, and possessed full knowledge of the primary qualities of the event (the facts of the event in question, the moral principles that were violated, the effects upon the well-being of the victim, etc.), and yet this knowledge of primary qualities was not accompanied by knowledge of secondary qualities in the form of affect (e.g. she did not experience an accompanying feeling of outrage), then her moral knowledge of the situation is incomplete – in much the same way that if someone were to perceive a red object without any phenomenal experience of the redness of the object, their knowledge would likewise be incomplete.

There is also further evidence in support of the claim that affect is required for full moral knowledge: those with deficits in their capacities for emotion processing are unable to distinguish between moral and convention violations (Blair 1995). This provides further evidence that affect is bound up with our capacity for (at least some forms of) moral knowledge. Consequently, affective blunting runs the risk of impeding one's ability to have full moral knowledge. Furthermore, insofar as affect often provides or supports the motivation to act virtuously in the moral realm, affective blunting may also reduce the likelihood that one will engage in the relevant morally virtuous behavior. For example, moral outrage over some injustice provides much of the motivational energy to work to rectify that injustice, while moral elevation at having witnessed someone perform an especially exemplary action provides the motivation to imitate and mirror such virtuous actions. Consequently, affective blunting may impede both moral knowledge and moral motivation.

There is a final concern about the characterological effects of life "close to the machine," which is that for data scientists who are working with personal data, they may come to see humans in overtly reductive terms, as clusters of data points to be influenced for the data scientist's own ends. The vast majority of data science activities in the commercial sphere are aimed at predicting and influencing user behavior. Consequently, by working closely with personal data for such ends, data scientists may have the occupational hazard of objectifying other humans as clusters of data points to be influenced, rather than as ends in themselves with rich inner lives. In fact, in addition to data science jobs, there are also other jobs which involve seemingly similar kinds of occupational hazards leading to a tendency to dehumanize or objectify others. For instance, prison personnel who carry out the death penalty have a tendency to engage in dehumanization, in order to make it easier to carry out their job (Osofsky, Bandura, & Zimbardo, 2005).

The occupational hazard for data scientists of viewing others in reductive ways (as clusters of data points to be influenced) way would constitute a failure of the proper perception of others (Bommarito, 2017), most notably of "loving attention" (Murdoch, 1970), which Murdoch describes as "…a just and loving gaze directed upon an individual reality. I believe this to be the characteristic and proper mark of the active moral agent" (p. 34). Nevertheless, one could argue that it is actually rare for data scientists to single out and look at the individual data of concrete individuals – instead, they deal with massive sets of data as a whole that contain many individuals. On this objection, there are not actually occupational hazards of improper moral perception of others, because data scientists do not typically look at concrete individuals in the course of their jobs. This objection, however, misses the point. Underlying the activities of harvesting, processing, and leveraging all of this data are other preconceptions that involve failures of loving attention toward the individuals who are involved in and implicated by the data set. For example, what is missed as one goes about harvesting personal data are stakeholder preferences and realities, which are that they often have not given their informed consent to turn their data over for these purposes, they often are unaware of the scope of the information gathered on them, and they often have little idea of the purposes for which their data could be used. If users were better informed about these questions, they may well have objections to their data being used in this way.

Furthermore, in the course of processing the data and allowing it to be leveraged to influence user thinking and behavior, another presupposition is that it is permissible to influence people in this way (e.g. seeking to mold their perception, thinking, and behavior through "dark patterns"). Virtues relating to the proper perception of others, such as loving attention, would make the data scientist reflect on the individuals who are represented by the data or will be influenced by how it is processed, and upon their preferences, what is fair to them, and what conduces to their well-being. If stakeholder consideration and engagement (practices that draw from virtues of perception and loving attention) were a part of data harvesting, data analytics, and data leveraging for tech companies, then data analysts would be occupied with virtuous perception of the individuals involved. As such, the nature of the job (and its occupational hazards) has them largely ignoring the individuals involved and operating in a way that fails to attend to their needs and preferences.

