AI & SOCIETY (2023) 38:1659–1683
https://doi.org/10.1007/s00146-021-01328-4

OPEN FORUM

Studying human-to-computer bias transference

Johanna Johansen[1] · Tore Pedersen[2,3] · Christian Johansen[4]

Received: 2 June 2021 / Accepted: 22 November 2021 / Published online: 27 December 2021

Abstract

It is generally agreed that one origin of machine bias is resulting from characteristics within the dataset on which the algorithms are trained, i.e., the data does not warrant a generalized inference. We, however, hypothesize that a different 'mechanism' may also be responsible for machine bias, namely that biases may originate from (i) the programmers' cultural background, including education or line of work, or (ii) the contextual programming environment, including software requirements or developer tools. Combining an experimental and comparative design, we study the effects of cultural and contextual metaphors, and test whether each of these are 'transferred' from the programmer to the program, thus constituting a machine bias. Our results show that (i) cultural metaphors influence the programmer's choices and (ii) contextual metaphors induced through priming can be used to moderate or exacerbate the effects of the cultural metaphors. Our studies are purposely performed with users of varying educational backgrounds and programming skills stretching from novice to proficient.

Keywords Biases · Programmers · AI · Cultural background · Metaphors · Priming · Randomized controlled trial

1 Introduction

Biases are often difficult to study because of the complex thinking 'machinery' that makes up the human brain and because of the human's interaction with its complex social environment. Moreover, people are usually unaware of their own biases and they may even be prone to rationalize their own biased tendencies. Nevertheless, by employing carefully designed experiments, both the specific psychological mechanisms that lead to biases and the occurrences of specific human biases have successfully been identified and well established (e.g. Gilovich et al. 2002; Tversky and Kahneman 1974; Oliver 2014; Wilson and Gilbert 2003).

The tendency to be biased is, however, not solely a human affair. Due to the increase of power and societal penetration of artificial intelligence (AI) algorithms, the occurrence of machine biases have also been observed and described, something which is rather counter to our intuition that machines are unbiased and objective. Because of the increased use of AI in social systems, biases in AI have a strong negative impact in society, prompting organizations such as ACM and the European Parliament to issue strong statements (ACM Policy Council 2017; STOA 2019) and prominent researchers to publish lengthy reports (Brundage et al. 2018) warning against biased AI.

The occurrence of biases in AI have been observed to appear due to biased training data on which these algorithms are built (Feldman et al. 2015; Mittelstadt et al. 2016; Caliskan et al. 2017; Silva and Kenney 2019). For example, if the limited sample-data that the algorithms are trained on are not sufficiently representative of the larger population-data that the algorithms are subsequently unleashed upon, then the algorithmic judgments would not be valid but instead biased. Or, for example, if the training data shows a strong relationship between two variables, say, ethnicity and crime (Zou and Schiebinger 2018; Dressel and Farid 2018), a bias may occur because it is assumed that the relationship is causal

✉ Johanna Johansen
jjohanna@ifi.uio.no

Tore Pedersen
tore.pedersen@oslonh.no

Christian Johansen
christian.johansen@ntnu.no

[1] Department of Informatics, University of Oslo, Blindern, P.O. Box 1080, 0316 Oslo, Norway

[2] Department of Psychology, Oslo New University College, Oslo, Norway

[3] Centre for Intelligence Studies, Norwegian Defence Intelligence School, Oslo, Norway

[4] IIK-NTNU, Norwegian University of Science of Technology, Gjøvik, Norway

when the relationship between the two variables may in fact be non-causal and instead be caused by a third variable, say, poverty.

However, the idea that the human programmer is a source of biases, has not been investigated equally wide (Baeza-Yates 2018; Silva and Kenney 2019; Cowgill et al. 2020). We argue in this paper that biases may also be transferred from the human programmer into the final artifact, i.e., the program/algorithm. Transfer (or contagion) of biases between humans is well-known, such as, for example, the conformity bias (e.g., Moscovici and Faucheux 1972). Additionally, a transfer of biases due to influence on humans from social and cultural institutions such as media or education is equally known (e.g., Bourdieu and Passeron 1977; Lakoff and Johnson 2008). Given the human cognitive tendencies (explained more below) to employ inappropriate mental judgment-modes in situations that are "uncertain", combined with influences from institutional agendas, human biases are ubiquitous. There are, however, few studies (e.g., Cowgill et al. (2020)) to support the related possibility of programmer biases being encoded, in some way, in their programming artifacts.

Contrary to making an error, which represents a single incident in which one makes an incorrect judgment, *a bias* is a systematic tendency to commit the same type of error over time or in different situations. Particular situations in which biases can appear are, e.g., when processing information that is too voluminous or too complex for the human brain to handle, or when forced to make a rapid judgment in a time-frame that is too short to review the information at hand, or when there is insufficient information for making the decision, such as in underspecified software requirements in programming. This happens because the brain's preferred cognitive mode is the automatic System 1, also termed Intuitive thinking or "fast thinking", which operates outside our conscious awareness (Kahneman 2011). In situations characterized by certainty, System 1 usually works well because we are on a "familiar terrain" where the useful mental shortcuts employed by System 1 are adaptive and functional. The problem is that our brain employs System 1 also in situations characterized by *uncertainty*, when instead it should be doing controlled and conscious cognitive processing, often termed Analytic thinking, System 2-thinking, or "slow thinking". Even if System 2 is the preferred cognitive mode for arriving at a correct judgment when the situation is in fact uncertain, this mode is not always easily attained.

Two key concepts in the employment of mental shortcuts, also termed heuristic processing, are cognitive *accessibility* (resulting from the availability heuristic) and cognitive *representativeness* (Tversky and Kahneman 1974; Gilovich et al. 2002; Thaler and Sunstein 2009). When something is easily retrievable from memory, we have a tendency to wrongfully regard it as something that is also occurring frequently, even if it is not. We may also make an incorrect judgment about an unfamiliar phenomenon by identifying superficial resemblances to a familiar phenomenon. Because it is cognitively effortful to identify substantial similarities between two phenomena, particularly in situations characterized by incomplete information and uncertainty, superficial similarities are more easily identified. In many instances this results in an incorrect judgment. Thus, situations or contexts characterized by uncertainty in one way or another, prompts an inappropriate cognitive processing in System 1-mode, where the employment of mental shortcuts leads us to arrive at an incorrect judgment, thus exhibiting a cognitive bias. The mental shortcuts employed by System 1 are heuristics in terms of being psychological mechanisms that may lead to a biased judgment under conditions of uncertainty, as with incomplete software requirements. This aspect of underspecification is what we study empirically in this paper and what our programming test scenario from Sect. 4.1 is based on.

1.1 Algorithmic bias: data or the programmer

Media as well as the general public seem to assume that machines and algorithms are neutral and objective. However, it has been known for quite some time that complex algorithms, such as those from artificial intelligence, may exhibit biases s.a.: racial bias (Schlesinger et al. 2018), gender discrimination (Zou and Schiebinger 2018) and other socially relevant types of biases (Friedman and Nissenbaum 1996; Boyd and Crawford 2012; Jobin et al. 2019), when processing information in the support of decision making (Corbett-Davies et al. 2017; Dressel and Farid 2018; Grgić-Hlača et al. 2019; Vaccaro and Waldo 2019).

This phenomenon is commonly labeled machine/algorithmic bias (Chouldechova and Roth 2020), and has been confirmed in different areas, e.g., in big data (Hajian et al. 2016), web (Baeza-Yates 2016, 2018), autonomous systems (Danks and London 2017). Among institutions that have raised concerns about the existence of "biased algorithms" are: the ACM US Public Policy Council[1]; the EU Parliament[2]; the New York City Council bill on "Accountability and transparency in algorithms for public agency support"[3];

[1] ACM U.S. Public Policy Council and ACM Europe Policy Committee (2017). Statement on Algorithmic Transparency and Accountability. https://www.acm.org/binaries/content/assets/public-policy/2017_joint_statement_algorithms.pdf.

[2] EU Parliament (2016). EU Framework on algorithmic accountability and transparency. https://www.europarl.europa.eu/doceo/document/E-8-2016-007674_EN.pdf.

[3] New York City Council (2018) A local law in relation to automated decision systems used by agencies. http://www.legistar.council.nyc.gov/LegislationDetail.aspx?ID=3137815&GUID=437A6A6D-62E1-47E2-9C42-461253F9C6D0.

ERCIM (Rauber et al. 2019); World Wide Web Foundation[4] and many more (Cath et al. 2018), joined by major publication venues such as Science and Nature (Obermeyer et al. 2019; Zou and Schiebinger 2018; Gianfrancesco et al. 2018) and by scholarly books (Boden 2008; O'Neil 2016).

These works generally look at the data that AI algorithms train on, and show how the data contains biases. However, other sources of biases have been recognized and described, although less investigated, e.g., Danks and London (2017) describe five categories of biases in AI where, apart from biased training data, it is also described how inappropriate usages of the algorithms might result in biased decisions, e.g., as when the decision system is used in a different context than where it is supposed to. Humans as a source of AI bias is mentioned in two recent articles (Baeza-Yates 2018; Silva and Kenney 2019), e.g.:

- Silva and Kenney (2019) describe nine types of biases (present at five different algorithmic stages: input, algorithmic operations, output, users, and feedback), some of which can be studied in conjunction with the general bias transfer that we demonstrate in this paper; whereas
- Baeza-Yates (2018) mentions the users and producers of the web content as sources of bias related to the data, but also points out different forms of bias originating from the user interface made by interaction designers, whom could be regarded as 'programmers'.

In the light of the results of this paper regarding the hypothesis of 'bias transfer', we consider it particularly useful to study empirically all the different forms of biases described in the works above, and especially their transference, maybe using methods similar to what we present in this paper. This is also supported by Baeza-Yates (2018) who recognizes in conclusion the same general sources of biases as we study here, i.e.: "each program probably encodes the cultural and cognitive biases of their creators", and points in the introduction "measuring bias" as a major challenge, which is what we do here.

The idea of *transference* of biases from the programmer to the programs is not studied nor empirically proven (at least not to our knowledge). The most relevant work is from Cowgill et al. (2020) who, like us, encourage the community to conduct more empirical studies of programmers to better understand biases in AI. Thus, rather than pointing once more to the problem itself, the present work provides insights into why the bias transfer phenomenon may occur. We operate within the same paradigm and with a similar agenda as those who study human behavior in multidisciplinary research themes such as Behavioral Economics (Tversky and Kahneman 1974; Kahneman et al. 1991), Behavioral Transportation Research (Pedersen et al. 2011; Gärling et al. 2014) and our own contributions termed Behavioral Artificial Intelligence (Pedersen and Johansen 2019) and Behavioural Computer Science (Pedersen et al. 2018).

In order to function as a guide, this paper purposely describes the instruments and methodology that we use for revealing bias transference.

- First, one needs to find a way to *test the subjects for biases*; in our case this is the bias revealing test from Sect. 2, whereas, e.g., Cowgill et al. (2020), though not focusing so much on testing of the subjects, do use an implicit association test to see the links that their subjects make between {*Man, Woman*} and {*Math, English*} which is appropriate for their setting.
- Second, one needs to find a way to properly *include such a bias revealing test in a programming task* (the more realistic the programming environment, the better); e.g., the setting of (Cowgill et al. 2020) is particularly appropriate because they went into a boot-camp for AI programming students.
- Third, one needs to be able to *test whether the programming artifact is also biased*, using the same bias tested on the programmer in the first stage. This is especially relevant for AI systems where one finds biases that have traditionally been found when humans take decisions (Johnson 2021), such as racial or gender.

Our study broadens these aspects in two ways, and departs somewhat from (Cowgill et al. 2020).

- First, even if the idea of *bias transference* has much relevance for AI, we make the point that it may be relevant for all types of programs/software systems; and therefore we talk in this paper quite often about the "programming artifact", thus not limiting it to only AI algorithms.
- Second, we want to open the community (both the empirical software engineering community as well as the psychology and ethics of technology communities) to more "types" of programmers, including novice and "not-so-structured" programming. This is because we see more and more software systems being built (or configured) by non-programmers, both professionals in their own fields, e.g., physicists working with image recognition libraries or software, but also non-professionals, e.g., lay people that need to setup or configure increasingly complex IoT systems in their environments (either at work places or in their homes). Therefore, our "imaginative" programming task is meant to reach this kind of people,

[4] World Wide Web Foundation (2017) Algorithmic accountability: applying the concept to different country contexts". https://webfoundation.org/docs/2017/07/Algorithms_Report_WF.pdf.

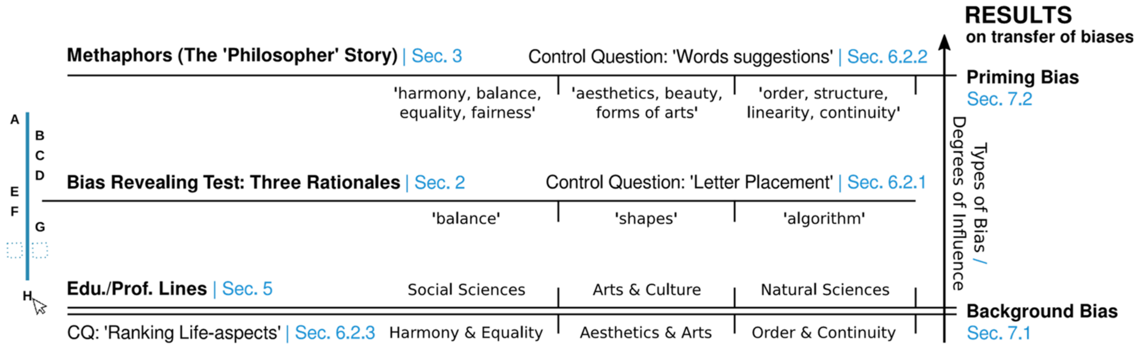

RESULTS
on transfer of biases
Methaphors (The 'Philosopher' Story) | Sec. 3
Control Question: 'Words suggestions' | Sec. 6.2.2
'harmony, balance, equality, fairness'
'aesthetics, beauty, forms of arts'
'order, structure, linearity, continuity'
Priming Bias
Sec. 7.2
A
B
C
D
E
F
G
H
Bias Revealing Test: Three Rationales | Sec. 2
Control Question: 'Letter Placement' | Sec. 6.2.1
'balance'
'shapes'
'algorithm'
Types of Bias / Degrees of Influence
Edu./Prof. Lines | Sec. 5
Social Sciences
Arts & Culture
Natural Sciences
CQ: 'Ranking Life-aspects' | Sec. 6.2.3
Harmony & Equality
Aesthetics & Arts
Order & Continuity
Background Bias
Sec. 7.1

Fig. 1 Overview of the contributions of the paper with references to relevant sections

especially because we want to test the bias transference hypothesis in these people as well.

