Explaining Epistemic Opacity
Ramón Alvarado

Introduction

Conventional accounts of epistemic opacity, particularly those that stem from the definitive work of Paul Humphreys, typically point to limitations on the part of epistemic agents to account for the distinct ways in which systems, such as computational methods and devices, are opaque. They point, for example, to the lack of technical skill on the part of an agent (Burrell, 2016; Kaminski, 2017), the failure to meet standards of best practice (Saam, 2017; Hubig, 2017), or even the nature of an agent (Humphreys, 2009) as reasons why epistemically relevant elements of a process may be inaccessible. In this paper I argue that there are certain instances of epistemic opacity— particularly in computational methods such a computer simulations and machine learning processes—that (1) do not arise from, (2) are not responsive to, and (3) are therefore not explained by the epistemic limitations of an agent. I call these instances agent-neutral and agent-independent instances of epistemic opacity respectively. As a result, I also argue that conventional accounts of epistemic opacity offer a limited understanding of the full spectrum of kinds and sources of epistemic opacity, particularly of the kind found in computational methods. In particular, as I will show below, the limitations of these accounts are reflected in the way they fail to provide satisfactory explanations when faced with certain instances of opacity.

Humphreys' (2004; 2009) seminal definitions are by now commonly accepted as agent-based definitions (Symons and Alvarado, 2016; Alvarado, 2020; San Pedro, forthcoming).[1] That is, they seek to provide an account of epistemic opacity in virtue of an agent's epistemic limitations. Therefore, as I will show here, they are limited in this respect vis-à-vis instances of agent-independent instances of opacity. However, more recent accounts of epistemic opacity are also limited in that they fail to recognize —or in some instances deliberately seek to explain away— the fundamental differences in kind between *essential opacity* (Humphreys, 2009) and a more general and circumstance-relative epistemic opacity. Hence, taken together, I argue that conventional accounts of epistemic opacity, both old and new, offer a limited understanding of the phenomenon of epistemic opacity in that

a) They are all agent-based,
b) So far they have failed to account for differences in kind of opacity, and
c) They are not adequately explanatory of the phenomenon itself in important instances.

Let us briefly examine each one of these issues. First, conventional accounts of epistemic opacity are limited in that they fail to account for the fact that there are instances of epistemic opacity that are not explained by agential limitations regarding resources, practices, and/or efforts. So, not all instances of epistemic opacity are agent-based.

[1] This particular aspect of Humphreys' definitions was often overlooked due to the fact that epistemic opacity, as an epistemic phenomena was understood to always involve an epistemic agent. However, what was often neglected was that just because an epistemic phenomena requires an agent, doesn't mean that the agent is the cause and/or the source of the phenomena. As we will see below, this is an extremely valuable distinction in this context.

Secondly, unlike Humphreys' (2004; 2009) original definitions, recent efforts to provide a comprehensive spectrum of epistemic challenges in computational methods have sought to explain away the fact that Humphreys' second and stronger formulation of epistemic opacity is not only a stronger version of the same phenomenon, but rather a *different kind* of epistemic opacity. In particular, these recent efforts (Kruger et al, 2020; San Pedro 2021) have either failed to recognize and/or sought to actively minimize the importance of an *impossibility clause* in Humphreys' stronger definition. In other words, they offer a deflationary account of essential epistemic opacity as something that could be mitigated through resourceful strategies. However, it is this particular clause regarding impossibility that posits that while some instances of epistemic opacity may be minimized or overcome by changing the circumstances of an agent, some other instances offer an *impossible challenge* to some agents *given their nature*. This is both an important and inescapable contribution of Humphreys' early insight into the phenomenon of epistemic opacity. Importantly, the phenomenon identified by Humphreys is also not something that can be wished away with better practices.

Finally, while the insights provided by conventional accounts of epistemic opacity—both of the original definitions offered by Humphreys and more recent efforts—have proven extremely useful in the understanding of some important epistemic challenges posed by computational methods, these accounts suffer from an explanatory limitation: because they are all agent-based accounts of epistemic opacity— that is, because they all make reference to agential circumstances and limitations— they lack the conceptual framework to explain the features of the world, other than the agent, that often contribute, in a very relevant sense, to the reasons why something is or is not opaque. In this sense, they fail to properly explain the reasons why, the sources from which, and the nature of the emergence of epistemic opacity vis-à-vis certain processes. This is particularly the case in the context of computational methods such as computer simulations and machine learning processes in which, as I will show, opacity can sometimes emerges *in virtue of* something other than the agent's epistemic resources, efforts, and/or limitations.

Kinds of epistemic opacity

Broadly, the term epistemic opacity refers to the lack of access that an inquiring agent may have to the relevant aspects by which a process achieves whatever task it is supposed to achieve. In other words, an epistemically opaque process is a one whose significant details about its functioning are not available to someone seeking to understand it, make use of it, or someone who is subject to the results emerging from the fulfillment of its task.

Humphreys succinctly defines epistemic opacity the following way:

"A process is epistemically opaque relative to a cognitive agent X at time t just in case X does not know at t all of the epistemically relevant elements of the process." (2009)

As can be noted, defining epistemic opacity this way allows for a very broad interpretation of the processes that can be characterized as epistemically opaque. We can, for example imagine many processes to be epistemically opaque to many people at many times: the processes in your computer, the processes by which books are bound, the processes of bending wood can all be opaque to some of us. These processes are epistemically opaque in so far as the relevant elements by which they fulfill their task or arrive at a result are not accessible to someone in their capacity as a knower. Consequently, a process that is generally opaque in the way defined above is also only opaque to some people but not to others. And, furthermore, just as these type of processes are opaque to some people but not others, such processes can also cease to be opaque to a single person at a different time. In other words, the main feature of this account of epistemic opacity is that it is very generally applicable. It is, in other words a general and relative account of epistemic opacity as it is relative to a specific agent the agent's circumstances at a specific time (Humphreys, 2004; Symons and Alvarado, 2016; Alvarado, 2020).

These three points above make the concept of general epistemic opacity not only too broad but also not very informative concerning the particular causes and sources of epistemic opacity, particularly when it comes to epistemic opacity in computational methods. The definition basically says that whenever an agent is in a circumstance and/or a time in which a process' epistemically relevant elements are not known, then the process is opaque, which is most of the time for most of the processes for most of the agents. As defined, the concept seems to capture every single instance of an agent facing a process previous to knowledge-acquisition. That is to say, before anybody knows anything about something that something can be characterized as opaque in the manner defined above. One can, for example imagine that at every point in someone's lives some things will be epistemically opaque without representing a significant epistemic challenge. This makes the concept seem a little trivial. In other words, almost every single process a growing human will come to understand was at some point epistemically opaque to them. It is in this sense too that the concept may be mistaken as not very philosophically interesting. After all, some things are unknown to some of us at some point, known to others at the same time, and can become known at another time by the same agents to which it was unknown. However, while opaque may sometimes imply unknown, unknown does not necessarily imply opaque. Pointing to something as being epistemically opaque in the manner defined above does not provide any interesting and or peculiar information about the system in relation to anything else. This is particularly the case if what one is trying to understand and say something about is a specific piece of technology.

