DOI: 10.1017/psa.2025.10160
This is a manuscript accepted for publication in *Philosophy of Science*.
This version may be subject to change during the production process.

RESEARCH ARTICLE

Apriori Knowledge in an Era of Computational Opacity: The Role of AI in Mathematical Discovery

Eamon Duede[1,2], Kevin Davey[3]

[1]Purdue University, [2]Argonne National Laboratory, [3]University of Chicago
Corresponding author: Eamon Duede; eduede@purdue.edu, Kevin Davey; kjdavey@uchicago.edu

Abstract
Can we acquire apriori mathematical knowledge from the outputs of computer programs? Although we claim Appel and Haken acquired apriori knowledge of the Four Color Theorem from their computer program insofar as it merely automated human forms of mathematical reasoning, the opacity of modern LLMs and DNNs creates obstacles in obtaining apriori mathematical knowledge in analogous ways. If however a proof-checker automating human forms of proof-checking is attached to such machines, we can indeed obtain apriori mathematical knowledge from them, even though the original machines are entirely opaque to us and the outputted proofs are not human-surveyable.

1. Introduction

Our main question is what role computers can play in expanding our purely rational capacities and purely rational knowledge. When we learn a fact from a computer, should we think of ourselves as merely having done an experiment of sorts and thus having acquired only *empirical* knowledge? Or can learning something from a computer sometimes expand our non-empirical (i.e., our purely rational or *apriori*) knowledge?

The obvious case to focus on is mathematics. Suppose a computer tells us that some mathematical claim is true. In the right circumstances, might we then know that mathematical fact on purely rational (i.e., apriori) grounds? Presumably the answer to this question will depend on what sort of computer we are talking about. We thus phrase our question as follows:

> **Main Question**: Are there situations in which we can acquire apriori knowledge of a mathematical fact X purely on the basis of a computer outputting the claim that X is true? If so, what sorts of situations are these?

This question concerns the acquisition of apriori knowledge *purely* on the basis of a computer outputting the claim that something is true. Thus, suppose that a computer outputs the claim that some mathematical fact X is true, *and* a correct proof of X. If in some circumstances merely witnessing the computer output the claim X yields apriori

knowledge of X, then in such circumstances apriori knowledge of X is acquired purely on the basis of the computer outputting the claim X. But if it is only once we check the proof of X ourselves that we acquire apriori knowledge of X, then it is *not* true that in those circumstances we come to know X apriori purely on the basis of the computer outputting the claim X.

Early discussions of our Main Question were motivated by Appel and Haken's 1977 computer proof of the Four Color Theorem (henceforth 4CT) (Appel and Haken, 1989), which is so long that it cannot be human-checked. Because of this, some thought (Tymoczko, 1979) that the idea that mathematical knowledge is essentially apriori had to be rejected and room created for merely empirical or experimental mathematical knowledge.

However, following Burge 1998 we argue in Section 2 that when the running of a computer program can be understood as a mechanized exercise of ordinary human mathematical capacities, the output of a program can indeed give us apriori mathematical knowledge. In this way, Appel and Haken *did* acquire apriori knowledge of the truth of 4CT from the output of their computer program. Thus our Main Question can be answered affirmatively in the case of the Appel and Haken in 1977.

The problem, however, is that the argument of Section 2 does not apply to the output of machines like deep neural networks (henceforth DNNs) and large language models (henceforth LLMs), whose inner workings are (in a sense) opaque to us, but are nevertheless increasingly important to mathematicians. In Section 3, we argue that outside special cases we cannot *directly* acquire apriori mathematical knowledge from the reports of DNNs or LLMs. This result seems to impose a strong limitation on our ability to acquire apriori mathematical knowledge from AI.

However, in Section 4 we argue that mathematicians can overcome this limitation by applying a transparent proof-checker to an appropriately structured output of a DNN or LLM. So long as this proof-checker may be understood as a mechanized exercise of human proof-checking capacities, we claim that we can acquire genuine mathematical knowledge from the output of the proof-checker, even though this knowledge may not be obtained directly from the DNN or LLM itself.

We thus arrive at the surprising result that it is possible to acquire genuine apriori knowledge of a mathematical fact X purely on the basis of the output of a computer, where a proof of X has been generated by a process that is entirely opaque to us, *and* is so complex that the proof is not human-checkable. This suggests that AI can indeed play a role in generating substantive mathematical knowledge and that there is a large set of cases in which we can answer our Main Question affirmatively and acquire apriori knowledge purely on the basis of the output of a computer.