1.2 Two main questions

The present study investigates methodically, experimentally, and empirically the hypothesis of bias *transfer* in programming, providing a convincing argument for inspiring more empirical studies to be taken in the same direction. As such, in this study our main focus is to find support for (or against) the hypothesis that people may unknowingly and inadvertently transfer their biases to the computer programs that they build. However, we do not study specific biases (s.a., gender or racial), nor do we test or suggest specific programming methods and tools that could counteract bias transfers. This would be a task for future research. Although some may argue that robust quality assurance procedures eliminate any instances of biases in algorithms, at least in professional programming environments, we leave out for now testing whether the quality assurance procedures themselves have inherent biases or miss some forms of biases, and focus only on showing that *programmers may be a source of biases*, apart from the data given to the program. We set out to investigate the following:

I. *Are biases being transferred from the human programmer to the program artifact?*

This is studied in a basic form with a bias revealing test that we detail in Sect. 2, which we impose on the subjects of our study, as described in Sect. 4. The biases that we study are of both cultural and contextual nature.

II. *Can programmers be manipulated, i.e., primed by inducing a new bias, and is this new bias then transferred to the program?*

In Sect. 3 we describe the methods that we use for priming our study subjects towards the same biases studied for the first question. Subsequent sections then describe how we used the priming in our studies and their outcomes.

Because heuristic thinking is seen as the main psychological "engine" for generating cognitive biases, our experiments will also employ a heuristic approach, that is, relying on mental shortcuts such as "accessibility/availability" when inducing a bias on the participants in the study.

1.3 Contributions

The contributions of this paper are presented schematically in Fig. 1. The vertical axis of the diagram represents types of biases (or degrees of influence) with the arrow indicating a direction from more general, long-term, and strong biases to more contextual, short-lived, and weak forms of biases. We study two extremes, namely biases originating from the background of a person (including socio-cultural, educational, and professional influences), and biases resulting from manipulations such as priming. However, in between, one may study other forms of biases originating, e.g., from occupational cultures (Blackwell et al. 2019) (e.g., programmers working for Google compared to a startup), or from media and propaganda. These biases form over a considerable period, e.g., several years, which is shorter than for cultural biases, but longer than minutes as it is the case with priming.

Our results will be presented in Sect. 7, showing how these two types of biases are being transferred from humans to programs, providing evidence for the hypotheses that transfer of both cultural and contextual biases exists. To support this claim of transference, the rest of the paper is devoted to the development of our studies and analysis of the data.

We study users with different programming skill levels, i.e., from professional to amateur (Markopoulos et al. 2017; Paternò and Santoro 2019). This is motivated by the observation that increasingly more lay people (wrt. programming) are interacting and designing rather complex systems (Manca et al. 2019). Nowadays it is not only expert developers that program, but people with all levels of expertise carry out various programming-like tasks, from simple configurations of IoT systems in their smart home (Ur et al. 2016;

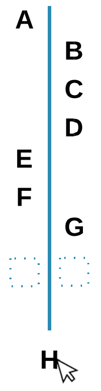
Markopoulos et al. 2017; Brich et al. 2017), to more complex installation and management of technology systems in their work, to more unconventional forms of programming using visual languages (Erwig et al. 2017; Akiki et al. 2017) (such as Fraunhofer's IoT Programming Language NEPO[5] or Google's Blockly) or domain specific languages, and even assembling ready-programmed components into a final software system as done, e.g., in the IBM's IoT development environment.[6] This is because of the proliferation of simple (abstract, graphical, etc.) programming languages and interfaces aimed at non-programmers to design domain specific information systems, e.g.: a biologist programming a DNA search or an oil-engineer programming a complex database search. Therefore, in our study we use a simple programming task presented as fictitious, i.e., a proxy task, where the participants are imagining that they are programming.

Thus, the first important element of our study is presented in Sect. 2 where we develop a bias revealing cognitive task, which can be used for both types of biases. This cognitive task allows respondents to answer only with one of the three rationales that are listed on the middle line of the diagram. In Sect. 6.2.1 we analyze how well our test worked, using one of our control questions.

Section 3 details the manipulation method that we used, involving priming participants with metaphors hidden in a fictitious 'Philosopher' story. How well these manipulations worked is studied again with a control question involving listing of 'similar words' in Sect. 6.2.2. We created three metaphors to match the three rationales, which in turn match with three kinds of educational lines (or views on life). This correspondence is reflected in the vertical alignment of these elements in the diagram. With this we study the influence of contextual metaphors, in addition to the cultural bias.

Thus, since we aim to investigate whether *inducing* a bias is effective it is important, to avoid any intrinsic debiasing, to "hide" from the subjects the real goal of the study behind a seemingly unrelated goal; in our case we used the title "Study of natural language in programming". This is a standard study approach in research on biases because many types of biases can also be experimentally induced using priming (Tulving and Schacter 1990; Yonelinas 2002). Once we have established in this paper whether or not priming also works in the setting of programming biases, future research can carry out more detailed studies about whether such priming already exists "out there" (intentionally or not) and what types of priming would work and to what degree.

Fig. 2 The puzzle game



We have thus chosen our participants to represent the three different backgrounds detailed in Sect. 5. To test the assumptions about our participants' backgrounds we used one control question (analyzed in Sect. 6.2.3) asking them to rank the three 'life-aspects' listed on the bottom line of the diagram.

We thus also investigate whether people educated in programming exhibit less biases and are less prone to manipulation. Moreover, we also aim to study whether it is possible to experimentally induce a bias on this category of users, or if this particular category is more resistant to priming and bias-transfer to programs.

The rest of the paper is devoted to presenting in Sect. 4 the major phases of designing our surveys and our studies using usability testing, and analyzing the data and demographics, in Sect. 6.

Our present work is motivated by the need to prove or disprove the idea that human biases could be transferable to the programming artifacts. However, which types of biases and how 'dangerous' these might be are not the subject of this study. Other specific studies would have to be devised, maybe similar to the research on human biases developed in the psychology field.

2 A bias revealing cognitive task

Over the years, we have used a simple cognitive task (Townsend 2003) (originally called Alice's Alphabet Puzzle) in lectures on judgment and decision making. In Townsend's book, this particular puzzle lists the letters on a horizontal line, where straight lined letters are placed above and curved lined letters are placed below the line. We changed the puzzle in a vertical position to trigger more the infinity of the line, and the balance of the sides. In the task, the audience is first shown, as in Fig. 2, a sequence of the letters divided by a vertical line and then asked to decide on which side of

[5] "Fraunhofer IAIS IoT Programming Language NEPO Roberta Lab" by Thorsten Leimbach and Daria Tomala. In ERCIM News 120, January 2020, Special theme: Educational Technology. https://ercim-news.ercim.eu/en120/special/fraunhofer-iais-iot-programming-language-nepo-in-the-open-roberta-lab.

[6] IBM Watson IoT Platform https://www.ibm.com/internet-of-things/solutions/iot-platform/watson-iot-platform.

the vertical line the next letter H should be placed and *why* it should be there.

When asked *why*, the respondents provide rationales that seem to fall into three categories, which we categorized as: (I) 'balance', (II) 'shapes', or (III) 'algorithm'. The quintessence of their arguments are as follows:

I. Some argue that there should be an equal number of letters on each side of the line: since there are already four letters on the right side and only three on the left side, the next letter, H, should go on the left side, thus indicating a sense of 'balance'.
II. Others argue that the straight-lined letters A, E, and F are on the left side of the line, whereas the curved-shaped letters B, C, D, and G are on the right side, something which makes it perfectly reasonable that H should be together with its "kin" on the left side, given the different 'shapes' of the letters.
III. Yet others argue that there is an inherent order (or pattern) in the sequence: e.g., some indicate the sequence Left-1, Right-3, Left-2, Right-4, thus suggest placing H on the right side due to perceiving the image/puzzle as having the characteristics of an 'algorithm/pattern'.

This exercise is simple enough to reveal cognitive tendencies of System 1, instead of consciously engaging the System 2 analytical thinking, as usually done by more complicated tasks. More importantly, the bias cannot be avoided because there is nothing else in the picture to help the person when making the decision, and any placement is correct; therefore only something from either (i) the background of the subject, alternatively, (ii) an experimentally induced prime, could help with making the decision—or random choice.

3 Metaphors as priming method

We use the above task to 'forcefully' reveal a bias, albeit an innocent one (compared to racial or gender), which would have its origin in the cultural background of the person (e.g., education, line of work, hobbies). This will be used to test our first main hypothesis, namely that cultural metaphors would influence the programmers' choices. To test our second hypothesis, we want to prime the subjects to non-consciously make a decision in one specific "direction", namely towards one of the three rationales that we identified in Sect. 2.

Our motivation for hypothesizing that programmers would non-consciously be affected by the prime comes from the well-known effect of cognitive heuristics (Gilovich et al. 2002). As mentioned previously, under conditions of uncertainty, where one does not *know*, but nevertheless has to make a judgment or a choice, one will non-consciously base one's judgments either on instances that spring easily to mind (i.e., the cultural background or the contextual prime triggers the availability heuristic), or on instances that resemble the current problem (triggering the representatives heuristic). The judgment can also be made as an approximation to the most recent, the most related or the most relevant information (anchoring heuristic). Thus, as regards cognitive processing, heuristic thinking in System 1 mode is very much an associative reasoning mode influenced by cognitive availability and perceived representativeness. However, one needs to also consider the *content* of the heuristic processing mode, e.g., what *is* actually easily accessible in memory. In terms of content in such associative reasoning, the metaphors and metaphorical thinking are strong sources of influence on how we as humans view the world.

The essence of a metaphor is, according to Lakoff and Johnson (2008), simply that "we understand and experience one kind of thing in terms of another". For example, an argument may be understood and experienced in terms of the metaphor *war*, where we may "attack weak points in others' arguments", we may "shoot down" others' arguments, and we may "win or lose" arguments. In fact, metaphors are so pervasive and ubiquitous in our lives that we simply cannot do without them.

Metaphorical thinking is something that can even be manipulated, e.g., Thibodeau and Boroditsky (2011) study of the effect of metaphors on preferences for crime-prevention measures. In an experiment carried out to test the effect of metaphors on these preferences, the authors 'reported' crime-rates in a fictitious city. Crime was either described in terms of "a beast" or in terms of "a virus". When exposed to the metaphor *crime is a beast*, the general public argued for harsher and more severe crime-preventing measures than what was the case when they were exposed to the metaphor *crime is a virus*. This can make us alter our view of the world, and most of the time we are not aware, neither of the fact that we think metaphorically, nor that our metaphorical thinking can be manipulated by governments, media, our employers, or others, either for commercial or political purposes.

3.1 Experimental manipulation using metaphors

Our experimental manipulation is in the form of 'a story about a philosopher who invented a puzzle', in which we vary the embedment of a different 'life-aspect', i.e., forming three different versions of the story. We also had one control condition, i.e., the story without any life-aspects (no metaphor), intended for a comparison to the experimental groups. The three different life-aspects are:

A. harmony and equality
B. aesthetics and arts
C. order and continuity

The metaphors include four words, placed in two groups, two words in the beginning of the story, and the other in the end, following indications from relevant literature (Lakoff and Johnson 2008; Thibodeau and Boroditsky 2011). The words are:

A. *harmony and balance;* then *equality and fairness*
B. *aesthetics and beauty;* then *forms of arts*
C. *order and structure;* then *linearity and continuity*

We hypothesized that each of the above life-aspects would metaphorically influence the participants in the respective group A/B/C to provide an explanation that could be interpreted as one of the rationales from Sect. 2, respectively *rationale I/II/III* (i.e., 'balance', 'shapes', 'algorithm').

The metaphorical primes were embedded in the following fictional brief story about the philosopher who was presented as the one who originally created the puzzle from Sect. 2. Each subject will read a story that differs only in the words shown inside square brackets below.

> "A philosopher who lived a life filled with [harmony and balance | aesthetics and beauty | order and continuity] created the riddle used in the game that we ask you to imagine that you program on the next page. Although the philosopher is nearly forgotten today, we know that the philosopher influenced many contemporary philosophers' view of the world. The most prominent influence seems to have been the importance of maintaining [equality and fairness | forms of arts | linearity and continuity] in life and in society."

4 Designing the studies

First, we incorporate the two instruments described in the previous sections into a programming task. We use a "paper-task" (described in Sect. 4.1) where the subject imagines to be programming, so that we can easily involve non-programmers, since part of our hypothesis is that people with various backgrounds (outside computer science) are involved in various "types of programming".

We then incorporate the programming task into the survey described in Sect. 4.2. This contains additional questions to collect information for different purposes, e.g.: identifying 'unserious subjects', i.e., subjects that did not pay sufficient attention to the task, but instead responded randomly; or for helping with the interpretation of the subjects' explanations of their rationales and their background.