Because of this last point, in the context of computational methods deployed in inquiry, philosophers (Kaminski et al., 2018) and sociologists of science and computation (Saam, 2017) have tried to be more specific about the distinct ways in which a particular computational processes, such as computer simulations and machine learning algorithms, may come to be opaque, as well as to whom it may be opaque to and when it may be opaque to them. These accounts, which I will examine in some detail below, have correctly identified the importance of identifying not only the systems that are opaque but also the importance of identifying the sources, reasons, and/or instances *in virtue of* which the system is opaque.

Below are a few examples of more detailed accounts of opacity that still can be categorized as general and relative as explained above. They are also very closely related to social aspects to the design, development and/or deployment of computational technology. Hence, I group them all in what some have called 'social opacity' (Burrell, 2016; Kaminski et al, 2017).

Social epistemic opacity

Sometimes social aspects contribute to making a process, set of processes, and/or a device epistemically opaque to someone. In these instances, what is at play is a kind of social epistemic opacity (Kaminski et al., 2017; Burrell, 2016). I will use this term to refer to any and all instances of epistemic opacity which arise in virtue of social features of a given circumstance. There are, however, distinctions within social opacity regarding the source of opacity. Social factors can be of many kinds. Division of labor, proprietary limitations on access, intentional secrecy, etc. all contribute to making the workings of a system, process or device inaccessible to some while accessible to others, and hence generally, and relatively epistemically opaque as defined above. Kaminski and others, for example (Saam, 2017; Hubig, 2017), point towards the fact that a non-trivial number of engineers and computer scientist are required in order to put together a successful computer model of the kind that simulates complex phenomena such as the weather. This makes knowing all the functional details of the process of simulation simply inaccessible to any one of the individuals involved. Hence, some of the epistemically relevant processes in a computer simulation of weather phenomena are epistemically opaque to many, if not all, of the individual agents involved in its creation.

This problem of course is not new, nor is its identification as a problem. Helen Nissenbaum (1990), for example, identified the difficulties to ascribe moral responsibility to any one agent in particular in the context of computational technology, which is often developed by vast teams belonging to (sometimes distinct) corporate structures. Knowing exactly who did what would prove an intractable task. She called this issue the "many hands problem" in responsibility ascription (Nissenbaum, 1990).

In a similar manner, the vast structural division of labor in massive computational projects, makes is so that thoroughly understanding the totality of the underlying functionality is not practically viable. This, of course, applies not just to the many hands involved in the design and development of a single piece of technology but importantly to the fact that each piece of technology that is used in the computational process is also made of many different devices (Simoulin, 2007; 2017). Moving away from the question of moral responsibility and towards the question of epistemic access to relevant elements of a system this is just a "many everything and anything problem" in modern technological devices, particularly those involving computational elements (Alvarado, 2020): many hands, many devices, many functions, etc., make it so that knowing the relevant epistemic elements of the process (in this case, whatever task these computational devices carry out) is generally opaque to most agents involved at most times.

Jenna Burrell (2016) on the other hand provides an even more detailed account of the epistemic opacity at play in what she calls socially consequential contexts. In her account one can find three different sources of epistemic opacity, two of which can be directly mapped with our discussion above concerning the social sources of such inaccessibility to relevant elements of a particular system. For her, there are two kinds of socially explained epistemic opacity. The first instance of opacity arises from the simple fact that no single individual can know everything and/or can know anything a priori (as I have also specified above). Rather, there will be specialization and there will be technical training to elucidate even things one is familiar with. This is called by others, particularly Kaminski et al. (2017), 'technical opacity': some systems are opaque simply because of the level and complexity of the technical details related to their functioning. This is in a sense why technical education is required and why there are many and diverse experts involved in a single, seemingly homogenous project, such as computer engineering. The second kind of social opacity in Burrell's work refers to the kind of opacity that arises from intentional barriers to access to the internal processes of a given technology. Here, very straightforward circumstances explain why a device and/or process may be kept from some people: someone wants to keep it that way. Institutional secrecy is often cited as the origin of this kind of deliberate social epistemic opacity.[2]

It is important to note here, that the different sources of opacity so far discussed above, even the different sources of social opacity, are all relative to an agent, to an agent's circumstances and may all be simultaneously at play in a given setting. This in itself makes a circumstance in which opacity arises slightly a bit more problematic than a simple instance in which only one source of general opacity is present. However, computational processes, such as the ones required to build successful computer simulations and/or the ones deployed to execute sophisticated analytics such as machine learning, may also prove to be opaque in an even more problematic way. A given device, method or process can be opaque simply on the basis that its inner processes are beyond what any given individual agent can reasonably track and/or understand (Humphreys, 2004; 2009; Symons and Alvarado, 2016; Burrell, 2016; Alvarado 2020). A system, device, or set of processes that is opaque in this manner—beyond the epistemic resources not just of a given individual at a given time but of any one individual *of the same kind* at *any time*— is what Humphreys called essential epistemic opacity (2009). Details of this account follow below.

Essential epistemic opacity

According to Humphreys, a system, process or device is *essentially* epistemically opaque

"iff it is *impossible, given the nature of X*, for X to know all of the epistemically relevant elements of the process." (2009)

Let us take a moment to look at the details in Humphrey's account of essential opacity. Given recent attempts to explain away the difference in kind of epistemic opacities (See San Pedro, in this issue), a question arises as to whether the difference between the general account of epistemic opacity described above and essential opacity is a matter of kind or of degrees. Here I argue that their difference is a difference in kind. That is, essential opacity is not just a stronger version of the general opacity described above. Rather, it is a different kind of opacity. This is made clear by two main factors: the first one is that it is the kind of opacity that arises *in virtue of the agent's nature* and not in virtue of something else.[3] Essential opacity arises in virtue of

[2] As we will see below, it is clear that just calling all of these instances of opacity "social" opacity, is a bit of a broad stroke. Afterall, there are significant differences between the opacity that arises from the division of labor and the one that arises in virtue of institutional secrecy. They may in fact be different kinds of opacity, except for the fact that they are both relative and general.

[3] It is the appeal to the nature of an agent that makes this definition 'essential'. In other words, one can interpret the opacity in question as having arisen in virtue of the essence of the agent's epistemic nature.

something other, something more essential, than general opacity. It is also conceptually and practically distinct. Solving or minimizing cases of general opacity do not in any way, shape or form solves or minimizes essential epistemic opacity. Secondly, as the definition of essential opacity suggests, *given the agent's nature*, it is *impossible* for that agent to access the relevant epistemic elements of that process, system and/or device. This impossibility is a markedly distinct feature of essential epistemic opacity. I argue that even if one can compare one challenge with another, the fact that one is a challenge that allows for a possible solution and the other one is impossible to overcome makes one a different kind of challenge, namely: an impossible challenge.[4]

Both items—the agent's nature and the impossibility related to it— in the definition are important enough for Humphreys to have felt the need to formulate a second, independent, definition of epistemic opacity. So, let us spend a few lines on each of those clauses, starting by the distinction between a kind of opacity arising *in virtue of an agent's circumstances* and/or time of appraisal, and another instance of opacity arising *in virtue of an agent's nature*.