2. Knowledge of the Four Color Theorem

Serious discussion of our Main Question began in 1977 when Appel and Haken used a computer to verify 4CT (Appel and Haken, 1989). Appel and Haken argued that to prove 4CT it sufficed to verify the 4-colorability of a particular set of 1,834 finite graphs. For complex graphs, verifying 4-colorability is extremely time-consuming. Appel and Haken thus used a supercomputer to verify that all 1,834 graphs were 4-colorable, and this was regarded as establishing 4CT. The sheer length of this calculation meant however that human mathematicians could not survey it step-by-step. Indeed, even

today no-one has produced a proof of the 4CT that can be checked without computer assistance.

Philosophers immediately began reflecting on what this meant for mathematics. Tymoczko 1979 argued that mathematics had now become an empirical discipline in which proofs could be obtained by performing *experiments* such as the running of computer programs.[1] More specifically, Tymoczko thought that our knowledge of the 4CT rested on an argument involving the premise

Rel: Carefully written computer programs reliably output true claims.

Tymoczko saw **Rel** as an *emprical* claim stating the reliability of a piece of scientific equipment. Knowledge obtained using **Rel** was thus empirical. Tymoczko concluded that our knowledge of 4CT, while genuine knowledge, was not apriori – i.e., not justified *purely* on rational grounds – but rather merely empirical. Others concurred (Detlefsen and Luker, 1980).

We however do not find this view compelling. Instead, we are persuaded by another way of looking at things due to Burge 1998. Rejecting the views just described, Burge argued that the output of Appel and Haken's program gives a genuinely apriori[2] warrant for believing 4CT.[3]

2.1. Memory as a Rational Resource

To motivate this view, consider that in proving or surveying a theorem we typically must use our memories. When proving a theorem, there come points where we may wonder whether we have already proven some lemma. Perhaps we pause, recall that we have, and then continue reasoning. In that pause when we ask 'Have we already proven this lemma?' and decide upon consulting our memory that we have, are we doing an experiment with our brains? Is the resulting knowledge thus merely empirical? And like ordinary pieces of scientific equipment, can we only rely on our memory if we have empirical knowledge of its reliability?

Burge thinks that relying on our memory is *not* doing an experiment that yields at most empirical knowledge. His view is rather that we have *defeasible, apriori* grounds for believing what we seemingly remember. So when we have a seeming memory of proving a lemma, we are entitled on purely rational grounds to believe that we have proved the lemma. It is not the case that we must first do memory-tests to establish the reliability of our memory before we have grounds to believe what we seemingly remember. Rather, Burge's view is that we have purely rational grounds for believing the lemma, *because we have a memory of proving it*. These grounds are of course defeasible.

[1] Detlefsen and Luker 1980 further argued that mathematics had always been an empirical discipline and that there was therefore nothing philosophically novel about Appel and Haken's accomplishment.

[2] Burge uses the term 'apriori' to indicate that the justification does not rely on empirical evidence. While there is debate about how to define the apriori (see (Williamson, 2013)), we simply assume ordinary mathematical arguments are apriori, without making any contentious claims about what this means.

[3] Appel and Haken's program strictly proves only the 4-colorability of 1,834 graphs; an additional human-generated argument is required to yield 4CT. For simplicity, we treat the program as offering an apriori warrant for 4CT, though technically it only warrants belief in the 4-colorability of these 1,834 graphs.

We might later realize that we were misremembering. Purely rational grounds are not infallible on this picture.

To be sure, searching our memories for an episode of proving a lemma is something like an empirical investigation. It is only an empirical fact that I proved the lemma yesterday, and it is only an empirical fact that I have a memory of this happening. Nevertheless, upon finding the memory, I have purely rational (i.e., apriori) grounds for believing the lemma. Critically, note that the lemma is not inferred from the existence of the memory – rather, the lemma is inferred *from the reasoning that has been remembered.*

More generally, Burge claims that even when only exercising our rational capacities, there are various resources on which we can rely. He calls these *rational resources*. When these rational resources offer us claims, we are (defeasibly) apriori entitled to trust them. So, for example, my memory is a rational resource that I can rely on in my reasoning, as is my visual system when I inspect a diagram in a proof. A thermometer however is not a rational resource, because when I trust a thermometer, I am doing something going beyond mere reasoning. A thermometer is rather an *empirical* resource. Burge's general claim is that '*resources for rationality are, other things equal, to be believed*' (Burge, 1998, p.5). This extends to the use of these resources outside cases of pure reasoning, though we shall not dwell on this here.

2.2. Computers as Rational Resources

We now return to Appel and Haken's computer program. This program is designed to go through all 1,834 basic maps and verify their 4-colorability in exactly the way Appel and Haken might. It organizes these maps systematically into a list and does exactly what they would to check each case, though more quickly and indefatigably. Appel and Haken's computer is thus simply a mechanized application of their ordinary rational capacities, performing their reasoning for them. Indeed, they understand *exactly* what the computer does, so that at any moment they could (in principle) say something like 'the computer is now considering map #734, and is at such-and-such a stage of checking for a 4-coloring.' We capture this aspect of Appel and Haken's program by saying that it is *mathematically transparent* to them.