We carried out our work in two stages. First we performed pilot studies, which we used to make improvements to the design. In particular, we first carried out specific usability testing in one pilot survey (described in Sect. 4.3). Then we improved the survey by using eye tracking technology to make sure that the priming is being read and to see more of how the subjects would interact with the survey (see Sect. 4.4).

4.1 The programming task

We designed a fictitious programming task in which actual programming was not undertaken during the session, but where the focus was on the subject's *reasoning* about the programming task. Thus, we informed the subject that she should *imagine* herself in the *role* of a programmer. This type of experiment is conventional within judgment and decision making, i.e., such experiments are framed as scenarios and 'imagine that you are' type of tasks, because it is often difficult to conduct 'real life' experiments with 'real' tasks. Although such scenarios may lack the degree of ecological validity that a real life experiment has, the use of scenarios and 'imagine that' is in many instances a first approximation to study the same phenomenon in a real life context at a later stage, if possible. Therefore, our study can be considered as a first step to establish whether it is useful to explore further the phenomenon of bias transference. There are, however, obvious limits to how far we can stretch our conclusions—although the same limitations exist in all of these types of studies.

The task was to 'program a game for children' where the image from Fig. 2 would be the game board. The game would consist of the player (which would be different from the subject/programmer) having to place the next letter H into one of the two designated empty boxes. Upon correct placement, the game (i.e., the programmer) would reward the player. The design of the boxes was purposely made in order for the game to be perceived as continuing downwards. This was done to reduce the risk of being confounded by unintended biases (i.e., to avoid the subject perceiving the game board as finite, with letter H being the last one).

The task description text can be seen as the "requirements" that programmers receive from their clients (or elicited during a requirements engineering process); sometimes these include, so called, "user stories", which are realistic descriptions of the functionality of the software in terms of how a user (in our case, a player) would interact/work (in our case, play) with the software (in our case, the game). Our requirements contain one major intended omission (i.e., it is incomplete) in that it does not say what would constitute a "correct" placement of the letter H. In consequence, the subjects need to decide for themselves to which side of the line they should give the reward. We hypothesized that the uncertainty inherent in the task would elicit heuristic thinking prompted by either cultural or contextual metaphors.

To introduce the priming metaphor, the game board image was linked to the story of the philosopher by saying that this “puzzle” was created by the philosopher. This link was made after the pilot testing (see details in the respective subsection below). We hypothesized that, if the participants were offered a simple explanation of the origins of the puzzle, then the philosopher story, containing the primes, would prompt the subject to non-consciously choose an explanation similar to the inherent rationale in the respective prime.

The task description that we used is the following (see Johansen et al. 2020, Appendix A) for exact layout).

> “First, spend one minute imagining how you would be programming the simple task below. Then proceed to answer the following questions.
> Imagine that you are a non-expert programmer who is developing a simple puzzle game. The game is based on a riddle made by the philosopher that you read about previously. Imagine that you have already drawn the game board that you can see below:
> [The image from Fig. 2]
> Now you are going to program the player’s interaction with the game.
> The player (not you, you are the programmer of the game) has to solve the puzzle by drag-and-dropping the letter H on to one of the two dotted boxes. The player is rewarded if the program accepts the placement of the letter H as the correct placement.”

4.2 The survey

The survey is created in SurveyMonkey,[7] bilingual, the Norwegian respondents having the possibility to choose between English and Norwegian. Screenshots of all the pages of the survey are given in (Johansen et al. 2020, Appendix A).

‘Page 1: Introductory text’: presents the goal of the survey and how the data is going to be dealt with. The goal of the experiment is only partially disclosed, and the true hypothesis remains completely undisclosed. Since the respondents have various backgrounds, other than computer science, it was also important to mention that no prior knowledge of computer programming is required for taking part in the survey/task/exercise.

‘Page 2: Instructions’: contains information that we consider important for the respondents to know before starting the survey:

> “The back button is disabled. You will not be able to go back to a previous question, so we ask you to read each question carefully, because some depend on the previous ones.
> Please put effort into reading carefully everything on each page.”

Note that some text is being emphasized, in the case of skim-reading. We need the participants to actually read the texts in the survey for the primes to work and for understanding the requirements in the programming task. For the mTurk and SurveyMonkey respondents, who were paid, we also added information about required minimum time for completion (average completion time was 6 min).

‘Page 3: Philosopher story’: contains our story intended for priming, which we have detailed in Sect. 3. We experienced during the pilot tests that the participants might not read a text if the information there cannot be used for answering questions in the survey. Therefore, we added one question meant as extra motivation (see Fig. 3).

‘Page 4: Programming task’: contains the text from the previous Sect. 4.1. We had three questions on this page, only one of these being important for the study, i.e., it asks about the placement of the letter H. To conceal the importance of this question we added two more questions completely irrelevant for our experiment. However, all three questions are made to look like questions that concern the programming task, i.e., it makes the task more realistic. If we would have left only the question about the choice of placement then the subject could have observed the missing information in the requirements that we gave and thus perceived the task as less realistic.

‘Page 5: Self explanation of choice’ and ‘Page 6: Alternative explanations’: where the respondents give, respectively, choose, an explanation for the choices they made in the programming task. We detail these two pages in the next subsections.

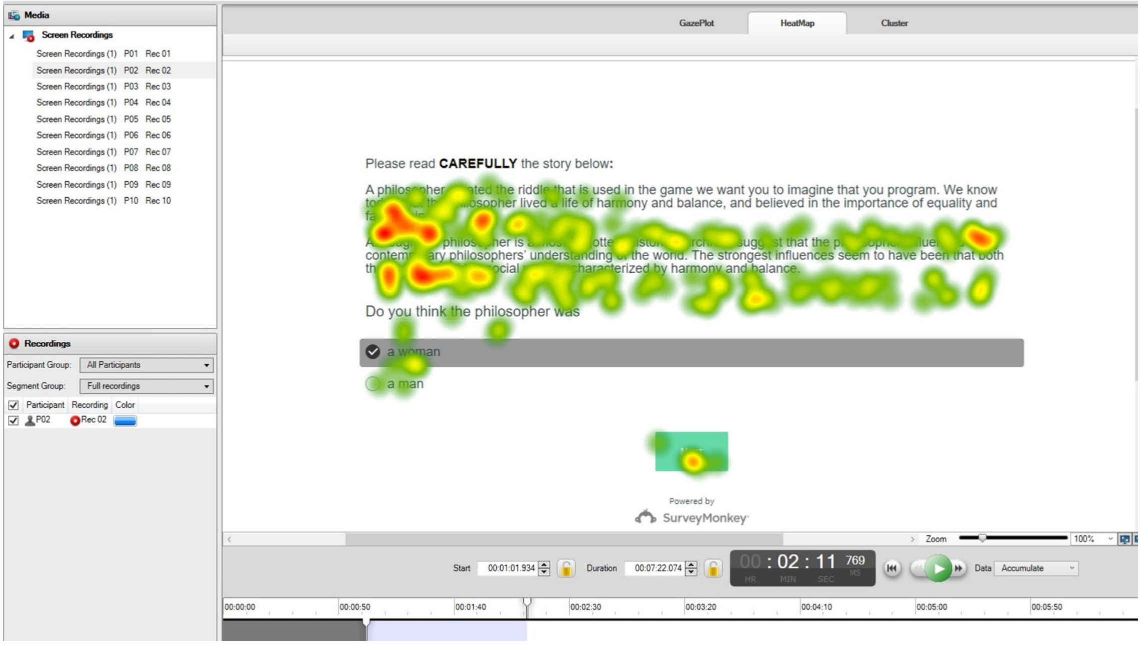
The rest of the questions on the following pages are meant to gather more information that could influence the results of the experiment, i.e., one’s view on life, hobbies, educational background, and demographics (age, gender).

‘Pages 7: Ranking life-aspects’: where the three alternatives from Sect. 3 could be ranked.

> “Please rank the following three pairs of life-aspects in the way that best reflects how you view life yourself (where 1 is the highest while 3 is the lowest).
> [Options: harmony and equality | aesthetics and arts | order and continuity.]”

This is a form of self-evaluation, where the subjects express directly their order of preference for the three instances of priming metaphors (this is done after they have completed the main task, and they are not aware that they were themselves randomly exposed to one of the metaphors). If they rank the prime that they were exposed

[7] Platform for creating online surveys: https://www.surveymonkey.com.


Media
Screen Recordings
Screen Recordings (1) P01 Rec 01
Screen Recordings (1) P02 Rec 02
Screen Recordings (1) P03 Rec 03
Screen Recordings (1) P04 Rec 04
Screen Recordings (1) P05 Rec 05
Screen Recordings (1) P06 Rec 06
Screen Recordings (1) P07 Rec 07
Screen Recordings (1) P08 Rec 08
Screen Recordings (1) P09 Rec 09
Screen Recordings (1) P10 Rec 10
Recordings
GazePlot
HeatMap
Cluster
Powered by
SurveyMonkey
Zoom
100%
00 : 02 : 11 769
HR MIN SEC MS
00:00:00
00:00:50
00:01:40
00:02:30
00:03:20
00:04:10
00:05:00
00:05:50

Participant Group: All Participants

Segment Group: Full recordings

☑ Participant	Recording	Color
☑ P02	Rec 02	🟦

Start 00:01:01.934

Duration 00:07:22.074

Data Accumulate

Please read **CAREFULLY** the story below:

A philos[illegible]pher [illegible]ated the riddle that is used in the game we want you to imagine that you program. We know to[illegible] th[illegible]losopher lived a life of harmony and balance, and believed in the importance of equality and fa[illegible]

A[illegible] philos[illegible]her is a[illegible] otte[illegible] istor[illegible] rch[illegible] sugg[illegible]st that the p[illegible] ophe[illegible] luen[illegible] contem[illegible]ary philosophers' understanding [illegible] the world. The strongest influences seem to have been that both th[illegible] ocial [illegible] characterized by harmony and balance.

Do you think the philosopher was

☑ a woman

○ a man

Fig. 3 Heat map with offset gaze

to highest, this might indicate that the prime has had an influence. The UI for ranking questions is made well by SurveyMonkey so that when the question is required, then the subject must indeed provide a ranking, and not just leave the default.

'Pages 8: Words suggestions': where the subjects could suggest one to three words characterizing each of the three life-aspects, from the "Ranking life-aspects" question. The open-ended format chosen for this page has several reasons.

- We wanted to have a way to identify unserious subjects or robot-generated answers (as detailed later in Sect. 5.3).
- We also wanted to have another way to check the metaphorical priming effect by looking whether, and how often, our priming words appear among the answers (see details in Sect. 6.2.2).
- We also wanted to gather more data for future studies; i.e., others could use some of these words in future metaphor studies.

'Pages 9: Demographics': where the subjects had 5–7 questions about age, gender, years of education, field of study, and leisure activities.

In the following section we explain the reasoning behind the way the questions are composed, as a result of the discoveries we have made during the pilot testing.

4.3 Pilot testing for usability

To improve the usability of the survey we performed several pilot tests. The first pilot test used the method of usability testing (Dumas and Redish 1999), with our survey being the product under test. One goal with testing the survey for its usability is to see whether the explanatory texts, requirements and questions are written in a clear and easy-to-understand language. Moreover, since we intended to prime the subjects, we needed to make certain that the story of the philosopher was read carefully and not just skimmed through.

The usability study (see more details in Johansen et al. 2020) involved five participants. Four of these participants have a background in computer science and one in arts and design. Subjects with these two types of backgrounds were going to be used in our full-scale experiments as well.

The participants were asked to take the survey while being observed by us, sitting next to them (one of us took the role of a moderator, while another researcher was only an observer). The test was run with one participant at the time. Before taking the survey, the subjects were explained verbally the purpose with the test session, which was to help us improve the face-value quality of the survey, although the hypothesis was not revealed. They were also presented the order of the tasks: first they were to take the survey, without any interruption, and then were supposed to answer questions meant to elicit suggestions for improvements of the survey.

The test also helped with validating our initial decision of disabling the back button in the survey. For the priming to work, the participants should not realize the connection between their choice in the programming task and the *'Philosopher story'*. If the participants would understand at a later stage in the survey that such a connection existed, they should not be allowed to navigate back and read the 'Philosopher story' again. In this pilot study one of the participants had the back button purposely left enabled. This participant did just what we expected, s/he navigated back to the 'Philosopher story', read it again, and adjusted her/his answers to reflect the view of the philosopher and not her/his own as the question required.

4.4 Eye tracking for better insights

A more exact way to reveal the behavior of the participants when reading the information, and which flow they follow, is by using eye tracker technology (Bojko 2013). We created two versions of the *'Philosopher story'*—a 'Short story' and a 'Long story'. We employed summative research[8] in combination with eye tracking methods for comparing and deciding which version was more effective.

The test was done in a usability laboratory set up with eye tracking equipment. We used a combination of single-subject and between-subject design, where each participant (ten in total) was exposed to only one of the test stimuli, so to avoid any carryover effects between the stories. Both stories contained the same priming words. The 'Long story' was created with the purpose of helping the reader to immerse in the story—by giving more background information on the philosopher—and preparing the participant for the *'Programming task'*. Since we intended to prime the subject, we needed to hide the priming words well in the story, so that unintended debiasing (e.g., reactance) would not occur. At the same time, a too long story could make the subject not read the whole text and thus possibly skip the priming words. A shorter version of the story would also reduce the cognitive burden on the subject. The eye tracking testing was thus meant to help us identify whether the subjects skip our priming words, and also how much cognitive effort (i.e., how much time) they puts into reading the stories.