Degrees, kinds and impossibility in essential epistemic opacity

The first issue, concerning two challenges being distinct given that they emerge in virtue of distinct sources is fairly straightforward. It is easy to conceive, for example, that problems emerging in virtue of two distinct things are two different kinds of problem, even when the context in which they emerge is somewhat similar.[5]

Consider the following situations and the challenges associated with remedying them. When mathematicians, philosophers and craftsmen were trying to assess the reliability of early telescopes they faced many different kinds of challenges. One of the contexts in which these different kinds of challenges came to the fore was not only to do with the instrument per se, but rather with the development of the tests to test the instrument (Van Helden, 1994). While craftsmen were focused on the accuracy of the instrument and the development of optical techniques for appropriate lenses, a different kind of challenge arose. What testers found out as they were testing their apparatus out in the field was that what was on the other side of the test was of equal importance than what was inside the instrument. Even when the lenses of the telescope were not really accurate, some of the people taking the test could accurately describe what was on the other side. This was because early tests involved known texts and a pool of participants that were well-read and therefore familiar with such texts. So they could just guess from the first couple of words what followed. When the testers changed the text to display a random assortment of letters, they found out a similar problem arose given the prevalent features of some fonts: people could still discern between different letters on the other side of the field even when the lenses were not very good. So, lens craftsmen and those early natural philosophers who were focused on the nature of light and glass had to partner up with people in

[4] As we will see below, the impossibility at the heart of this definition of epistemic opacity is non-trivially debatable (San Pedro, this issue; Duran and Formanek, 2017; Duran, 2018). I will argue that even a weaker version of impossibility is sufficient to mark the phenomenon of essential epistemic opacity as a different kind of opacity from that of the general account.

[5] Please see footnote 2.

the paper-making industry as well as with those in the printing press industry. This was done in order to solve the issue of reliability of a new instrument.

Here were two challenges: one, make a glass lens and an apparatus to carry the light in such a way that the other side of a field becomes clearly visible to someone looking through the apparatus; and two, develop a testing methodology that includes appropriately vetted materials to avoid biased testing. Both challenges were related to the reliability assessment of the telescope. Both challenges could be compared with regards to difficulty. It may have been more or less difficult to produce an accurate lens than to design a less salient font or a less reflective paper. But there is nothing to suggest that they are the same challenge, or even the same kind of challenge.

In a sense, solving one challenge may prove more difficult than solving the other, and in a scale of solvability of challenges one problem may be at one end of the spectrum and the other one at the opposite side. However, this is not indicative of both being the same kind of problem nor of their difference being solely a matter of degree. One challenge arose in virtue of the nature of glass and the instruments meant to guide light into someone's eye. The other challenge emerged in virtue of the knowledge that agents had and in virtue of the material with which they were presented. In order to solve one of the problems, glass, the nature of glass and the behavior of light had to be addressed. The other challenged had to be addressed taking into consideration the salient features of words and letters as well as the degree of erudition on the part of the participants. The source as well as the relevant features required to consider in order to assess the emergence of the problem and/or the solving of each one of the problems were distinct. So is the nature of each of the problems.

Similarly, when Humphreys suggests that a general sense of epistemic opacity arises in *virtue of* an agent's epistemic position in relation to a process at a certain time t, and that an essential sense of opacity emerges in *virtue of the nature* of the agent, he is talking about *two different kinds of epistemic opacity* and not just two different degrees of opacity. Of course, if one is so inclined, one can construct a broadly encompassing spectrum comparing the degree of difficulty *of different kinds* of epistemic challenges. As expected, it will turn out that insofar as one of the challenges is impossible to overcome and the other one is possible to overcome, the former will be at the end of the spectrum of difficulty while the latter will be somewhere prior. However, this does not mean that the difference between both challenges is merely a matter of degrees. Their difference of degrees lies only in that we have built a spectrum that admits of *different kinds* of challenges and in that they are being compared in relation to a shared property: difficulty to solve. But even here, there is something suspect already. It is disingenuous too assume that something that is not solvable should be included within the brackets containing members of the class of solvable items. If one were strictly categorizing challenges in terms of solvability, there is no reason to include challenges that are by definition not solvable, i.e. impossible. Unless, of course, one is already questioning and/or dismissing their solvability. This takes me to the second feature of Humphreys definition of essential epistemic opacity: the impossibility clause.

Humphreys, (2009) suggests that essential epistemic opacity is the kind of opacity that doesn't just obstruct an agent from knowing the relevant elements of a system or process, but rather makes it impossible for such kind of agent to know the epistemically relevant elements of that process. This obstruction is different from the one in the previous definition in that the task is, well, impossible. This impossibility is, of course, as mentioned above, in virtue of the nature of the agent— its essence so to speak.[6]

There is much to say about the nature of the impossibility which goes beyond the scope and space of this paper. There is much to be said about what exactly this possibility entails. It is unclear for example if Humphreys is here referring to metaphysical, logical, or another kind of impossibility. Durán and Formanek (2018) rightly point that in the context of essential epistemic opacity, For Humphreys, "eligible possibilities are restricted by human action, which takes place in finite time and space and is constrained by final mental capabilities." (p.651). This is what they call a 'common sensical' reading of possibility. What this entails is that even if one is unable to provide a full fleshed impossibility theorem regarding the possibility of an agent, any kind of agent, ever having access to the relevant epistemic elements of a given process in all possible worlds, one can for all intents and purposes define this impossibility in terms that are constrained to this particular world, with the physical and temporal dimensions that it has. In this world, humans are finite agents with finite resources. So are most other agents, and so one can easily speak of challenges as impossible if they are practically so for an agent such as us to overcome in any reasonable amount of time.[7] For Humphreys, an example of these type of challenges was the opacity at play in computational processes such as computer simulations.