Because Appel and Haken's program simply performs their reasoning for them in this way, we view Appel and Haken's computer as a rational resource. Because rational resources are (other things equal) to be believed, we agree with Burge that '*we have apriori prima-facie entitlement to accept [the print-outs of the program] as true*' (Burge, 1998, p.13). Thus, Appel and Haken can be understood to have defeasible, apriori grounds for believing 4CT.

Crucially, this warrant does not depend on an empirical fact like **Rel** about the reliability of computers. Appel and Haken infer 4CT *from the reasoning that the computer has done for them.* This reasoning involves only purely mathematical considerations and not **Rel**. Nevertheless, the warrant for believing 4CT is defeasible. For example, Appel and Haken could come to learn that the computer was malfunctioning, in which case they would no longer be justified in believing 4CT. This however does *not* mean that justification for believing 4CT first requires positive empirical grounds for thinking that the computer is *not* malfunctioning. Instead, they are entitled to believe the results of mathematically transparent processes so long as they *lack* reason for thinking the

relevant resource unreliable. So it is enough that they had no reason to think that their computer was malfunctioning.

It is true that in writing the program, Appel and Haken had to perform all sorts of tests to establish that the program was behaving as expected. But this does not mean that the warrant for believing 4CT is merely empirical. Empirical tests are necessary to establish that the computer is a device that is capable of performing our reasoning for us through mathematically transparent processes. Nevertheless, once we are confident of this and use the computer as Appel and Haken did, the ultimate ground for accepting 4CT is then simply the existence of the mathematically transparent process demonstrating it. This warrant is apriori insofar as it is just a mobilization of human mathematical capacities, albeit in a way that relies on rational resources.

Thus, in the same way that ordinary mathematicians infer theorems from the existence of purely mathematical arguments *not* involving claims about their memories (even though they rely on their memories in convincing themselves of the existence of such arguments), so too Appel and Haken inferred 4CT from the existence of a purely mathematical argument that did *not* make any claim like **Rel** about computers (even though they relied on computers in convincing themselves of the existence of such an argument).

3. Transparency and AI Assisted Proof

So far we have been talking about 'old-fashioned' computing. Recently, mathematicians have turned to deep learning models (DLMs) for assistance with challenging mathematical problems in for instance low-dimensional topology (Davies et al., 2021), geometry (Trinh et al., 2024), and combinatorics (Romera-Paredes et al., 2023). One might expect that like Appel and Haken these mathematicians can gain apriori knowledge from computers in the right circumstances. However, the notorious opacity of DLMs creates significant differences between the use of traditional computers and contemporary AI in mathematics.

To see this, we consder the sense in which DLMs are opaque and thus not mathematically transparent to us. Creel's account of algorithmic and structural transparency in complex computational systems is helpful here (Creel, 2020).

For Creel, computational systems can be 'algorithmically' and 'structurally' transparent.[4] A computational system is algorithmically transparent to the extent that the procedures governing its behavior are known and intelligible. In the case of the procedures performed in the proof of the 4CT, the rules at the algorithmic level those describing how a mathematician might check the 4-colorability of basic graphs.

The system is structurally transparent to the extent that it is possible to see how this algorithm is realized in actual code. Thus, a program is structurally transparent just when its code is surveyable, and it is possible to understand how the code generates results in accordance with the algorithm it instantiates. In cases where a computational system is algorithmically *and* structurally transparent (as in Appel and Haken's program), the reliability of the computational system at run-time can be (defeasibly) trusted (Frigg

[4]Creel's treatment of computational transparency can be seen as a refinement of computational concepts of understanding going back to Marr 2010.

and Reiss, 2009; Duede, 2022), even if in practice one cannot transparently survey the running of the program. (Humphreys, 2009).

While the computations used in Appel and Haken's program resulted in an unsurveyable proof, the computations themselves were on Creel's account algorithmically and structurally transparent. Thus the computations were mathematically transparent in the sense discussed in the last section. However, we will argue that the use of DNNs and LLMs in mathematics are often neither structurally nor algorithmically transparent.

3.1. Opacity of Deep Learning

It is often said that DNNs lack epistemic transparency. It is important however to distinguish the *training* of a DNN from fully *trained* models. The procedure for training a DNN is algorithmically and structurally transparent. In simple cases it is algorithmically transparent that training works through the minimization of loss-functions via iterative updating of weights on the connections between parameters by backpropagation of error gradients. There are extensive repositories containing structurally transparent implementations for training a wide variety of network architectures in this way, and students taking a class in machine-learning are often required to write their own implementations of such algorithms. There is nothing opaque about how such models are trained.