The heatmaps and gaze plots visualizations[9] provided both spatial and temporal insight into how the participants interacted with the text on each page of our survey. We obtained information about which areas of the text were fixated and for how long, the number of fixations and the order in which the fixations occurred.

Interpreting this data we concluded that there was no noticeable difference in how the text, and especially the priming words, were read between the long and short version of the story. For both cases, the participants read the text thoroughly, line by line (Fig. 3). This shows that the instruction on the *'Philosopher story'* page about reading the story "carefully" had the wished effect. The difference in reading the long story in 1:10 min compared to 40 s for the short one, meant a reduction of ca. 50% in cognitive load and time, and thus we decided to use the 'Short story' in our full-scale studies.

Another aspect that we analyzed with the help of eye tracking was whether the question about the philosopher being a man or a woman works as extra motivation for the participants to read the story. We found out that in order to answer this question, the participants returned to reading the story several times. In addition to the motivational aspect, questions such as this one help in drawing potential attention away from our true hypotheses. More aspects that we investigated with eye tracking are detailed in Johansen et al. (2020, Sec. 5.4).

5 Methods

5.1 The participants

The participants were chosen based on their educational or occupational background, to span three main domains. This is meant to cover well different computer programming skill levels as well as socio-cultural influences, properties and preferences. We reason that, when enrolled in a certain university study line or field of work, people have already developed predominant skills and characteristics needed for the specific education or occupation.

We had three main cohorts of respondents, totaling ca. 300 respondents:

A. *'Social sciences' cohort*—composed of students studying psychology;
B. *'Natural sciences' cohort*—composed of students studying computer science; and
C. *'Arts and Culture' cohort*—composed of a group of participants working in the field of arts and design, a group of students studying theatre, and another group studying music.

This categorization based on the educational and professional background is confirmed by the analysis of the data obtained from the control question on the *'Demographics'* page, specifically about which field of study or/and line of

[8] Summative research implies comparing an interface or product to its other versions, competitors, or benchmarks (Bojko 2013).

[9] The gaze was offset vertically by approximately one line. This was due to the mismatch between the Operating System version and the version of Eye tracking software at the time of testing. The offset has been consistent across the participants and did not affect our interpretations.

work the respondents affirm their background to be mainly consistent with. (See Johansen et al. 2020, Sec. 7.2 for details.)

Based on the conditions to which the respondents were exposed, we also categorize the three cohorts into:

I. 'helped' and 'confined',
II. 'helped' and 'not confined',
III. 'not helped' and 'not confined'.

The 'not confined' respondents took the survey in the environment of their choice, which was unknown to us, whereas 'confined' means taking the survey in a more controlled environment (i.e., the university auditorium). In the second full-scale study we introduced an extra page in the survey, offering such alternative explanations with possible answers to choose from, meant to reduce the number of uninterpretable answers. The '(not) helped' classification refers to whether the respondents were (not) given alternative explanations to pick from, regarding their choice for the placement of the letter 'H'. The 'Social sciences' cohort belongs to the category (III), as the survey they were given did not contain the 'Alternative explanations' page and they could take the survey at the time and place of their choice. The mTurk and SurveyMonkey respondents from the 'Arts and Culture' cohort were helped with 'Alternative explanations' and were free to choose the environment where to take the survey. The 'Cultural studies' students were also 'helped' but confined to a classroom, where the course leader and one of the authors were also present. The 'Natural sciences' cohort was both 'helped' and 'confined' as the survey was taken as part of their regular course-work. The 'confined'/'not confined' and 'helped'/'not helped' are categories used for analyzing the sensical vs. nonsensical data in Sect. 6.1.

The environment of the participants and the support they received is related to three main types of environment where (future) programming activities can take place in:

A. *Typical professional programming environment*, where the programmer is 'confined' to an office space and has to her disposal all the professional resources necessary to fulfill her tasks. In our case, for the programming task and the required explanations, we tried to reproduce this type of environment for the group of computer science students, by both confining them to the classroom and course hours, and offering them helping answers.
B. *Semi-professional environment*, where an expert in some technical field (other than programming, e.g., railway engineering) has professional tool support for simple programming/configuration, e.g., by using a GUI based programming tool or a graphical programming language. However, programming is not their main task or responsibility and thus are not supposed to put too much effort into it, which we consider as 'not confined'. The mTurk and SurveyMonkey respondents were thus 'not confined' but 'helped'.
C. *Non-professional environment*, where people, e.g., in their homes, are configuring an IoT system without any professional support nor prior knowledge. The *'Social sciences'* respondents were neither 'helped' nor 'confined', and can thus be seen to some extent as fitting this profile.

For the purpose of studying the influence of the priming, we further group the respondents from each cohort by the metaphor they have been exposed to (or not), according to Sect. 3.1:

A. a control group which is not primed in any way,
B. a group primed as in Sect. 3.1.A (which we call, 'primed with harmony and equality'),
C. a group primed as in Section Sect. 3.1.B (i.e., with 'aesthetics and arts'), and
D. a group primed as in Sect. 3.1.C (i.e., with 'order and continuity').

The control group is meant to serve as a baseline to observe what the programmers' preferences for task-solutions are in the absence of primes. This is relevant for our first main question.

The three primed groups are meant to help us test whether the bias can be induced upon the programmer, and subsequently transferred from the programmer to the algorithms.

The cohort with students from the computer science study line is also meant to help us test whether programmers shut away the other two biases, except the pattern/infinite way of thinking, which is sometimes assumed that the programmers do.

5.2 Methods employed

The studies employ a combination of experimental design and comparative design. In the analyses of both (i) the comparative aspect, i.e., differences *between* the three cohorts, and (ii) the experimental aspect, i.e., differences *within* each cohort, resulting from the experimental manipulation, we employed both (a) inferential statistics, more specifically chi-square analyses of categorical data, as well as (b) descriptive statistics to report frequencies and percentages. We performed an experiment on each cohort, as well as compared the three cohorts to each other, regardless of the experimental manipulation. Since the three cohorts were different in terms of cultural and educational background, we were able to study the unique effect of background per se.

Conforming to the true experimental design method (Lazar et al. 2017; Cook et al. 2002), we first assigned the participants of each cohort randomly to one of three *experimental conditions* where we induced one specific type of contextual metaphorical thinking in each, or to a *control condition* containing neither of the three primes. The control condition contained the neutral non-prime story and was meant to serve as a "baseline" to establish whether the participants, without being primed, were inclined to favor one of the three "rationales" over the other.

The subjects are given the programming exercise described in Sect. 4.1. The programming task, the educational/professional background of the subjects, and the story containing the primes, are the *independent variables* in our experiment. The choice of what will be the right solution for the puzzle is the *dependent variable*. We are interested in finding out if the primes and the background of the participants (the independent variables) influence how the puzzle is programmed (the dependent variable), following the rationale that it is the programmer who decides to give the player a prize based on what the programmer thinks qualifies as the right answer.

The *conditions* (or treatments) that we intend to compare are reflected in the explanations that the subjects provide, being under the influence of three contextual metaphor primes and three types of cultural background.

The experimental conditions are controlled and kept constant to the extent that we recorded the time spent on the tasks and thus ensured that the tasks were completed within a reasonable time-frame. Thus, we excluded the effect of any seriously potentially confounding variables, such as diffusion of experimental manipulations (i.e., we reduced the possibility of participants sharing the contents of the tasks with other participants). Participants completed the task individually and received identical instructions, and the hypotheses were not revealed to the participants. Such non-disclosure of hypotheses is the most robust experimental procedure, and it is employed in around 87% of all experimental-psychology research (Hertwig and Ortmann 2008) because it allows for the elicitation of valid measures of behavior instead of relying on less valid measures by means of other methods, s.a. self-reports (Bröder 1998; Christensen 1988; Kimmel 1998; Trice 1986; Weiss 2001).

For analyzing the second main hypothesis that we proposed in the Introduction, pertaining to the potential influence of the context, the *research hypothesis* is that the manipulation ("prime") will increase the number of the corresponding explanations the participants give. The participants' explanations for their respective choices were qualitatively coded according to the three predefined categories. Explanations conforming to one of the three predefined categories were categorized both according to their discrete category (i.e., 'balance', 'shapes' or 'algorithm') as well as whether they were 'sensical' (i.e., eligible for inclusion in the predefined categories) or 'nonsensical'. Non-interpretable explanations were thus labeled 'nonsensical' and discarded (see Sect. 6.1 for a thorough analysis of this). If the rationale of prime manipulation in the respective condition is chosen significantly more than the other rationales, this would imply that the participants were influenced by external features that are not relevant to the programming task itself.

We implemented one additional variable to control for the bias, resulting from an observation in our practical use of the cognitive task from Sect. 2, that the choice of placing the letter H is also an indication of the rationale. Particularly, participants choosing *Left* would be those using the rationales I and II from Sect. 2, whereas participants choosing *Right* would be those using the rationale III for 'algorithm'. This is analyzed in Subsection 6.2.1.

Even though we chose the subjects based on their educational and professional background, we also asked them to provide information about their educational background themselves, as well as information about their preferred free-time activities. This was done to disclose a possible relation between this particular aspect of the background of the participants and their choices in the programming task. Moreover, this information from free-text questions can also help detect respondents that did not relate seriously to the task, as well as to control our qualitative coding of their explanations and background.

Alternative explanatory variables

The age and gender of the participants are analyzed as *alternative explanatory variables.* Other alternative explanatory variables that might occur could result from the subjects not understanding the task well, the task being too difficult, or the prime not being strong enough as a result of superficial reading. However, these factors were something that we detected and removed through our *pilot tests*.

By implementing the questionnaire questions related to individual preferences and extracurricular activities ('Ranking life-aspects' and 'Words suggestions' pages, and the question about hobbies on the 'Demographics' page), we expect to be able to clearly identify if the choice was dictated by the bias. Moreover, the programming task is mean to be very simple, thus requiring very little cognitive effort. For such cases, it is empirically proven that the individual differences have a small impact (Lazar et al. 2017).

5.3 Learning from the first full-scale study

The first full-scale study, also referred to as the 'Social sciences' cohort, consists of undergraduate students enrolled in a psychology study program. The link to the survey on SurveyMonkey was sent through email by the study program administrators and resulted in 77 responses. Observations

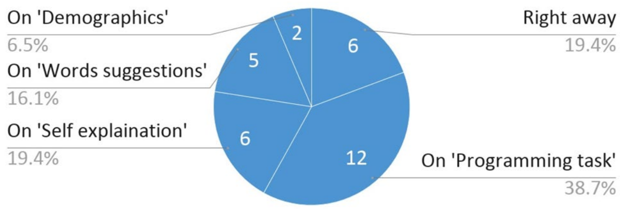


Fig. 4 Overview of the dropout number of respondents from the ‘Social sciences’ cohort; including the names of the pages where the dropout happened

made after the first study helped with improving the following studies. An analysis of the incomplete responses (31 out of 77) from the first study is shown in Fig. 4.

The high number of dropouts on the ‘Programming task’ page could be explained by the fact that these psychology students may have deemed this task as not relevant, not interesting, or maybe too difficult. Based on this reasoning, in the subsequent studies we introduced on the first page the mention “It is not required to have any prior knowledge of computer programming.”, and on the ‘Programming task’ page we wrote that the puzzle is simple.

Another solution for further motivating the respondents to finish the survey was to add a progress bar indicating how much of the survey was left until completion. For the last three pages we also added a page-footer informing, consecutively, that ‘there are three, two, and one pages left’, i.e., aiming to reduce the two latter types of dropouts.

Recall that the first study was conducted with volunteer *social science* students that were neither paid nor participating during their normal class hours. In contrast, the second study was run in a lecture hall, before the break, as part of a first year *computer science* course. In the case of mTurk and SurveyMonkey respondents from the third study, we consider the payment as an important motivating factor (see Sect. 5.3).

In order to reduce the cognitive effort required (and hopefully the dropout rate) on the ‘Self explanation of choice’ page we added, immediately after this page, the page called ‘Alternative explanations’, containing a list of predefined explanations to choose from. This was meant to reduce the high dropout rate that we saw on the ‘Self explanation of choice’ page. Moreover, adding these alternatives in the second study reduced the number of uninterpretable answers significantly (see Sect. 6).

For the second study, the total number of responses received was 53. Of this total number, one respondent dropped out on the ‘Programming task’, one on the ‘Self explanation of choice’, one on the ‘Ranking lifeaspects’, and two on the ‘Words suggestions’ page. The small number of dropouts in this second study indicates that the adjustments made after the first study were successful.

5.4 Transitioning from volunteering students to professional respondents

For the third cohort, we recruited people with a background in arts and culture in general. We started the third study with two groups of students, studying music and theater. Though we had no dropouts from these groups, the numbers of students in the classes were small (which is specific to these kinds of studies), i.e., 10 respondents from music and 10 from theater. To increase the number of responses we also recruited respondents through the specialized platforms Amazon Mechanical Turk and SurveyMonkey. These would no longer be volunteers but professional respondents who are paid for their participation and do such tasks often.

From the total of 128 responses we removed 13 respondents that spent less than four minutes on completing the survey (average response time from the previous studies was eight minutes). Additionally, seven more respondents were rejected as we deemed them unserious (e.g., computer generated answers). Out of the remaining 108 responses, one participant dropped out on the ‘Self explanation of choice’ page. Moreover, six respondents that spent more than four minutes were still deemed unreliable and thus removed from the analysis. This was decided based on the quality of the responses given in the open-ended questions. (See more details in Johansen et al. 2020, Sec. 6.4.)

6 Analyzing the data

All the examples of answers from the participants are presented here in English, but many of them are translations from Norwegian (including grammar corrections; though not for the English ones, which are kept verbatim, including their grammatical errors). Moreover, all examples are marked with the information (ID and cohort) useful to track the respective response within our dataset, which can be made available upon request (or by following information that will appear in the updated long version associated to this paper (Johansen et al. 2020)).