The point above is in sharp contrast with recent attempts at explaining away the impossibility clause in Humphreys definition of essential epistemic opacity. San Pedro (in this issue), for example, suggests that given that an agent's nature is, all things considered, itself a contingent feature of the world, then the definition of essential opacity only refers to a special case of a weaker version of epistemic opacity (general epistemic opacity above). The reasoning is the following: just like the circumstances of a given agent at time t can be what makes a process opaque and can be changed at a different time making the process either not opaque or less opaque in the general definition, so is the nature of an agent. In other words, the process is only opaque to a given agent given the agent's nature, change the nature of the agent or change the agent and the process will cease to be opaque or will become more or less opaque. San Pedro grants that changing an agent's nature is not as easy as changing the circumstances of an agent. That is why San Pedro also grants that overcoming essential epistemic opacity is harder. As we will see below, accounts such as this risk neglecting Humphreys point in calling such instances of opacity *essential epistemic opacity*: there is an impossibility to it and such impossibility is tied to the essence of something. These two elements of the definition are meant to capture the fact that if you change the nature—i.e. essence—of that something then it ceases to be that something

[6] While it may prove strange to some practitioners and pragmatist philosophers for Humphreys to use the terms and concepts of impossibility, nature and/or to elicit the ghosts of essentialist metaphysics, that Humphreys does this as a philosopher is not strange at all. This is particularly evident if one considers his body of work concerning non-anthropocentric epistemologies: in his book *Extending Ourselves* (2009) , for example, he speculates about a science neither for us or by us. Similarly, in his paper "Network Epistemology" (2010), he posits the possibility of knowledge *by* complex objects such as networks. This is another instance in which it is worth noting that Humphreys' use of this term is neither trivial nor superfluous but rather deliberate.

[7] As we will see below, one can do the same thing even considering a broadly extended version of an agent, particularly when that agent, any agent, is faced with challenges that— given the physics, biology and temporal dimension of this world— would take more than the age of the universe to overcome.

and the fact that even if you do change the nature of that something, access to the epistemically relevant elements of a same process *continues to be impossible for the same agent, prior to being changed*. Hence, there are instances of epistemic opacity that are impossible to overcome for an agent in virtue of that agent's nature. If you change the agent or their nature you did not change the impossibility, you only changed the problem.

In other words, there are instances of *essential* epistemic opacity and they can occur when some kinds of agents encounter some computational processes. Deflationary accounts of essential epistemic opacity fail to capture the *essential* aspect of essential epistemic opacity. What approaches of this kind may have been able to show is that Humphreys initial assertion that "perhaps all of the features that are special to simulations are a result of this inability of human cognitive abilities to know and understand the details of the computational process. The computation involved in most simulations are so fast and so complex that no human or group of agents can in practice reproduce or understand the process" (Humphreys, 2009) may have been too fast, too broad, and/ or too early stated. It may very well be the case that at least *some* computer simulations, and/or some instances of opacity, that he thought were essentially opaque were not. But this only shows that some kinds and instances of epistemic opacity in computer simulations may not be of the essential kind. This does not take away, however, from the philosophical enterprise of identifying two distinct kinds of epistemic opacity and/or of identifying a second, different and insurmountable, kind to emerge in the context of computational methods such as software-intensive artifacts— and by extension computer simulations, machine learning methods, etc. (Symons and Horner, 2014; Symons and Alvarado, 2016; Alvarado, 2020).

Agent-based, agent-neutral, and agent-independent epistemic opacity

In this section I argue that there are instances of epistemic opacity that are either neutral to and/or independent from the limitations of agents. That is, they arise in virtue of factors that are not responsive to or are not related to agential resources. In so far as the general account of epistemic opacity is based on (defined/explained by appeal to) an agent's circumstances, and insofar as the account of essential epistemic opacity is based on an agent's nature, then the presence or absence of epistemic opacity will be determined by an agent's limitations. They are agent-based accounts of epistemic opacity (Alvarado, 2020 p. 179 note 1). Hence they offer only an incomplete account of the phenomenon of epistemic opacity. Importantly, both agent-neutral and agent-independent instances of epistemic opacity, as I will show through some examples below, are often to be found in relation to computational methods such as software-intensive instruments, processes and methods that include computer simulations and machine learning techniques.

In particular, the accounts of epistemic opacity examined in the sections above are limited in the following two respects. The first one is that, as I will show, they fail to recognize and account for important instances of opacity in computational methods that have little to nothing to do with agential limitations. The second limitation is that as far as explanations of a phenomena go, agent-based explanations in general are unsatisfactory when agential features are not what account for what we are trying to understand. What I mean by this will become clearer in the paragraphs below.

Before I move on with an analysis of the limitations of agent-based accounts of epistemic opacity, it would be useful for me to provide an account of what an agent-neutral instance of epistemic opacity would look like. This is because, as far as accounts of epistemic opacity go, agent-neutral instances of opacity can still be formulated in relation to agential limitations. In this sense, they represent a sort of preliminary conceptual detachment from the agent-based accounts examined above and a first step towards understanding the agent-independent accounts I will describe at the end of this section.

So, here is a working definition. Some instances of epistemic opacity are agent-neutral

iff it is impossible, in virtue of a feature F of the nature of X which is also shared by all other epistemic agents W, for X (and by extension W) to know the epistemically relevant elements of a process.

As we can see from this definition, essential epistemic opacity, as described in the previous section is still a comprehensive enough formulation to accommodate for the fact that even agent-neutral instances of opacity are in a strong way connected to a feature of *the nature of the agent(s)*. However, there is something extra to the agent-neutral explanation. It would be misguided, for example, to assert that all the instances of opacity in such cases are *due to the nature of X* when X stands for an individual/specific kind of agent and the opacity arises only because of a property (F) shared by all cognitive/epistemic agents agents.

When it comes to software-intensive artifacts, such as computer simulations and/or machine learning processes, here is an example of an instance of opacity from Symons and Horner (2014) that applies in the way defined above.

We can begin by conceiving of a small software program consisting of only 1000 lines of code. In conventional software practices, on average 1 out of 10 of those 1000 lines of code will be a command of the form 'if/then/else'. This is in contrast to commands that follow the form 'if/then'. That is the former, but not the latter include an extra alternative bifurcation of possible paths the code can follow. Checking a similar size program that only had if/then commands can be done by examining each one of the 1000 lines of code. Including an if/then/else command, however, bifurcates the possible paths that the code can take. This increases the lines of code that have to be checked. Consider that a program with 10 if/then command lines has 10 lines to check. The same program with one if/then/else command will have 20 lines. If we have a program with 1000 lines of code, which includes an average of 1/10 if/then/else commands will have so many lines of code to check that it will take several times the age of the universe to do so (Symons and Horner, 2014; Horner and Symons 2014; 2018; Symons and Alvarado, 2016). Software systems involved in computational methods such as machine learning and computer simulations of scientific phenomena by far exceed the number of lines of code used in this example.

A task that would take anywhere near the age of the universe to be completed is in fact a process that will be opaque to many more agents than just humans and our enhancing technologies, including computer simulations and machine learning methods. The process as described is evidently essentially epistemically opaque in so far as the agent in question is us. That much is clear. But this instance of epistemic opacity is also *agent-neutral* in the following way. It doesn't matter which agent takes on the task. It will remain opaque even for those agents that do have millions of years to spare. While the opacity in such a system is related to the fact that most, if not all, epistemic agents will be finite and/or younger than the universe, it is not significantly reduced or enhanced by any change in the agent's capacities or resources. This is what makes it agent-neutral. The opacity arises in virtue of feature F, mainly X's finite nature. Yet, being finite, is a property of most—if not all—epistemic agents. Therefore, this kind of opacity is not responsive to agential resources, capabilities or efforts. You can replace any other agent with any other resources and the outcome will still be the same. Details about the agent do not enhance or minimize the fact that the process is opaque in any significant manner. Even if we imagine a breakthrough in computing processes that could significantly increase the speed at which testing software could be done, reducing the time of that task (assessing the reliability of the code in a reliable manner [8]) from several times the age of the universe to half that time: the process will remain opaque to any agent that is younger than or equal in age to the universe (Symons and Horner, 2014). So agential resources, such as time, computing capabilities, etc. do not in any way, shape or form have any significantly effect on the opacity of the process. Again, it is an instance of agent-neutral opacity.