IV. Why This Matters

There are three reasons why studying and considering the characterological effects of life "close to the machine" is a particularly important and urgent task. First, digital technology is not value-neutral: it contains the agendas, commitments, preferences, and biases of whoever has developed it (or supplied the data on which it has been trained) – in short, the technologies are, to a certain extent, made in the image of their creators. The values of the creators are then baked into the technologies themselves, and then become further engrained in users (and ultimately society), as user interact with the platform and thereby come to be influenced by these values, in turn. The contention of this paper is that there are ways in which the developers of these technologies may have their own character changed by working on digital technologies. Consequently, it is crucial to better understand what these changes are, because these characterological changes will likely inform how they build these digital technologies, leading to downstream effects on users and society. In other words, we need to understand how developers and programmers are formed, so that we can better interrogate how they are going about forming the technologies on which they are working.

Secondly, digital technologies operate within an ecosystem of trust. End users need to have a certain degree of trust in the individuals and corporations who have created the technology that they are using. If life "close to the machine" can be leveraged to help form virtuous individuals, and if any potential characterological pitfalls of life "close to the machine" can be dealt with, then end users will be able to have a high level of trust in the digital technologies they use, if they know that they were created by virtuous individuals. There are, then, strong commercial incentives for tech companies to take all of this seriously. Additionally, work on "moral contagion" has revealed that individuals tend to avoid objects that have been created by or come into contact with unvirtuous individuals, because they fear (on some level) that the moral contagion could spread to them through the object (Nemeroff & Rozin 1994; Tapp & Occhipinti 2016). While this literature has so far only dealt with tangible objects, it is plausible that this effect could extend to non-tangible objects, too. Consequently, digital technologies that have been developed by unvirtuous individuals may be avoided by end users, due to a "moral contagion" reaction. If this is the case, then this is another commercial consideration for corporations to consider the (reputation) of the character of their employees.

Third, there have been an explosion of jobs in programming, software engineering, and data science, leading to increasing numbers of students majoring in (and then entering) these jobs. Additionally, because of the job prospects available through having programming skills, primary and secondary schools are also beginning to offer coursework in programming in increasing numbers. With so many people coming to spend increasingly larger portions of their lives "close to the machine," it will be vital to understand what the characterological benefits (and hazards) of these activities might be.

V. Conclusion and Next Steps

Ultimately, more empirical work on the characterological effects of life "close to the machine" is needed, in order to provide greater insight into the urgency and scope of this concern (see also Scherer, 2016 for a similar call for more empirical work in this area). The evidence for characterological effects presented in Section I, which while sparse and anecdotal is compelling and is (so the sources claim) apparently fairly widely known in the programming and engineering communities, along with the studies of the beneficial characterological effects in Section II, begin to sketch an intriguing picture that awaits a more detailed filling-in. Indeed, while there is a host of research, funding, and attention being poured into how the *use* of digital technologies may be forming the character of end users, it is crucial that we not neglect the question of how working on designing and developing digital technologies themselves forms the character of the individuals involved, since these creators in turn determine the kind of shape that those digital spaces will take.

As activities "close to the machine" take up larger portions of increasing numbers of individuals' lives, through increased job prospects and coursework in these areas, it will be crucial to know how to leverage any positive opportunities for character formation (such as those outlined in Section II), while guarding against any attendant risks (such as those outlined in Section III). Further empirical work in this area would do well to explore the relationships between the length and depth of engagement in human-machine proximity with characterological change. For instance, do the concerns expressed in Section III around how virtues of attention and moral perception can be impeded only hold for those who spend long periods of time coding or working with personal data? Might there be techniques that could be protective against any such threats to character development (e.g. taking breaks, engaging with stakeholders to reshape one's attention)? There are also pressing questions regarding whether the characterological changes brought about by human-machine proximity are domain-specific (applying only in that context), or whether it will bleed out into other areas of life, away from work on the machines. As it stands, this area of research is wide open, and poses questions of pressing philosophical and practical import.

References:

Angeli, Charoula and Giannakos, Michail. (2020) Computational thinking education: Issues and challenges. *Computers in Human Behavior.* 105

Baehr, J. (2011). *The Inquiring Mind*. Oxford: Oxford University Press.

Baehr, J. (2018). Intellectual Creativity. In *Philosophy and Creativity* edited by Gaut Berys and Kieran Matthew. London, Routledge.