6.1 Sensical vs. nonsensical answers

The participants’ explanations were analyzed qualitatively and coded into one of the three rationales from Sect. 2. During the first full-scale study we found one answer (pID 38, first cohort) which triggered us to introduce another category or rationales, called ‘sounds’; the answer explained the choice of letter placement as “If you sing the alphabet in Norwegian then the best fit with the rhythm is to place ‘H’ to the left, because you have a small pause before singing ‘H’ after ‘G’.”.

There were still many answers that could not be categorized, either because they did not make much sense, or the reason given was no reason at all. However, many of these answers were recurrent, transcending even the language differences, and this allowed us to group them in categories. (See Johansen et al. 2020, Sec. 7.1 for details.) Some of the more generic answers were so similar between English and Norwegian that we could regard them as 'universal'.

- 'Logical': "I think it would be logical put the H in the right position" (pID M:11272137574, third cohort); or "Left seems right because it seems logical" (pID 53, first cohort); or "because it seemed most logical" (pID 12, first cohort).
- 'Pattern': "My choice was made by what I thought was a pattern" (pID M:11282013578, third cohort); or "because of the order of the previous ones." (pID 47, first cohort); or "Due to previous placements above." (pID 50, first cohort); or "The left seems to follow the pattern" (pID M:11270235127, third cohort); or "Because I think the pattern follows that path." (pID 4, first cohort).
- 'Random': "Just chose something" (pID 33, first cohort); or "It seemed like the pattern of the letters would place the H on the right, but there isn't enough information for me to decide, so it is kind of a guess." (pID M:11270119183, third cohort).
- 'Alphabet": "…going in reverse alphabetical order." (pID M:11272389655, third cohort); or "The letters are to be placed based on the alphabet song." (pID M:11271323609, third cohort).
- 'Handed': "I'm right handed so I favor my right side and it just seemed like the 'correct' answer to me." (pID M:11271930008, third cohort); or "Most people are right-handed, so dragging the letter to the right felt like an automatic default action. Dragging it to the left requires a more deliberate choice." (pID M:11270365264, third cohort); or "I chose the right because in every day society its pretty common for the right side of thing to be accepted as good, such as right handed people, the right hand of god, etc. etc. I also chose the right side because its 'right'." (pID M:11270469031, third cohort).
- 'No reason': "Because it looked most natural compared to what has already been done." (pID 36, first cohort); or "it looked natural" (pID 7, first cohort); or "It felt reight" (pID S:11178992036, third cohort); or "Seems better" (pID S:11174871629, third cohort).
- 'micro-balance': "H on the right side follows the pattern of the EF on the left side, which are a pair." (pID M:11272137574, third cohort); or "Because it makes sense to me that H and G are grouped together, since there is a grouping on the other side as well." (pID T:11058678726, third cohort); or "In my opinion it looks nicer to have 'H' after 'G'. It has a bit to do with how 'E' and 'F' are positioned." (pID T:11058678669, third cohort).

To reduce the number of uninterpretable answers, starting with the second full-scale survey, we introduced the alternative answers which were formulated based on the wordings that we encountered among the responses from the first study. Thus, the first study provided a type of 'saturation' of alternatives. As a result of coding and categorization we arrived at five alternative answers, as well as a sixth and seventh alternative: "I just chose something" and "I already gave an explanation".[10] We also used these to help us code the answers, i.e. when they did not give any explanation (it was not required) but instead chose from our list, we used that choice as the rationale. When they gave an explanation that did not make sense, but then also chose one of our example explanations, we again used the one that they chose, for our categorization. There was also the case when their explanation somewhat seemed to contradict the choice that they picked. In this case, we still used the choice for the categorization. The following are a few explanations that made no sense, but an alternative was chosen: "The left side seems like the logical, correct side when compared with the letters that came before it." (pID M:11270382691, third cohort) but then chose the alternative answer that sounded "Because of the appearance/form of the letters. On the left side they have straight lines, whereas on the right side are rounded."; or "I choose left because i think it can be very good with random letters in the left." (pID M:11270101280, third cohort) but then chose the alternative "The same number of letters on each side."; or "There seems to be a pattern. Placing the letter on the left makes the most sense to continue that pattern." (pID M:11271499180, third cohort) but then chose a pattern from the alternative that sounded "It creates a pattern of the type: 1–3, 2–4, 3–5, …or 1–3–2, 1–3–2, …or 1–3–2, 2–3–1, …".

We thus define as *Sensical* those answers that were interpretable and allowed for category inclusion in one of the three rationales from Sect. 2, and we define as *Nonsensical* the remainder of the answers.

In the following we make two observations about our sensical vs. nonsensical perspective on the responses.

6.1.1 Helping with the self-explanations

The first regards the level of help that the different cohorts received. We observe in Fig. 5 a substantial increase of answers that allowed for interpretation when the respondents were offered the alternative explanation choices. The

[10] Recall that this was a 'required' question/page (marked with*), as opposed to the previous 'Self explanation of choice' page, and thus a choice must be made on this page.

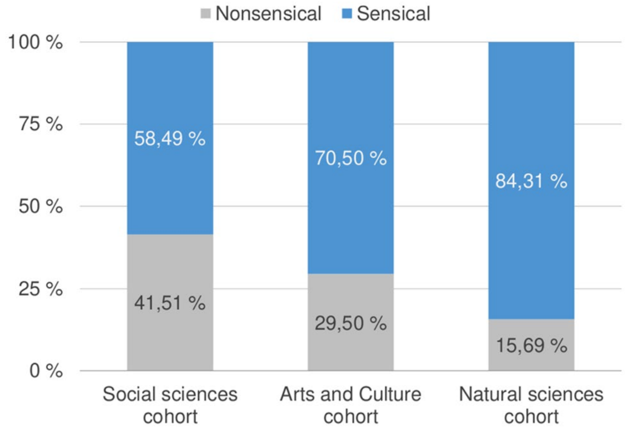


Fig. 5 The nonsensical answers are decreasing after improvements done to the survey

‘Social sciences’ cohort were not helped and the percentage of sensical responses is only 58.49%. To all other respondents we allowed them to skip the ‘Self explanation of choice’ question and required that they at least chose one of the alternative explanations. The sensical answers increased significantly to 70.50 and 84.31% for the ‘Arts & Culture’ and the ‘Natural sciences’ cohorts, respectively. It is particularly noteworthy the increased level of interpretability that this choice in the design of our studies brought. We have counted 22 answers given by the participants in the ‘Arts and Culture’ cohort that were not understandable only by themselves, but could nevertheless be coded because of their choice of alternative explanation. This would have otherwise tilted the percentage to only 54% sensical answers. We also had 10 that chose to skip the self-explanation and only select one of the alternatives.

6.1.2 Programming environment confinement

One can observe that as soon as the participants were confined their explanations became even more sensical. Here we look at the two cohorts to whom alternative explanations were offered, and we notice the increase from 70.50 to 84.31% in the case of the respondents from the ‘Natural sciences’ cohort who were confined to the classroom and course working-hours. This bears evidence that the transition from a nonprofessional towards a more professional programming environment would trigger the programmers to be more careful about their choices. This could also contribute to lowering the amount of bias. Indeed, we have observed that several participants tried to think in terms of games, since the task consisted of programming a game. Examples of such explanations are: “I choose left because it’s a game and i think according to the pattern gamer will choose right side psychologically. Thus he/she will lose.” (pID M:11274822275, third cohort); or “I feel the right side would be the most common choice so if the player was thinking creatively they would choose the left side to place the h” (pID M:11274883590, third cohort).

Another aspect of the confinement is that it triggers the System 2 thinking, which is known to result in a reduction of human biases. We have also observed instances of System 1 vs. System 2 thinking, i.e., ‘starting’ as a System 1 response, but then ‘self-apprehended’ and activated a System 2 response; e.g.: a participant wrote in the ‘Self explanation of choice’ page “I choose right previously but actually left makes more sense. Balancing the sides; 4 letters on the left, 4 letters on the right.” (pID M:11272410463, third cohort).

Such observations should be further investigated using more controlled experiments. In any case, one piece of conclusive advice that we can offer is that it is useful for the outcome of the experiment if the respondents are given (i.e., as help) alternative choices of answers/explanations (or rationales in our case). These choices should be carefully made, preferably using answers that are observable in the target population (as we did ourselves, extracting answers from the first survey). A more controlled experiment should yield more sensical answers, e.g., by carrying out the experiment in a more strict ‘laboratory’ setting. It seems that only paying the participants, as we did through the two platforms SurveyMonkey and Amazon’s mTurk, does not increase the quality of the answers.

6.2 Control questions

In the study we included several additional questions with the purpose to control for various aspects. As one can recall from Sect. 5.3, we have used the open-ended questions to identify robot/automated answers. Three questions were of particular importance, as they were meant to control, or to reinforce, three important assumptions that we have. Essentially, Sect. 6.2.1 reinforces our bias revealing test from Sect. 2 as a good instrument; Sect. 6.2.2 tests how well our priming metaphors from Sect. 3 worked, since such story-based metaphors may be revealed within listings of words/synonyms; whereas Sect. 6.2.3 reinforces our beliefs and categorization of the backgrounds of the three cohorts that we study, thus confirming that the categories/labels we provided in Sect. 5.1 are appropriate, and the bias transfer results that we report in Sect. 7.1 are well informed.

6.2.1 Left/right placement

On the *‘Programming task’* page of the survey, the respondents are asked to decide whether to reward the player for the

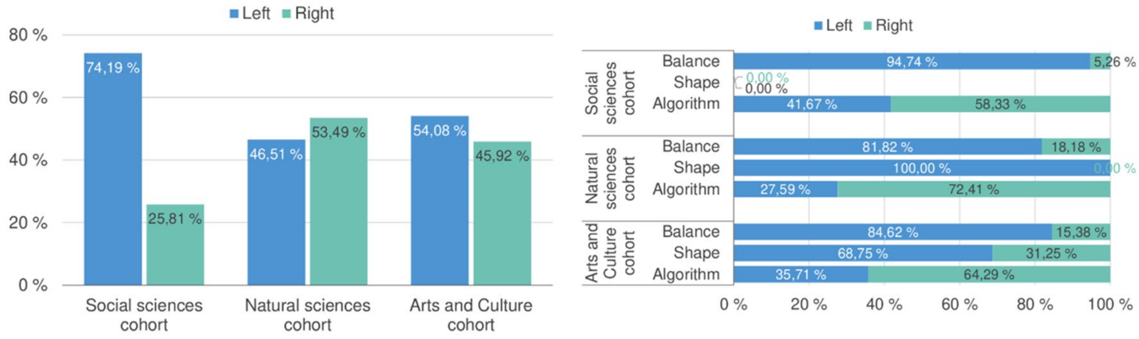


Fig. 6 Overview of the choice of placement of the letter 'H' to the Left or Light. **a** Inside each cohort. **b** Distributed by the three rationales (into which the respondents' answers to the 'Self explanation of choice' question were categorized)

placement of the letter 'H' on the left or right side of the vertical line on the game board.

This is one of the three questions on this page, intended as a control question for the hypothesis that we made in Sect. 2, i.e., that choosing to place the letter to the 'right' should indicate a preference for the 'algorithm' rationale, while choosing 'left' a preference for the 'balance' or 'shapes' rationales.

An analysis of the 'left/right' placement with respect to each of the three rationales confirms this initial assumption, see Fig. 6b for numbers. In particular, observe that in the case of the 'algorithm' rationale the choice of placement to the 'right' is overwhelming for each cohort; and similarly, 'left' is the preferred choice when answering with the 'balance' or 'shapes' rationales in all cohorts.

Moreover, the analysis of the 'left/right' placement overall inside each cohort, which we summarize in Fig. 6a, confirms our earlier observation that the background of the participants is reflected in their preference for one choice of placement (and thus for one type of rationale).

6.2.2 Words suggestions vs. priming metaphor

The participants were given three pairs of words to suggest synonyms for, each containing two of the four priming words used in the 'Philosopher story'. In analyzing the responses for the *'Words suggestions'* question, we looked for the occurrence of the other two words that were used in the 'Philosopher story' as primes (cf. Sect. 3 also). In Johansen et al. (2020, Sec. 8.2) one can find a thorough analysis of the words given by the respondents to the 'Words suggestions' question of the survey, which was meant to reveal which of the primes worked and how well.

The numbers from our analysis show that the priming metaphors for 'balance' and for 'shapes' were chosen well, whereas the words for the 'algorithm' metaphor were too difficult, which diminishes the strength of the priming.

This analysis also indicates two other factors that might have had influence on the priming effect. One is how familiar the respondents are with the priming words. If the words are very little known or not understood, people will not be primed by them, as it is the case of the 'algorithm' words. For the second factor, if the respondents have a large vocabulary at their disposal, the System 1 will be less inclined to use the priming words in this synonyms question. Such observations can be made in the case of the 'Social sciences' students in comparison with the 'Natural sciences' students: 147 unique words compared to 95. We see how the priming was stronger in the latter cohort compared to the former (they strive to find similar words, and the availability heuristic retrieves the primes from the short term memory).

The words the participants choose the most can also be affected by other immediate contextual elements. In the case of the 'Natural sciences' cohort the survey was taken by the students as part of a course on logic. This made the word 'logic' occurs the most for 'order and continuity'.

6.2.3 Life-aspects ranking

This was meant as a control question for the way we identify the background in our cohorts. That is to say, we want to check whether there is a correlation between the self-ranking of the 'life-aspects' and what we have considered as the background of the respondents. Moreover, we want to also look at the coded answers from the 'Self explanation of choice' compared to the 'Ranking life-aspects' because if the correlation is similar to the one we have observed previously from the background, then this would reinforce our perception of background.