Now that we have an idea of what an agent-neutral kind of epistemic opacity would look like, we can begin to see a disconnect between the kinds of opacity that appeal to an agent's nature and/or circumstances, and the kind of opacity that arises from and/or is explained by factors that have little to no bearing on agential limitations. So far, this agent-neutral account of opacity, as described above, only captures the fact that some instances of opacity, even when arising in virtue of some agent-related features, will be unresponsive to agent-related efforts and resources.

---

[8] Exhaustive testing of code lines is only one way of assessing reliability in software systems. A more used and useful approach would be to conduct tests on random samples of the code. By using conventional statistical inference theory, one can randomly select samples of code and test their execution. By inference, and with the assumption of random distribution of error in the system, a reliability assessment of the samples should give us an idea of the rate of error in the whole system. If 20 sample lines out of a 100 fail to do what they are supposed to do then we can say that the system as a whole has a 20% failure rate. However, there are two main issues with this approach. Even before software systems reached the size and complexity seen today, it was well established that the nature and distribution of error in software systems is not like that of other systems (Parnas et al., 1990 p.638). This is because of two reasons. The first one is that while in other engineering projects one can assume certain resilience to small errors, in software even a punctuation error can be catastrophic (1990, p.637). Further, the kinds of error that that can affect other engineering projects can be assessed by assuming that these errors are "not highly correlated" (1990 p.638). Because of the interconnectedness of software functioning and the fact that many errors are due to design (Floridi, Fresco and Primiero, 2014), errors in software cannot be considered statistically independent (Parnas et al., 1990) nor can they be considered randomly distributed (Parnas and Kwan, 1990; Horner and Symons, 2014; 2016). Thus, deploying statistical inference techniques to test samples under the assumption that the errors found will be randomly distributed is an invalid approach when it comes to software. Further, even if this technique were valid, consider the following. If the number of lines of code to be tested is of the order of magnitude postulated in the example above, what constitutes a significant sample set? Testing even 5% of the lines of code of a system with 100,000 lines of code and an average path complexity of one if/then/else bifurcation per every 10 lines would put us in the same position as with the original example. It would take ages to do. If we managed to test 1% of the lines it would be difficult to say that we now have a dependable assessment of the system's reliability. Any feasible sample is too big a sample and we fall back to the path complexity challnege described by Symons and Horner.

This account does not capture the fact that some instances will also have the added feature of *arising from* features of the world not related to agents.

It is important to note at this point that agent-based accounts of a phenomena (in general) are only satisfactory accounts of that phenomenon in those instances in which something about an agent has something to do with the emergence of the phenomenon in question. Recognizing this point is extremely important for it implies that if a given phenomenon arises from— is caused by, emerges in virtue of— something other than anything related to the agent, then it will not be properly accounted for by an exclusively agent-based account of that phenomenon. This applies to accounts concerning the *reasons why* something is difficult to achieve. When we consider, for example, the difficulty of solving Navier-Stokes equations it would be misguided to assert that they are difficult to solve merely *because* or *in virtue of* the fact that we are limited epistemic agents while ignoring integral aspects of the equations themselves that contribute to their status as difficult ones to solve. They are not merely difficult for us. In a spectrum of difficulty in which other equations are compared (linear, etc.) these equations would be more difficult than the others. If the Navier-Stokes equation is hard for a baby to solve, for example it *is not because* it is a baby trying to solve it. It is not in virtue of the baby's status as a baby. Similarly, if such kind of equations are difficult for a trained physicist to solve it is not *because* of the physicist's limitations.[9]

Just to be clear on this point, here's another simple toy example contrasting two possible accounts of the same challenge. Imagine that a child asks you why we cannot see into the future. You come up with three possible explanations for this limitation, all of which are in fact true. The first one is that we can't see into the future because we, personally, do not have the capacities, abilities, or tools 'required' to look into the future. This explanation would look something like this: "we cannot look into the future *because* you and I particularly are limited in such a way that we can't look into the future." This implies that perhaps other humans can. The second explanation you come up with is that we cannot see into the future *because* we as humans in general do not have the capacities, abilities, or tools 'required' to look into the future. This implies that perhaps other epistemic agents can. And the third explanation you come up with is that the future is the kind of thing that hasn't yet happened, and things that haven't happened cannot technically (logically, metaphysically, etc.) be 'seen into'. Though all explanations are true in a way, two are agent-based while the other one is agent-neutral. I suggest that as far as accounts of the reasons for why a challenge is a challenge go, the latter is a more satisfactory account. It tells you something about the phenomenon in question (the future) that explains our inability to access it. In order to see why this is a more satisfactory answer more clearly, we can remind ourselves of the reason why a Navier-Stokes equation is difficult to solve for a baby. It is not *because* of the baby's status as a baby.

---

[9] The problem regarding the difficulty of solving Navier-Stokes is not just whether or not there exists an easy way to solve the equations that we just don't know about yet. The philosophical problem of explaining—accounting for—the difficulty in solving the problem isn't solved that way. If someone were to ask, in a not too distant future in which an easier solution had been found, why humans had such trouble solving these type of equations it would still be misguided to simply state that it was *because* human beings were limited in such and such way without noting anything about the equations themselves that made them hard to figure out.

As we will see below, in fact the third explanation is more than just agent-neutral. It is agent-independent in that the reason why we can or can't see into the future has nothing to do with the agents in question. It has to do with the features of the phenomenon we are inquiring about: namely, the future.

What agent-based accounts of epistemic opacity, including agent-neutral accounts such as the one defined above, fail to capture is the fact that some instances of opacity have nothing to do with an agent's epistemic limitations. In the context of computational methods, such as machine learning techniques and/or simulation models, what this means is that these accounts fail to capture the fact that sometimes these processes are "not opaque because of sloppy modeling [practices]" or because these processes are "poorly understood", rather they can be opaque simply "*because* they are complex." [10] (Saam, 2017 p.80) This quote from Nicole Saam has many important elements. The first one is that it distinguishes between sources of opacity: bad practices, inadequate epistemic skills, abilities, etc. The second one is the use of the term 'because' rather than any other term, which captures the causal determination of the source of opacity. But lastly, it also includes a much more important distinction for our discussion, namely the fact that it is "they"—the processes, that are complex. It points to a cause and source of opacity that is not derived from an agential feature and captures the fact that some instances of opacity arise in virtue of properties of the processes per se—things outside the agent's epistemic properties. Consequentially, we can see now that while the distinction between general opacity and essential opacity will capture many instances of opacity, it will fail to capture instances of opacity that a) are not *enhanced or reduced by changes in* the limitations of an agent or b) *do not emerge from* the limitations or abilities of agents. It is these instances that I call agent-neutral and agent independent respectively. I will expand in detail about them below.