Blair, R. (1995). A cognitive developmental approach to morality: Investigating the psychopath. *Cognition*, *57*, 1-29.

Bommarito (2017). *Inner Virtue*. Oxford: Oxford University Press.

Carey, L J. and Flower, L. (1989). Foundations for creativity in the writing process. In Glover, J. A., Ronning, R. R., and Reynolds, C. R. (eds.). *Handbook of Creativity*. London, New York, 283-303.

Clements, D. H. (1995). Teaching creativity with computers. *Educational. Psychology Review*. 7, 141–161.

Clements, D. H., and Merriman, S. (1988). "Componential developments in LOGO programming environments," in *Teaching and Learning Computer Programming: Multiple Research Perspectives*, ed R. E. Mayer (Hillsdale: Lawrence Erlbaum Associates, Inc.), 13–54.

Denning, P. J. (2010). Great principles of computing. *American Scientist*. 98, 369–372.

Dijkstra, E. W. (1972). The Humble Programmer. ACM Turing Lecture [Transcript].

Dijkstra, E. W. (1975/1982). "How do we tell truths that might hurt?" In *Selected Writings on Computing: A Personal Perspective* (pp. 129-131). New York: Springer-Verlag.

Dyck, J. L., and Mayer, R. E. (1989). Teaching for transfer of computer program comprehension skill. *Journal of Educational Psychology*. 81, 16–24.

Gaut, B. (2014). Mixed Motivations: Creativity as a Virtue. *Royal Institute of Philosophy Supplement*, 75, 183-202.

Horton, J. and Ryba, K. (1986). Assessing Learning with Logo: A Pilot Study. *Computing Teacher*, 14(1), 24-28.

Johnson, M. K. (2019). Imaginative simulation in the moral life (Doctoral dissertation). University of Cambridge.

Kieran, Matthew. (2014). Creativity as a Virtue of Character in Paul, Elliot Samuel and Kaufman, Scott Barry (eds.), *The Philosophy of Creativity: New Essays*. New York: Oxford University Press, 125-146.

Lemoine, B. (2021). The history of ethical AI at Google. *Medium*. https://cajundiscordian.medium.com/the-history-of-ethical-ai-at-google-d2f997985233

Lye, S. Y., and Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: what is next for K-12? *Computer Human Behavior*. 41, 51–61.

McDowell, J. (1998). *Mind, value, & reality*. Cambridge, MA: Harvard University Press.

Murdoch, Iris. (1970). *The Sovereignty of Good*. London: Routledge.

Nemeroff, C., & Rozin, P. (1994). The contagion concept in adult thinking in the United States: Transmission of germs and of interpersonal influence. *Ethos*, *22*(2), 158–186.

Osofsky, M. J., Bandura, A. & Zimbardo, P. G. (2005). The role of moral disengagement in the execution process. *Law and Human Behavior*, *29*(4), 371-393.

Overmann, K. A. & Wynn, T. (2019). On tools and making minds: An Archaeological perspective on human cognitive evolution. *Journal of Cognition and Culture*, *19*, 39-58.

Pardamean, B., Suparyanto, T., and Evelyn. (2015). Improving problem-solving skills through Logo programming language. *New Education Review*. 41, 52–64.

Scherer, R. (2016) Learning from the Past–The Need for Empirical Evidence on the Transfer Effects of Computer Programming Skills. *Frontiers in Psychology*. 7, 1390. doi: 10.3389/fpsyg.2016.01390

Tapp, C., & Occhipinti, S. (2016). The essence of crime: Contagious transmission from those who have committed moral transgressions. *British Journal of Social Psychology*, *55*(4), 756–772.

Tarnoff, B. & Weigel, M. (2020). Voices from the Valley: Tech workers talk about what they do – and how they do it. New York: Farrar, Straus and Giroux.

Taylor, A. H. & Gray, R. D. (2014). Is there a link between the crafting of tools and the evolution of cognition? *WIREs Cognitive Science*, *5*, 693-703.

Ullman, Ellen. (1997) *Close to the Machine: Technophilia and its Discontents*. New York: Picador.

Zagzebski, Linda. (1996). *Virtues of the Mind: An Inquiry into the Nature of Virtue and the Ethical Foundations of Knowledge.* Cambridge: Cambridge University Press.