For creating the three types of cohorts we have considered the educational and professional backgrounds. However, these are only one part of a person's background, arguably a large part, but yet a larger part is made of the society and culture that the respondents belong to. This is especially so for younger people, such as students.

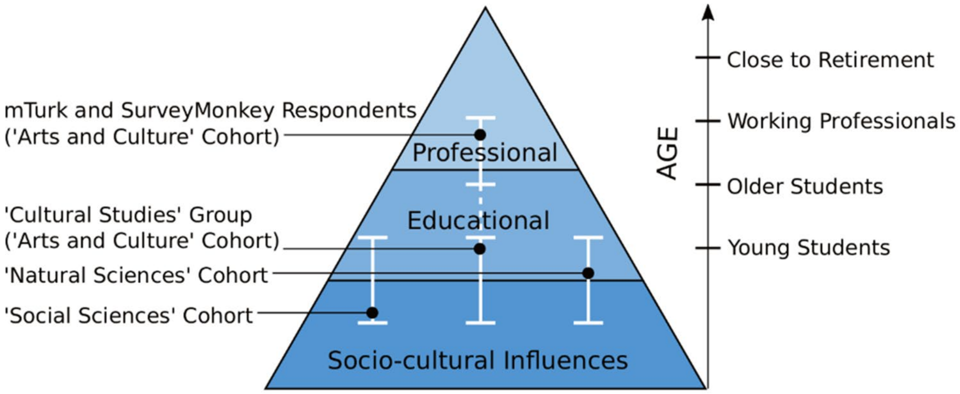


Fig. 7 Three sources of influence, related to the age when they are most strong, for the backgrounds observed for our cohorts, also indicating the age groups observed from demographics data

For the 'Ranking life-aspects' we observe influences that come from the socio-cultural as well as educational and professional backgrounds. How strong these are, and how much they relate to the bias transfer that we have observed before, is what we investigate in this section. Recall that the names that we gave to the 'life-aspects' to be ranked by the participants were each using two of the four words used in Sect. 3 as priming metaphors, i.e.: 'harmony and equality', 'aesthetics and arts', and 'order and continuity'. One word from the start of the story and another from the end of the story.

One's view on life is, among others, highly influenced by society and culture (Cialdini and Goldstein 2004; Schultz et al. 2007; Cialdini 2009). For children this may be the main influence (e.g., through their parents), whereas for young adults (like many of our respondents who are young students) other factors of their own life-experience start to influence their views, including their education when they are studying and their professional environment when they start working. We summarize the three types of influences in Fig. 7, organized as a pyramid to suggest the strength and time of the influence.

From the data analyzed in detail in Johansen et al. (2020, Sec. 8.3) we can conclude that the control question about 'Ranking life-aspects' confirms our assumptions about the backgrounds for our three cohorts and the fact that we have associated each of these cohorts with the life-aspect that is most predominant for those respondents. Therefore, we consider adequate the claims that we make throughout the paper where we correlate the background of a cohort with one specific life-aspect, and thus with one specific corresponding bias/rationale.

7 Results

The data are analyzed both quantitatively and qualitatively. The qualitative analysis is done to detail the quantitative data, by analyzing the participants' responses to the open-ended questions. Since our study is exploratory, we employ a combination of statistical and descriptive analysis. Statistical analyses were not possible in all situations because of the small number of respondents in those categories.

7.1 Influences from the cultural background

Students with a cultural background from social sciences differed significantly from students with a cultural background from computer science. Social sciences students were significantly more prone than computer science students to describe their choices matching the rationale 'balance', whereas computer science students were significantly more prone to describe their choices matching the rationale 'algorithm': $X^2(1,N=71)=8.1686$, $p<0.05$ (with calculated p value of 0.004262). These results support the hypothesis that the cultural background influences people when they carry out programming tasks under conditions of uncertainty.

The statistical significance test, as well as the graph in Fig. 8a consider the total number of responses, from all four treatments. The same observations about the cultural background influence are confirmed also when looking only at the control group (see the graph in Fig. 8b), though a statistical test is not relevant in this case, given the small number of responses. For both graphs the percentages are calculated from the 'sensical' answers only.

When analyzing the results from the 'Arts and Culture' cohort in comparison with the other two cohorts

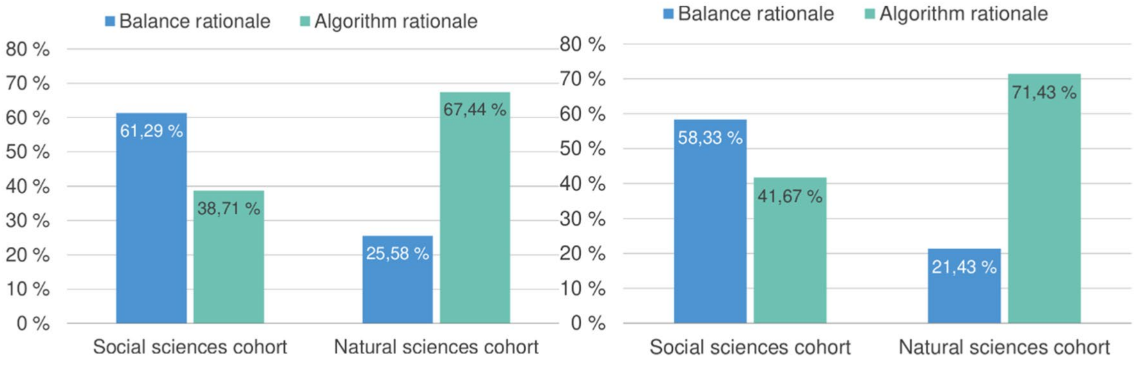


Fig. 8 Comparison of background influences (cultural biases transferred). **a** All treatments included. **b** Control group only

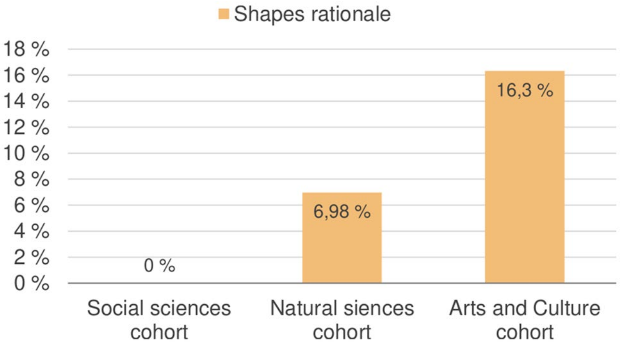


Fig. 9 Comparison of number of answers categorized in the 'shapes' rationale

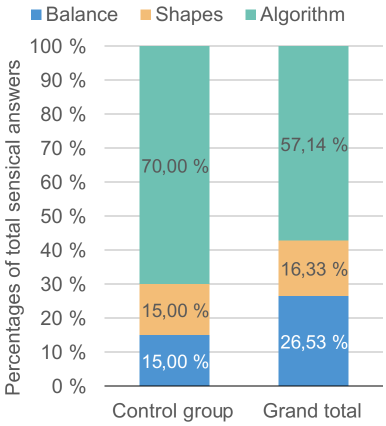


Fig. 10 Arts and culture cohort—percentage of answers distributed by rationale

(see Fig. 9), we see that the influence of their artistic background makes them choose much more the 'shapes' rationale.

Analyzing further this cohort by itself, independently of the results obtained for the other cohorts, we see in Fig. 10 that the answers conforming to the 'algorithm' rationale are dominant; both when looking at all responses as well as only at the control group. This dominance could be explained by the fact that the respondents tried to comply with the nature and requirements of the exercise, i.e., a programming task where they were asked to assume the role of a programmer. One example of an answer from this cohort confirms this affirmation: "It was always drilled into my head in school, that when it came to math (which I assume is what most programming deals with) that the right side is always the right way… 'right side right way' that's my reasoning here." The respondent tries in this case to bring in to his/her help the math knowledge s/he has from the school, as s/he assumes that informatics "deals with" mathematics. Another example is "I can't think of a better explanation but to involve mathematics in this game…".

Moreover, when analyzing qualitatively the answers to the 'Self explanation of choice' question we found a considerable number of respondents that brought the game aspect of the task into their reasoning (more than 30 out of 110 explanations of the 'Arts and Culture' cohort), i.e., they think in terms of programming a game. This is also an indication that these respondents focused on the task at hand, seeing the puzzle as part of this game programming task—as they have been asked to—and did not try to solve the puzzle per se. This increases our confidence in the fact that there was no debiasing happening, and that the respondents did not recognize that the task was in fact meant to reveal a background bias, let alone one of our three rationales or cohort backgrounds that we have assumed. Another aspect that could trigger debiasing is the fact that our puzzle does not have a 'correct' answer wrt. the letter placement. However, we have found only two responses that have identified this fact ("[…] because of both dotted boxes are the correct answer. However, I feel[…]" from pID M:11270469031, third cohort, and "[…]there isn't enough information for me to decide, so it is kind of a guess." from pID M:11270119183, third cohort); therefore, we rule out this debiasing possibility as well.

7.2 Influences from the priming metaphors

People who are influenced through priming generally do *not* realize it, and thus one does not normally see the priming expressed per se in the respondents arguments. Instead, the respondents being primed would make use of one or more of the heuristics that we mentioned in Sects. 1 and 3, e.g., the availability heuristic uses information from the immediate environment, which in our case is the *'Philosopher story'* in the programming task specification. An example of the unconscious manifestation of this heuristic is pID M:11272410463, third cohort: "I choosed right previously but actually left makes more sense. Balancing the sides; 4 letters on the left, 4 letters on the right.". The qualitative analysis of the respondents' answers to the 'Self explanation of choice' reveals many such *unconscious* uses of the priming words; see details in (Johansen et al. 2020, Sec. 8.2). However, we also found three instances where participants from the 'Arts and Culture' cohort quote directly the primes from the 'Philosopher story' to help in arguing their reasoning behind the choice of letter placement. These are examples of *conscious* use of the helping material from the context of the task. Interestingly, all invoke only some of the four words that we used for priming, and these words are taken both from the start and end of the story, which confirms our decision of using several words placed at different points inside the 'Philosopher story'.

- pID M:11271203853, third cohort: "If this game is based on the philosopher's tenet of *balance and harmony*, then[…]"[11]
- pID M:11270378880, third cohort: "The player should be rewarded when he/she places the letter on the right side because that is in keeping with the *continuity and linear structure* of the game."[12]
- pID 103, second cohort: "The philosopher thought *balance and equality* were important, and the player should therefore be rewarded for restoring the balance between the number of letters on the right and left sides."[13]

Heuristics are used substantially in situations of uncertainty, which is the case for our puzzle since we ask participants to find one 'solution' to this new puzzle, which at the same time does not have one single correct answer, as any argument would be acceptable. In cases of uncertainty two additional heuristics are usually employed, namely the representativeness heuristic and the anchoring heuristic. If the problem at hand is new, then the mind tries to find another previously encountered problem that, to some extent, has some similarities. This is the case with the puzzle that we devised, aiming to trigger associations with aspects from the cultural/educational background of the person, e.g., 'Natural sciences' respondents were expected to cling on to algorithms and the alphabet as an ordered source of indexing in mathematics, thus continuing along the line in our puzzle. The anchoring heuristic is even more important for priming since it is often employed when no useful information is readily available for the problem at hand, so the mind looks into the immediate context (e.g., physical, s.a., surroundings, or temporal, s.a., information received in the recent past, from the short-term memory) for clues. In our case the mind would anchor into the 'Philosopher story' metaphor, and maybe draw on the meaning of one of the four priming words.

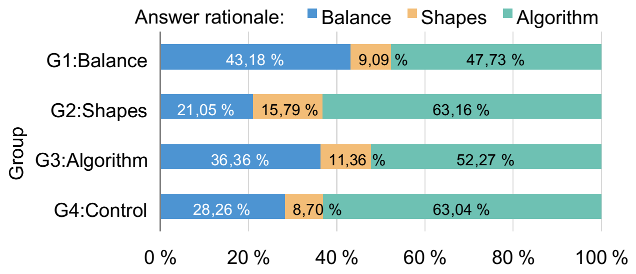


Fig. 11 Effect of priming, irrespective of background—respondents from all cohorts put together

We observed influences of our experimental manipulations, albeit not reaching statistical significance. Thus, since we can neither rule out a Type I error nor a Type II error, in the rest of this section we present results from quantitative analyses of the priming effect and whether or not this transferred to the programs.

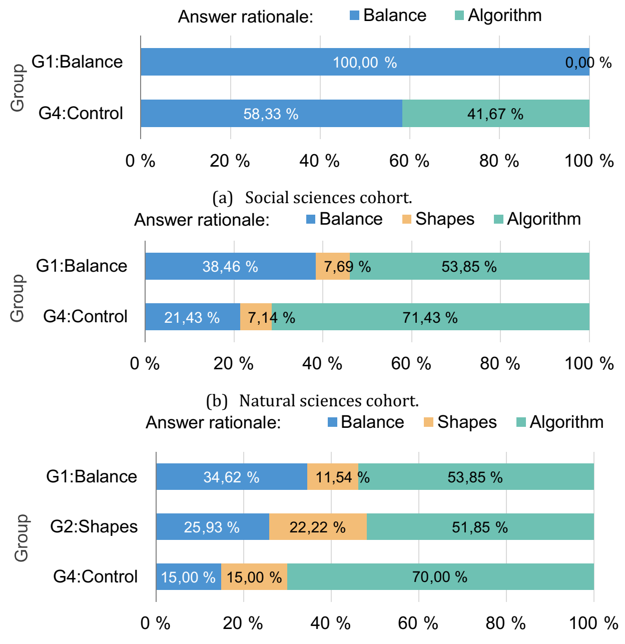
The graph in Fig. 11 shows the influence of the three groups of priming metaphors when the responses from all the cohorts are put together. This shows the priming effect irrespective of the participants' background. We compare each group with the control group.