Let us now take a look at agent-independent opacity, of which I have already given a sense of given the discussion in the paragraphs immediately above this one. It would not be very useful to simply provide yet another account of general epistemic opacity, only this time without an agential reference. This is because, once again, the account would capture too much. Afterall most things in the world are opaque to most agents for reasons that are beyond the agent itself. These instances of opacity arise in virtue of and are explained by factors other than anything related to the agent. Yet, many of them are in fact responsive to agential resources and/or efforts. That is, their opacity is enhanced and or minimized relative to things the agent can have or can do. In this sense, they are not truly agent-independent.

So, an account of agent-independent opacity must include both the fact that the opacity does not arise in virtue of anything related to an agent *and* the fact that it is not responsive to agential resources and/or efforts.

So, I shall say that a process P is opaque in an agent-independent manner

iff it is impossible, in virtue of a feature F (and/or set of features) *of the process* P, which is irresponsive to (not enhanced or minimized by) agential resources, to know the epistemically relevant elements of the process.

[10] Emphasis and italics are mine.

Roughly, we can consider any stochastic process which is also vastly overdetermined to fit this description in that it is precisely the combination of its stochasticity (the randomness of paths chosen) and the vast overdetermination (the fact that many—too many— different paths lead to the same outcome) that make inquiry into the actual paths taken (the relevant epistemic elements of the process) inaccessible. Given that the process can and could have taken any of the possible paths or combination of paths and given that there are too many paths knowing exactly what relevant steps (paths) were taken for the process to produce its results, say by reverse engineering or by running another iteration, is not feasible.

A similar problem arises in mathematical error related to the coupling of vast computer simulation models of complex dynamic systems (Kaminski, 2017; Lenhard and Hasse, 2012; Lenhard, 2017). Mathematical error is inherent to certain kinds of computational processes and computational components. Due to discretization techniques and resource constraints like array size, to name a couple of examples, computational approximates of continuous equations rely on rounding results.[11] When each of the multiple components and processes in a computer simulation round up or down mathematical results the discrepancies between what should be and what the computer produces are non-trivial. The number of individual components involved in a simulation practice that will be prone to the kind of rounding up error described above, as well as their 'bundled' nature—the fact that they are often grouped together as elements of subsequent operations (Lenhard and Hasse, 2012)—make tracking and correcting for such errors simply unfeasible.[12] Kaminski et al. (2018) refer to instances such as this as *mathematical opacity*.[13]

Similarly, machine learning methods involve processes that are often unknown and untraceable as they do not require and often cannot consider (Burrell, 2016) human capabilities and needs—such as explanations and/or causal orderings of a tractable size—in order to work (Alvarado and Humphreys, 2017). There is a sense in which if machine learning processes were designed to be accessible to limited representational abilities of humans they just would not work (Burrell, 2016 p.5; Mitchell, 2019). The salient features to an algorithm are not the salient features that humans recognize. This can be clearly seen when we consider that modern vehicles with screens and automated features require two different kinds of 'camera': one kind for human perception in

[11] An array is a data structure in which values can be stored and used by computational processes following an index or a mathematical formula. A simple array can be represented by boxes, each of which can contain 0's and 1's or single digits from 0 to 9. Although their capacity grows, arrays in computers will store a finite number of digits and will lead to rounded results when the results of a computation include a number of digits that exceed the number of arrays available for storage.

[12] Presumably this limitation applies to cognitive agents such as ourselves, but also to any software-based automated attempt.

[13] There is a sense in which mathematical opacity can be avoided, of course, but not minimized. That is, if a process is opaque in the manner described above one can refuse to engage with it, use it, or implement it. But this is not the same as minimizing its opacity. Once we found out it is opaque, the process remains opaque. One does not minimize the opacity of an object merely avoiding it. This something that one must consider regarding essential opacity as well. As we saw above, at the heart of the definition of essential opacity is an impossibility clause. That is, it is impossible for an agent to access the epistemically relevant elements of a given process. There is a sense in which it is hard to imagine, as some do, that one can minimize the essential opacity of a process. Either it is essential opacity, or it isn't. There is something strange about stating that something can be made more or less impossible.

which colors and shapes matter, and another kind that identifies depth and rate of change for the machine learning system to process.

But there is something else to machine learning algorithms of this sort that make them opaque in an agent-independent manner, as described above. The scale of data, the number and properties of salient features that will be picked up by the process, the alterations to the main code done by the learning aspect of the process all contribute to its complexity and opacity in relation to us (Alvarado and Humphreys, 2016; Burrell 2016). However, widespread methods in machine learning such as deep neural networks work in such a way that their hidden layers process information through activation thresholds that are sometimes modified through techniques such as back-propagation. These techniques, in which the algorithm takes the difference between an incorrect and a correct result and uses it to modify the threshold weights of the neurons involved in the computation, are often automated. The vastness of iterations implemented for back-propagation error correction to work as intended in machine learning technology, as well as the fact that some of the activation threshold in the deep layers of a neural network are often initially randomly selected (particularly in unsupervised analytics), both contribute to making these processes opaque in an agent-independent way. Data processing methods of this kind are also able to find hidden relations between *hidden* features of a data set, or *concealed statistical structures* (Alvarado and Humphreys, 2016). Alvarado and Humphreys (2016) call the ability of machine learning algorithms to process concealed statistical structures *representational opacity.* Just like in the stochastic methods briefly described above, the establishment of the weights, the ways by which these weights are modified and the manner by which these weights in the different layers then produce a result are simply not amenable to inspection, correction, or even assessment. They are not epistemically available, and they are the epistemically relevant elements of the process. That is, it is in virtue of these processes that the process itself produces a result. This is an important departure from the opacity related to testing lines of code: the algorithm may be intractable as described above, but the weights—and the processes by which the weights are established— used by the algorithm to optimize its output are opaque in a more significant manner: essentially and independently.

Taking Stock

In the first part of this paper I focused on providing an account of what I call general epistemic opacity. I showed that while in its original definition it proved to be too broad, important developments had led researchers to formulate more particular accounts. These development made important contributions in the attribution of opacity to social factors such as technical skills, division of labor, and institutional secrecy. The combination of which pointed to more severe and seemingly inescapable cases of epistemic opacity such as essential epistemic opacity.

The discussion in the second part of this paper aimed to show that there are significant differences even between conventional accounts of epistemic opacity. In particular, I showed the fundamental differences between general instances of opacity and essential epistemic opacity. I argued that while a general sense of epistemic opacity is too broad to provide an informative sense of the sources and nature of epistemic opacity, the essential sense of epistemic opacity cannot be stripped from fundamental aspects of its definition, such as its appeal to an agent's nature and the impossibilities related to it, without collapsing it into just another instance of general epistemic opacity, which it is not. This shows, that some recent attempts seeking to explain away clauses regarding the nature of an agent and/or the impossibilities related to it fail in a fundamental way to properly account for the significance of Humphreys stronger account of epistemic opacity. This also led to the implication that while a spectrum could be formulated in which both instances of opacity could be compared to one another, their differences were definitely not solely a matter of degree but rather a matter of kind. One kind of opacity was relative and tied to the agent's epistemic status at a certain time, while the other one arose from an impossibility related to an agent's essential nature. Both proved to be significantly different kinds of challenges. In that section of the paper I also began to introduce the notion that some instances of opacity may not be at all related to agential limitations. This provided a framework with which to group the two broad categories of epistemic opacity so far explored as being agent-based, which in turn evidenced their limitations as incomplete accounts of the full range of instances of epistemic opacity. As a preliminary step towards a conceptualization of what I call agent-independent instances of epistemic opacity, I also provided a formulation and an example of what I called agent-neutral instances of epistemic opacity. These were instances of opacity that while still related to features of an agent they were so related in virtue of a property such that made them non-responsive (not enhanced or minimized in a non-trivial manner) to agential resources and/or efforts.

The main aim of this paper was to provide an account of what I call agent-independent instances of epistemic opacity and to show that conventional accounts of epistemic opacity, insofar as they were agent-based accounts of epistemic opacity, failed to captured them and hence were limited in relation to them. While the agent-neutral account of epistemic opacity, exemplified by challenges related to software error assessment, accounted for the way in which some instances of opacity were *not responsive* to agential modifications in terms of their epistemic resources, there was still work to do to show instances of opacity that *did not arise in virtue* of agential resources or lack thereof. To this aim, I provided a definition of agent-independent epistemic opacity which encapsulated both the unresponsiveness of agent-neutral opacity and this extra step of detachment signaling the origins of the opacity in something other than agential properties. These two elements of the definition were further exemplified by instances of epistemic opacity that arose in the context of mathematical error related to the rounding of results in computer simulations (mathematical opacity) as well as by the inaccessible processes in machine learning (representative opacity).

Conclusion

As I have shown, relevant cases of error assessment in software are agent-neutral instances of epistemic opacity. They do not respond to changes in agential resources. Further, because of their fundamental aspect, they still represent a significant instance of opacity in computation that is not properly captured by conventional accounts such as the ones discussed above. This is particularly because of the unwarranted focus on agents behind conventional accounts. Agent-based accounts of opacity in software fail to account for these cases and so fail to provide a complete picture of epistemic opacity. Furthermore, I have shown that there are certain instances of epistemic opacity, particularly related to computer simulations and machine learning methods, that are actually agent-independent in that the cause/origin of the opacity is non-agential. This is an important conceptual contribution to the understanding of epistemic opacity. As we move forward with the design and development of instruments such as computer simulations and other computational technology, the distinctions articulated above force us to look into the artifact's features and/or properties as sources and explanations of epistemic opacity. This is in sharp contrast to the historical and conceptual focus of conventional accounts of epistemic opacity on anthropocentric limitations. This is particularly important when we are in the business of creating instruments, such as computer simulations and other analytic tools, that have an epistemic aim that goes beyond simple predictions and results and towards genuine understanding of the phenomenon we are inquiring about (Symons and Alvarado, 2019). If we are building an instrument with the hopes of capturing an understanding of both the instrument and the phenomenon it is designed to investigate, it behooves us to distinguish between the limitations that are the product of our nature as epistemic agents and those that are the product of the artifacts themselves. This is not just for our own sake as knowers, but for science as a general enterprise of understanding and explanation: science by us that would be science to anybody.

Bibliography

Alvarado (2020) Opacity, Big Data, Artificial Intelligence and Machine Learning in Democratic Processes. In Kevin Macnish & J. Galliott (Eds.), *Big Data and Democracy* (pp.167-181) Edinburgh University Press.

Alvarado, Rafael, and Paul Humphreys. "Big Data, Thick Mediation, and Representational Opacity." New Literary History 48, no. 4 (2017): 729-749.

Angius, Nicola. "Computational Idealizations in Software Intensive Science: a Comment on Symons' and Horner's paper." Philosophy & Technology 27, no. 3 (2014): 479-484.

Arbesman, Samuel. Overcomplicated: technology at the Limits of Comprehension. Penguin, 2017.

Baird, Davis. Thing knowledge: A philosophy of scientific instruments. University of California Press, 2004.

Barberousse, Anouk, Sara Franceschelli, and Cyrille Imbert. "Computer simulations as experiments." Synthese 169.3 (2009): 557-574.

Barberousse, Anouk, and Marion Vorms. "About the warrants of computer-based empirical knowledge." Synthese 191.15 (2014): 3595-3620.

Beisbart, Claus. "How can computer simulations produce new knowledge?." European journal for philosophy of science 2, no. 3 (2012): 395-434.

Beisbart, Claus. "Advancing Knowledge Through Computer Simulations? A Socratic Exercise." In The Science and Art of Simulation I, pp. 153-174. Springer, Cham, 2017.

Bennett, Charles H. "Logical depth and physical complexity." The Universal Turing Machine A Half-Century Survey (1995): 207-235.

Durán, Juan M. "Varieties of Simulations: From the Analogue to the Digital." In The Science and Art of Simulation I, pp. 175-192. Springer, Cham, 2017.

Burge, Tyler. "Content preservation." The Philosophical Review 102.4 (1993): 457-488.

Corbató, Fernando J. "On building systems that will fail." In ACM Turing award lectures, p. 1990. ACM, 2007.

Dalmedico, Amy Dahan. "History and epistemology of models: Meteorology (1946–1963) as a case study." Archive for history of exact sciences 55, no. 5 (2001): 395-422.

Durán, J. M., & Formanek, N. (2018). Grounds for trust: Essential epistemic opacity and computational reliabilism. *Minds and Machines*, *28*(4), 645-666.

Floridi, Luciano, Nir Fresco, and Giuseppe Primiero. "On malfunctioning software." Synthese 192.4 (2015): 1199-1220.

Frigg, Roman, and Julian Reiss. "The philosophy of simulation: hot new issues or same old stew?." Synthese 169, no. 3 (2009): 593-613.

Hartmann, Stephan. "The world as a process." In Modelling and simulation in the social sciences from the philosophy of science point of view, pp. 77-100. Springer, Dordrecht, 1996.

Hehner, Eric CR. "Observations on the Halting Problem." arXiv preprint arXiv:1606.08703 (2016).

Horner, Jack, and John Symons. "Reply to Angius and Primiero on software intensive science." Philosophy & Technology 27, no. 3 (2014): 491-494.

Humphreys, Paul. Extending ourselves: Computational science, empiricism, and scientific method. Oxford University Press, 2004.

Humphreys, Paul. "The philosophical novelty of computer simulation methods." Synthese 169, no. 3 (2009): 615-626.