First, we observe that the 'algorithm' group gives answers that cannot be readily seen as being influenced by priming. The same inconclusive observation is found also when looking inside each cohort, comparing the 'algorithm' group there with the respective control group. This conforms with the observations made in Sect. 6.2.2, where the words used for this priming are little known or maybe not understood by the participants, and thus cannot have an impact on their

[11] The words are found at the start of the story.

[12] There are actually three of the priming words mentioned here: "linearity and continuity" (from the end of the story) and "structure" from the start of the story. However, the respondent puts together "linear structure".

[13] One word from the start of the story and one from the end.



(a) Social sciences cohort.

(b) Natural sciences cohort.

(c) Arts and Culture cohort.

Fig. 12 Effect of priming, inside each cohort

choice.[14] Therefore, we focus in the rest of the section on the other two groups of priming.

Secondly, when we analyze the other two groups we clearly observe priming influences, albeit of different kinds as explained further. For the 'balance' group we see that the 'balance' rationale increases from 28.26% (for the control group) to 43.18% (for the 'balance' group), whereas in the case of the 'shapes' group the 'shapes' rationale increase from 8.70 to 15.70%; irrespective of the background.

Besides these observations about the general strength of the priming metaphors, we are to a greater extent interested in their interactions with the educational/professional background of the participants, as discussed in the previous subsection. In the chart from Fig. 12a, related to the 'Social sciences' cohort, we observe that the background of the respondents is further strengthened by the priming metaphors, with an increase from 58.33 to 100%. Quite the opposite, in the case of the 'Natural sciences' cohort in Fig. 12b, we see a weakening effect of their background since the 'algorithms' rationale decreases from 71.43 to 58.85% in the group that was influenced with the 'balance' metaphor, in favor of the 'balance' rationale. For the 'Arts and Culture' cohort we again see in Fig. 12c that the 'shapes' metaphor strengthens their background since the 'shapes' rationale increases from 15 to 22.22% inside the group that was primed with the 'shapes' metaphor. For this cohort also the 'balance' metaphor has an influence (from 15 to 34.62%), due to the fact that this metaphor's words were well chosen, as we have observed in Sect. 6.2.2.

We can conclude that the contextual metaphors that have been deemed as strong enough in the control question 'Words suggestions' are also found to have an effect in strengthening or weakening the influence from metaphors in the cultural background of the respondents. Contextual metaphors have a strengthening effect when the words are representative of the respective cultural background. At the same time, well-chosen contextual metaphors can weaken the effect from the cultural background when they go against it, e.g., 'balance' metaphor applied in the 'Natural sciences' cohort.

8 Conclusions and discussions

The aim of this study, as well as its implications, are manifold. The study can be categorized as both (i) a comparative/experimental study of how biases from cultural and contextual metaphors can be transferred from programmers to programs, and (ii) an exploratory study on how to develop ergonomically valid and reliable instruments, procedures and testing conditions to empirically study such biases transfer.

As such, this paper is a foundation for future research endeavors to *improve* and *diversify* these instruments, procedures and testing conditions.

The strengths of this work reside in its exploratory nature in studying a hitherto not researched phenomenon, namely the transfer of human biases from the (not necessarily expert) programmer to the artifact that is developed (or configured). Concretely, we have exposed (in Sect. 7) interesting aspects of our main hypothesis, namely that machine bias may originate also from the programmer's biases in terms of influences from the cultural background as well as contextual influences from the programming environment.

In particular, under conditions of uncertainty (e.g., in the absence of instructions or specifications, something which is often the case for ubiquitous systems programming carried

[14] However, one needs to take this conclusion with a grain of salt because the priming metaphor, depending on the anchoring heuristic, has a temporal flavour as it is stronger closer to the time of the priming; i.e., in our case the *'Self explanation of choice'* question is very close to the priming metaphor, whereas the 'Words suggestions' question is farther away, maybe with a delay of a few minutes. This can mean that even if we do not see an effect of the metaphor in the 'Words suggestions' question one can still have some effect in the 'Self explanation of choice' question. Moreover, this can be compounded by other factors as well, such as for the 'Words suggestions' question we are looking only for two of the words whereas in the 'Self explanation of choice' question all our four priming words are in effect; or by the semantics of the words, which can have different meanings in different context, thus possibly causing one influence on the programming task and another influence on the synonyms generation task.

out increasingly by non-experts), we observe that the programmers' cultural background influences the choices they make and are subsequently transferred from the programmer to the program artifact. Thus, cultural metaphors in terms of irrelevant and inappropriate influences on the programming task at hand, represent instances of biases that are being transferred from humans to machines. This implies that human culture 'transfers' to machines through the humans that program these, thus representing a strong source of bias.

Interestingly, attempts to moderate the strong influence from the cultural metaphors by means of experimentally introducing 'hidden' (i.e., not consciously detected) contextual metaphors, were only successful to a certain extent. When the priming metaphor was chosen well (as in the case of 'philosopher story' related to the 'balance' rationale; with words that were easy to understand and rather common in a standard vocabulary) we observed influences in both directions of strengthening the cultural background as well as moderating it, each time tipping the balance of answers in the direction of the metaphor. These findings are orthogonal (i.e., do not contradict, but supplement) to what traditional and current machine bias research suggests, i.e., that machine bias originates from data, and thus our findings provide new insights into the origins of bias in the wide spreading AI and decision-support systems.

We believe that the present study shows how various aspects regarding design, instruments, and procedures can be successfully explored and controlled, and consequently incorporated in future studies that could (i) extend the present study by exploring related causes and mechanisms that lead to the transfer of bias from programmers to programs, as well as (ii) improve the designs, instruments and procedures in order to undertake this expanded endeavour.

One interesting speculative observation that we would like to make out of our results regards a potential effect resulting from the difference between (i) interpreting information based on its *structure* and thus as something systemic that is 'detached' from having individual characteristics, versus (ii) interpreting information based on its *content* and thus as having individual characteristics. For example, subject programmers that chose the rationales of 'balance' or 'algorithm' may view information (as the one coming from the 'game board' puzzle picture that we showed them) merely as representing structure and may thus have disregarded the notion that data could also have individual characteristics in addition to being part of an overall structure. Contrary to this, respondents that chose 'shapes' may in fact have acknowledged the notion that data do have individual characteristics and are thus not 'only' part of an overall structure 'outside' the data's individual characteristics. Interestingly, subjects in the arts & culture cohort provided explanations in terms of 'shapes' substantially more often than subjects in the 'Social sciences' cohort and the 'Computer science' cohort. This could indicate that people with a cultural background (judging from their education and/or profession) from arts and culture are more prone than others to view data as representation of individuals that have unique characteristics, rather than viewing data only as being part of an overall structure. In other words, people with a background in arts and culture may possibly exhibit a more 'human' interpretation of data, or at least they may be more prone than people from other cultural backgrounds to acknowledge data as 'individual' rather than 'systemic'.

8.1 Implications

One can see several immediate benefits of the present study alone. For example, in education one could measure how well programming courses train the students, by measuring the bias transfer-rate at the start and end of the courses. Another example is to measure how effective some technology quality assurance method is at removing or identifying programmer's biases, such as testing frameworks, peer programming, abstract/detailed specifications, code generators, etc. Moreover, regarding the growing population of 'lay' programmers in the smart-living and IoT-ubiquitous programming environments of today (i.e., almost everyone in technologically 'modern' societies) both business companies and consumers would benefit from more insight into the non-conscious influence of culture and context on the programming choices that are made by the 'novice' programmer that has no formal training. In terms of education and learning, we argue that this insight could be used to help consumers become more aware of the cultural and contextual influences that shape their cognitive tendencies when they are programming. The work reported in this paper is relevant for researchers from several fields. First of all, people working in AI and machine learning can be interested in our proposal that biases in machine learning can come not only from the data but also from the people programming the algorithms. We study this to some considerable detail, as we explain in the rest of this introduction. Second, people working in psychology and cognitive sciences can be interested in this new application that we propose, where they can apply their skills and methods to study this new form of human bias and its transfer to machines. Third, practitioners working with software engineering or managing software development teams can be interested in studying more various programming environments and tools to see how much human bias is transferred to the programs in each situation. Finally, at a macro level, both governments, private business enterprises, and NGOs would become aware of machine bias originating from human programmers who unknowingly transfer the influences from their own cultural backgrounds to the machine programs. Thus, the target audiences are diverse and would benefit both on a micro level, e.g., in research and development, and in (computer science) education, as well as on a macro level, e.g., in issuing improved knowledge-informed national regulations on the domains where automated decision-support systems operate.

8.2 Possible future research directions

One venue for future research would be to refine our study design's ability to elicit cultural or contextual influences in an even more fine-grained manner, specifically by improving our instruments and procedures. One possibility is to perform similar studies focused on specific categories of subjects that can be seen as programmers, e.g., one playful possibility could be to study children as programmers—programming languages/environments specific for children abound, s.a. Google's Blockly (Trower and Gray 2015; Weintrop and Wilensky 2017) or MIT's Scratch (Resnick et al. 2009; Maloney et al. 2010; Armoni et al. 2015). Studies on biases in adults are more available (Klaczynski et al. 1997; Klaczynski and Robinson 2000; Bruine de Bruin et al. 2007) whereas studies on biases in children are less (Baron et al. 1993; Klaczynski 1997). One could argue that this is because children are not biased; others could claim that ethical considerations make such studies of children too difficult to carry out; yet others could argue that biases in children are distinctly different from biases in adults, given the differences in mental representations from children and adults. However, we think that it is important to test the age aspect in biases transferred to programs, given the ubiquity and pervasiveness of IoT-programming in everyday life for all age groups.

One useful refinement of our work could be to study professional programmers in a professional environment (e.g., as done in Cowgill et al. 2020), both (i) classical programming environments guided by software development life cycle methods and tools, maybe focusing on current emerging programming cultures such as Scrum or DevOps; and (ii) non-expert programming environments, s.a. complex configurations, DSLs, graphical programming, or curating of big data. One question can be: *What are the avenues that bring biases into the programming environment?* We have assumed that biases are a result of underspecified requirements. This is a common form of uncertainty in programming; but there are others as well. It is thus important to know which of these give way to biases, so that one can build debiasing techniques (Jolls and Sunstein 2006; Blackwell et al. 2009; Cheng and Wu 2010), maybe even incorporated in the tools of the programmers, like in IDEs [similarly to how others have developed culturally adaptive user interfaces (Reinecke and Bernstein 2011)].

One good source of alternative investigations can be the study of specific biases in specific situations or social activities where software is paramount. One example can be biases related to privacy in the big data economy (sometimes called the 'surveillance capitalism' (Zuboff 2019)), e.g.: are privacy related concepts or views from the cultural background—which is specific to the programmer—transferred to the software—which is used on an international scale? One can imagine a programmer coming from a cultural background that always promotes the slogan "You have zero privacy; get over it!", or another programmer from a background that "is entrenched by rules and regulations about who/how any form of private electronic data can be used". Are such different cultural views transferred to the software built by these two different programmers? What is the global influence of such bias transfers? In this setting, one could alternatively study biases coming from the user of the software (not the programmer) to see whether the user biases (call them 'wishes' or 'needs') are transferred to the software through specifications elicitation, user stories, and other interaction design methods (Rogers et al. 2011; Lazar et al. 2017) that are now a popular way of developing software systems.

We have studied two sources of biases, namely cultural metaphors and priming, that we consider situated at the two extremes on the vertical axis from Fig. 1, which indicates the strength of the bias, and also a temporal aspect regarding the persistence of these biases (e.g., priming may not be as strong as the culture, and acts on a short time scale, usually minutes after the priming is applied). One could study other sources of influence that would lie in between on our vertical axis, e.g.: propaganda [i.e., misinformation (Mintz et al. 2012; Kumar and Geethakumari 2014) and disinformation (Graham and Metaxas 2003)], which may be done on limited but considerable stretches of time; or working cultures which can influence a programmer in different ways when changing jobs.

Acknowledgements The authors would like to thank the student volunteers, their tutors and student administrative personnel from Oslo New University College (formerly Bjørknes University College), from Institute for Informatics at the University of Oslo, and from Oslo Art School (http://www.oslokulturskole.no) and Norwegian Academy of Music (https://nmh.no/) for their help with our studies. We would also like to thank John Sören Pettersson for allowing us to utilize their Usability laboratory at Karlstad University and the eye tracking equipment for our pilot testing.

Author contributions All authors have contributed on all aspects of the paper, though in different degrees as reflected by the order of the authors.

Funding Open access funding provided by University of Oslo (incl Oslo University Hospital). The first author was partially supported by the project IoTSec—Security in IoT for Smart Grids, with number 248113/O70, funded by the Norwegian Research Council. The third author was partially supported by the project SCOTT—Secure COnnected Trustable Things, with EU-Commission Contract number 737422, financed in part by JU ECSEL and the Norwegian Research Council.

Availability of data and materials The dataset used for this research can be made available upon request directly from the authors only, and not through the journal's systems.

Code availability Not applicable.

Declarations

Conflict of interest Not applicable.