Irmak, Nurbay. "Software is an abstract artifact." Grazer Philosophische Studien 86.1 (2013): 55-72.

Kaminski, Andreas. "Der Erfolg der Modellierung und das Ende der Modelle. Epistemische Opazität in der Computersimulation." (2017).

Kaminski, Andreas, Michael M. Resch, and Uwe Küster. Mathematische Opazität. Über Rechtfertigung und Reproduzierbarkeit in der Computersimulation: Jahrbuch Technikphilosophie (2018)

Kroes, Peter. "Design methodology and the nature of technical artefacts." Design studies 23, no. 3 (2002): 287-302.

Kroes, Peter. "Coherence of structural and functional descriptions of technical artefacts." Studies In History and Philosophy of Science Part A 37, no. 1 (2006): 137-151.

Kroes, Peter. Technical artefacts: Creations of mind and matter: A philosophy of engineering design. Vol. 6. Springer Science & Business Media, 2012.

Kouw, Matthijs. "Standing on the Shoulders of Giants—And Then Looking the Other Way? Epistemic Opacity, Immersion, and Modeling in Hydraulic Engineering." Perspectives on Science 24, no. 2 (2016): 206-227.

Lenhard, Johannes. "Epistemologie der Iteration. Gedankenexperimente und Simulationsexperimente." Deutsche Zeitschrift für Philosophie Zweimonatsschrift der internationalen philosophischen Forschung 59, no. 1 (2011): 131-145.

Lenhard, Johannes, and Hans Hasse. "Fluch und Segen: Die Rolle anpassbarer Parameter in Simulationsmodellen." In Technisches Nichtwissen, pp. 69-84. Nomos Verlagsgesellschaft mbH & Co. KG, 2016.

Ludwig: Reproducibility in Science, Computer Science & Climate Science, Vortrag in Leogang (08.03.2017).

Mitchell, Melanie. Complexity: A guided tour. Oxford University Press, 2009.

Morrison, Margaret. "Models, measurement and computer simulation: the changing face of experimentation." Philosophical Studies 143, no. 1 (2009): 33-57.

Morrison, Margaret. Reconstructing reality: Models, mathematics, and simulations. Oxford Studies in Philosophy o, 2015.

Newman, Julian. "Epistemic opacity, confirmation holism and technical debt: Computer simulation in the light of empirical software engineering." In International Conference on History and Philosophy of Computing, pp. 256-272. Springer, Cham, 2015.

Nissenbaum, Helen. "Computing and accountability." Communications of the ACM 37, no. 1 (1994): 72-81.

Nissenbaum, Helen. "Accountability in a computerized society." Science and engineering ethics 2, no. 1 (1996): 25-42.

Nola, Robert, and Howard Sankey. Theories of scientific method: an introduction. Routledge, 2014.

Norton, Stephen, and Frederick Suppe. "Why atmospheric modeling is good science." Changing the atmosphere: Expert knowledge and environmental governance (2001): 67-105.

Nott, George, “Google’s research chief questions value of ‘explainable AI’,” Computerworld (2017).

O'Neil, Cathy. Weapons of math destruction: How big data increases inequality and threatens democracy. Broadway Books, 2017.

Oreskes, N., Shrader-Frechette, K. and Belitz, K., 1994. Verification, validation, and confirmation of numerical models in the earth sciences. Science, 263(5147), pp.641-646.

Parker, Wendy S. "Computer simulation through an error-statistical lens." Synthese 163, no. 3 (2008): 371-384.

Parnas, David L., A. John Van Schouwen, and Shu Po Kwan. "Evaluation of safety-critical software." Communications of the ACM 33, no. 6 (1990): 636-648.

Pincock, Christopher. Mathematics and scientific representation. Oxford University Press, 2011.

Pitt, Joseph C. "It’s not about technology." Knowledge, Technology & Policy 23, no. 3-4 (2010): 445-454.

Primiero, Giuseppe. "On the ontology of the computing process and the epistemology of the computed." Philosophy & Technology 27, no. 3 (2014): 485-489.

Pryor, James. "When warrant transmits." Mind, meaning, and knowledge: Themes from the philosophy of Crispin Wright (2012): 269-303.

Ruphy, Stéphanie. "Computer Simulations: A New Mode of Scientific Inquiry?." In The Role of Technology in Science: Philosophical Perspectives, pp. 131-148. Springer Netherlands, 2015.

Saam, Nicole J. "Understanding Social Science Simulations: Distinguishing Two Categories of Simulations." In The Science and Art of Simulation I, pp. 67-84. Springer, Cham, 2017.

Salmon, Wesley C. Scientific explanation and the causal structure of the world. Princeton University Press, 1984.

Schickore, Jutta. "Scientific Instrument." In Encyclopedia of Systems Biology, pp. 1897-1901. Springer New York, 2013.

Simoulin, Vincent. "Une communauté instrumentale divisée... et réunie par son instrument." *Revue d'anthropologie des connaissances* 1.2 (2007): 221-241.

Simoulin, Vincent. "An instrument can hide many others: Or how multiple instruments grow into a polymorphic instrumentation." *Social Science Information* 56.3 (2017): 416-433.

Symons, John, and Fabio Boschetti. "How computational models predict the behavior of complex systems." Foundations of Science 18, no. 4 (2013): 809-821.

Symons, John and Jack Horner. “Epistemic Limitations of Scientific Software.” Unpublished Manuscript.

Symons, John, and Jack Horner. "Software intensive science." Philosophy & Technology 27, no. 3 (2014): 461-477.

Symons, John, and Ramón Alvarado. "Can we trust Big Data? Applying philosophy of science to software." Big Data & Society 3, no. 2 (2016): 2053951716664747.

Symons John F. and Jack Horner " Software Error as a Limit to Inquiry for Finite Agents: Challenges for the Post-human Scientist In Powers, T. (ed.) Philosophy and Computing: Essays in Epistemology, Philosophy of Mind, Logic, and Ethics. Springer pp. 2017

Taddeo, Mariarosaria, and Luciano Floridi. "The case for e-trust." Ethics and Information Technology 13, no. 1 (2011): 1-3.

Turing, Alan Mathison. "On computable numbers, with an application to the Entscheidungsproblem." Proceedings of the London mathematical society 2, no. 1 (1937): 230-265.

Tymoczko, Thomas. "The four-color problem and its philosophical significance." The Journal of Philosophy 76, no. 2 (1979): 57-83.

Van Helden, A. (1994). Telescopes and authority from Galileo to Cassini. *Osiris*, *9*, 8-29.

Weisberg, Michael. Simulation and similarity: Using models to understand the world. Oxford University Press, 2012.

Winsberg, Eric. Science in the age of computer simulation. University of Chicago Press, 2010.

Winsberg, Eric, "Computer Simulations in Science", The Stanford Encyclopedia of Philosophy (Summer 2015 Edition), Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/sum2015/entries/simulations-science/>.

Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." In European conference on computer vision, pp. 818-833. Springer, Cham, 2014.