References

ACM Policy Council (2017) Statement on algorithmic transparency and accountability

Akiki PA, Bandara AK, Yu Y (2017) Visual simple transformations: empowering end-users to wire internet of things objects. ACM Trans Comput Hum Interact (TOCHI). https://doi.org/10.1145/3057857

Armoni M, Meerbaum-Salant O, Ben-Ari M (2015) From scratch to 'real' programming. ACM Trans Comput Educ (TOCE) 14(25):1–15. https://doi.org/10.1145/2677087

Baeza-Yates R (2016) Data and algorithmic bias in the web. In: 8th ACM Conference on Web Science, p 1. doi:https://doi.org/10.1145/2908131.2908135

Baeza-Yates R (2018) Bias on the web. Commun ACM 61:54–61. https://doi.org/10.1145/3209581

Baron J, Granato L, Spranca M, Teubal E (1993) Decision-making biases in children and early adolescents: exploratory studies. Merrill-Palmer Q 39:22–46

Blackwell AF, Rode JA, Toye EF (2009) How do we program the home? Gender, attention investment, and the psychology of programming at home. Int J Hum Comput Stud (IJHCS) 67:324–341. https://doi.org/10.1016/j.ijhcs.2008.09.011

Blackwell AF, Petre M, Church L (2019) Fifty years of the psychology of programming. Int J Hum Comput Stud (IJHCS) 131:52–63. https://doi.org/10.1016/j.ijhcs.2019.06.009 (**Special issue for 50 years of the International Journal of Human-Computer Studies. Reflections on the past, present and future of human-centred technologies**)

Boden MA (2008) Mind as machine: a history of cognitive science. Oxford University Press

Bojko A (2013) Eye tracking the user experience: a practical guide to research. Rosenfeld Media, Berlin

Bourdieu P, Passeron J-C (1977) Reproduction in education, society and culture, vol 5. SAGE Studies in Social and Educational Change

Boyd D, Crawford K (2012) Critical questions for big data: provocations for a cultural, technological, and scholarly phenomenon. Inf Commun Soc 15:662–679

Brich J, Walch M, Rietzler M, Weber M, Schaub F (2017) Exploring end user programming needs in home automation. ACM Trans Comput Hum Interact (TOCHI). https://doi.org/10.1145/3057858

Bröder A (1998) Deception can be acceptable. Am Psychol 53:805–806

Bruine de Bruin W, Parker AM, Fischhoff B (2007) Individual differences in adult decision-making competence. J Pers Soc Psychol 92:938–956

Brundage M, Avin S, Clark J, Toner H, Eckersley P, Garfinkel B, Dafoe A, Scharre P, Zeitzoff T, Filar B et al (2018) The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. arXiv preprint http://arxiv.org/abs/1802.07228

Caliskan A, Bryson JJ, Narayanan A (2017) Semantics derived automatically from language corpora contain human-like biases. Science 356:183–186

Cath C, Wachter S, Mittelstadt B, Taddeo M, Floridi L (2018) Artificial intelligence and the 'good society': the US, EU, and UK approach. Sci Eng Ethics 24:505–528

Cheng F-F, Wu C-S (2010) Debiasing the framing effect: the effect of warning and involvement. Decis Support Syst 49:328–334. https://doi.org/10.1016/j.dss.2010.04.002

Chouldechova A, Roth A (2020) A snapshot of the frontiers of fairness in machine learning. Commun ACM 63:82–89. https://doi.org/10.1145/3376898

Christensen L (1988) Deception in psychological research: when is its use justified? Pers Soc Psychol Bull 14:664–675

Cialdini RB (2009) Influence: science and practice, vol 4. Pearson Education, Boston

Cialdini RB, Goldstein NJ (2004) Social influence: compliance and conformity. Annu Rev Psychol 55:591–621

Cook TD, Campbell DT, Shadish W (2002) Experimental and quasi-experimental designs for generalized causal inference. Houghton Mifflin, Boston

Corbett-Davies S, Pierson E, Feller A, Goel S, Huq A (2017) Algorithmic decision making and the cost of fairness. In: 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 797–806

Cowgill B, Dell'Acqua F, Deng S, Hsu D, Verma N, Chaintreau A (2020) Biased programmers? Or biased data? A field experiment in operationalizing AI ethics. In: Proceedings of the 21st ACM conference on economics and computation, pp 679–681

Danks D, London AJ (2017) Algorithmic bias in autonomous systems. In: Proceedings of the 26th international joint conference on artificial intelligence (IJCAI), pp 4691–4697

Dressel J, Farid H (2018) The accuracy, fairness, and limits of predicting recidivism. Sci Adv 4:eaao5580

Dumas JS, Redish J (1999) A practical guide to usability testing. Intellect Books

Erwig M, Smeltzer K, Wang X (2017) What is a visual language? J vis Lang Comput 38:9–17. https://doi.org/10.1016/j.jvlc.2016.10.005 (**SI:In honor of Prof SK Chang**)

Feldman M, Friedler SA, Moeller J, Scheidegger C, Venkatasubramanian S (2015) Certifying and removing disparate impact. In: 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 259–268

Friedman B, Nissenbaum H (1996) Bias in computer systems. ACM Trans Inf Syst (TOIS) 14:330–347

Gärling T, Ettema D, Friman M (2014) Handbook of sustainable travel. Springer

Gianfrancesco MA, Tamang S, Yazdany J, Schmajuk G (2018) Potential biases in machine learning algorithms using electronic health record data. JAMA Intern Med 178:1544–1547

Gilovich T, Griffin D, Kahneman D (eds) (2002) Heuristics and biases: the psychology of intuitive judgment. Cambridge University Press

Graham L, Metaxas PT (2003) Of course it's true; I saw it on the Internet! critical thinking in the internet era. Commun ACM 46:70–75. https://doi.org/10.1145/769800.769804

Grgić-Hlača N, Engel C, Gummadi KP (2019) Human decision making with machine assistance: an experiment on bailing and jailing. Proc ACM Hum Comput Interact. https://doi.org/10.1145/3359280

Hajian S, Bonchi F, Castillo C (2016) Algorithmic bias: from discrimination discovery to fairness-aware data mining. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 2125–2126

Hertwig R, Ortmann A (2008) Deception in experiments: revisiting the arguments in its defense. Ethics Behav 18:59–92

Jobin A, Ienca M, Vayena E (2019) The global landscape of AI ethics guidelines. Nat Mach Intell 1:389–399

Johansen J, Pedersen T, Johansen C (2020) Studying the transfer of biases from programmers to programs. arXiv preprint http://arxiv.org/abs/2005.08231

Johnson GM (2021) Algorithmic bias: on the implicit biases of social technology. Synthese 198:9941–9961. https://doi.org/10.1007/s11229-020-02696-y

Jolls C, Sunstein CR (2006) Debiasing through law. J Leg Stud 35:199–242

Kahneman D (2011) Thinking, fast and slow. Penguin Books

Kahneman D, Knetsch JL, Thaler RH (1991) Anomalies: the endowment effect, loss aversion, and status quo bias. J Econ Perspect 5:193–206

Kimmel AJ (1998) In defense of deception. Am Psychol 53:803–805

Klaczynski PA (1997) Bias in adolescents' everyday reasoning and its relationship with intellectual ability, personal theories, and self-serving motivation. Dev Psychol 33:273–283

Klaczynski PA, Robinson B (2000) Personal theories, intellectual ability, and epistemological beliefs: adult age differences in everyday reasoning biases. Psychol Aging 15:400–416

Klaczynski PA, Gordon DH, Fauth J (1997) Goal-oriented critical reasoning and individual differences in critical reasoning biases. J Educ Psychol 89:470–485

Kumar KK, Geethakumari G (2014) Detecting misinformation in online social networks using cognitive psychology. HCIS 4:1–22

Lakoff G, Johnson M (2008) Metaphors we live by. University of Chicago Press

Lazar J, Feng JH, Hochheiser H (2017) Research methods in human–computer interaction. Morgan Kaufmann

Maloney J, Resnick M, Rusk N, Silverman B, Eastmond E (2010) The scratch programming language and environment. ACM Trans Comput Educ (TOCE) 10:16

Manca M, Fabio P, Santoro C, Corcella L (2019) Supporting end-user debugging of trigger-action rules for IoT applications. Int J Hum-Comput Stud (IJHCS) 123:56–69. https://doi.org/10.1016/j.ijhcs.2018.11.005

Markopoulos P, Nichols J, Paternò F, Pipek V (2017) Editorial: end-user development for the internet of things. ACM Trans Comput Hum Interact (TOCHI). https://doi.org/10.1145/3054765

Mintz AP, Benham A, Edwards E, Fractenberg B, Gordon-Murnane L, Hetherington C, Liptak DA, Smith M, Thompson C (2012) Web of deceit: misinformation and manipulation in the age of social media. Information Today, Inc.

Mittelstadt BD, Allo P, Taddeo M, Wachter S, Floridi L (2016) The ethics of algorithms: mapping the debate. Big Data Soc 3:2053951716679679. https://doi.org/10.1177/2053951716679679

Moscovici S, Faucheux C (1972) Social influence, conformity bias, and the study of active minorities. Advances in experimental social psychology, vol 6. Elsevier, pp 149–202

O'Neil C (2016) Weapons of math destruction: how big data increases inequality and threatens democracy. Broadway Books

Obermeyer Z, Powers B, Vogeli C, Mullainathan S (2019) Dissecting racial bias in an algorithm used to manage the health of populations. Science 366:447–453

Oliver RL (2014) Satisfaction: a behavioral perspective on the consumer. Routledge, London

Paternò F, Santoro C (2019) End-user development for personalizing applications, things, and robots. Int J Hum Comput Stud (IJHCS) 131:120–130. https://doi.org/10.1016/j.ijhcs.2019.06.002 (**Special issue for 50 years of the International Journal of Human-Computer Studies. Reflections on the past, present and future of human-centred technologies**)

Pedersen T, Johansen C (2019) Behavioural artificial intelligence: an agenda for systematic empirical studies of artificial inference. AI Soc. https://doi.org/10.1007/s00146-019-00928-5

Pedersen T, Friman M, Kristensson P (2011) Affective forecasting: predicting and experiencing satisfaction with public transportation. J Appl Soc Psychol 41:1926–1946. https://doi.org/10.1111/j.1559-1816.2011.00789.x

Pedersen T, Johansen C, Jøsang A (2018) Behavioural computer science: an agenda for combining modelling of human and system behaviours. HCIS 8:1–20. https://doi.org/10.1186/s13673-018-0130-0

Rauber A, Trasarti R, Giannotti F (2019) Transparency in algorithmic decision making. ERCIM News 1:10–11

Reinecke K, Bernstein A (2011) Improving performance, perceived usability, and aesthetics with culturally adaptive user interfaces. ACM Trans Comput Hum Interact (TOCHI). https://doi.org/10.1145/1970378.1970382

Resnick M, Maloney J, Monroy-Hernández A, Rusk N, Eastmond E, Brennan K, Millner A, Rosenbaum E, Silver J, Silverman B et al (2009) Scratch: programming for all. Commun ACM 52:60–67

Rogers Y, Sharp H, Preece J (2011) Interaction design: beyond human–computer interaction, 3rd edn. Wiley, New York

Schlesinger A, O'Hara KP, Taylor AS (2018) Let's talk about race: identity, chatbots, and AI. In: Conference on human factors in computing systems CHI '18. ACM. https://doi.org/10.1145/3173574.3173889

Schultz PW, Nolan JM, Cialdini RB, Goldstein NJ, Griskevicius V (2007) The constructive, destructive, and reconstructive power of social norms. Psychol Sci 18:429–434

Silva S, Kenney M (2019) Algorithms, platforms, and ethnic bias. Commun ACM 62:37–39

STOA (2019) A governance framework for algorithmic accountability and transparency. European Parliamentary Research Service (EPRS). STOA Scientific Foresight Unit—Panel for the Future of Science and Technology

Thaler RH, Sunstein CR (2009) Nudge: improving decisions about health, wealth, and happiness. Penguin

Thibodeau PH, Boroditsky L (2011) Metaphors we think with: the role of metaphor in reasoning. PloS one 6(2):e16782. https://doi.org/10.1371/journal.pone.0016782

Townsend CB (2003) The curious book of mind-boggling teasers, tricks, puzzles and Games. Sterling Publishing Company

Trice AD (1986) Ethical variables? Am Psychol 41:482–483

Trower J, Gray J (2015) Creating new languages in blockly: two case studies in media computation and robotics (abstract only). In: 46th ACM technical symposium on computer science education

SIGCSE '15. ACM, pp 677–677. https://doi.org/10.1145/2676723.2691916

Tulving E, Schacter DL (1990) Priming and human memory systems. Science 247:301–306

Tversky A, Kahneman D (1974) Judgment under uncertainty: heuristics and biases. Science 185:1124–1131

Ur B, Pak Yong Ho M, Brawner S, Lee J, Mennicken S, Picard N, Schulze D, Littman ML (2016) Trigger-action programming in the wild: an analysis of 200,000 ifttt recipes. In: Proceedings of the 2016 CHI conference on human factors in computing systems CHI '16. ACM, pp 3227–3231. https://doi.org/10.1145/2858036.2858556

Vaccaro M, Waldo J (2019) The effects of mixing machine learning and human judgment. Commun ACM 62:104–110. https://doi.org/10.1145/3359338

Weintrop D, Wilensky U (2017) Comparing block-based and text-based programming in high school computer science classrooms. ACM Trans Comput Educ (TOCE) 18:3

Weiss DJ (2001) Deception by researchers is necessary and not necessarily evil. Behav Brain Sci 24:431–432

Wilson TD, Gilbert DT (2003) Affective forecasting. Advances in experimental social psychology, vol 35. Academic Press, New York, pp 345–411. https://doi.org/10.1016/S0065-2601(03)01006-2

Yonelinas AP (2002) The nature of recollection and familiarity: a review of 30 years of research. J Mem Lang 46:441–517

Zou J, Schiebinger L (2018) AI can be sexist and racist—it's time to make it fair. Nature. https://doi.org/10.1038/d41586-018-05707-8

Zuboff S (2019) The age of surveillance capitalism: the fight for a human future at the new frontier of power. Profile Books