However, *fully trained* DNNs are said to be epistemically opaque (Humphreys, 2004; Boge, 2022), meaning that the epistemically relevant factors governing the model's behavior are fundamentally unsurveyable. In general, it is not possible to 'fathom' (Zerilli, 2022) in any meaningful sense the algorithmic principles governing the transformation of inputs to outputs of the model. As such, DNNs are opaque at both the algorithmic and structural levels. This lack of transparency is due to extreme dimensionality and nonlinearity of the model, as well as the autonomous, error-driven, and semi-stochastic processes of weight assignment guiding the final parameterization of the model.

Of course, it is possible that with a DNN that determines 4-colorability, we might be able to say at any moment which graph is being analyzed. There is also a numerically trivial sense in which the trained model is transparent insofar as the values on the weights themselves are available to inspection (though not surveyable) (Lipton, 2018; Duede, 2023). However, unlike Appel and Haken's program, we would not generally be able to say *how* that graph is being evaluated, and thus such an approach would neither be algorithmically nor structurally transparent.

3.2. Mathematical Knowledge with DNNs

Suppose that a mathematician wants to know whether every graph in a set of graphs is 4-colorable. Approaching this problem with a DNN, the mathematician trains a model on a large set of graphs known to be 4-colorable and a large set of graphs known not to be 4-colorable. The model is then evaluated (in the usual way) on a collection of graphs not included in the training set, and let us suppose that no graph is misclassified.

At this point, the mathematician unleashes their model on Appel and Haken's 1,834 basic graphs. After several minutes the model states that all are 4-colorable. However, given that the model is not algorithmically transparent, we cannot regard this machine as having performed on our behalf the kind of reasoning we would perform in verifying

the 4-colorability of the graphs. Because the system is not mathematically transparent, we would *not* be justified in believing its output on purely rational grounds.

Because the DNN reliably classifies graphs as 4-colorable or not, we do get strong *inductively* justified belief in the 4CT. Such a result bears some resemblance to a case (Davies et al., 2021) considered by Duede 2023, where mathematicians use a DNN to guide mathematical attention to promising connections that led to the formulation and proof of a theorem linking specific algebraic and geometric properties of low-dimensional knots. Such cases exemplify the potential for AI to assist mathematicians in their search for promising conjectures while leaving the actual proof of the conjectures to humans. In such cases however, knowledge is not a direct result of the DNN, insofar as ultimate responsibility for proof lies with human mathematicians.

Consider next a hypothetical case in which a DNN trained to classify graph 4-colorability classifies all of Haken and Appel's 1,834 graphs as 4-colorable, except for one which it classifies as *not* 4-colorable. Here, the model has suggested a counterexample to 4CT. Let us suppose that whether it is a genuine counterexample is something we can check ourselves by hand, that we check it, and we find that it is indeed a counterexample. In this case we now have genuine mathematical knowledge of a mathematical fact (namely, the falsehood of 4CT). However, this knowledge too cannot be said to follow *directly* from the output of the DNN, as it required human verification.

3.3. Mathematical Knowledge with LLMs

LLMs are particularly useful for mathematics as they output reports that are potentially linguistically and mathematically intelligible. However, like DNNs, LLMs are algorithmically and structurally opaque, and so they are afflicted by the epistemic limitations discussed in the previous section.

A recent case leveraging LLMs to achieve mathematical breakthroughs in combinatorics involves the Cap Set Problem (Romera-Paredes et al., 2023). A 'cap set' is a subset of $(\mathbb{Z}/3\mathbb{Z})^n$ for which no three distinct elements sum to 0 (mod 3). For each n, the problem is to determine the size of the largest cap set. It is known that this number must be less than $\leq 3^n$ (Grochow, 2019), but its exact value is only known for $n \leq 6$. Because the complexity of the solution space explodes for larger values of n, brute-force computational approaches are infeasible.

In (Romera-Paredes et al., 2023), researchers leverage an LLM to construct a cap set of size 512 for $n = 8$, a result significantly greater than the previously known largest value of 496. Their approach begins by specifying an evaluation function that scores a candidate solution, where a solution is itself a `Python` program for generating a potential capset. The LLM then outputs a candidate `Python` program that is executed and scored by the evaluation function. If the program executes sufficiently quickly and without obvious error, it is sent to a program database. The system then samples the database and passes prior output programs to the LLM as inputs to repeat the generative process. This iterative approach generatively 'evolves' candidate programs. Eventually, this process identified a cap set of size 512, which human mathematicians verified to be correct.

Unlike 4CT, the solution-generating computational procedure here is not mathematically transparent. However, a human mathematician can easily survey and check its output. So in this case we get apriori knowledge that for $n = 8$ there is a cap set of size

512. Nevertheless, as this involves a human mathematician verifying this fact, we cannot say that genuine mathematical knowledge has been obtained directly from the output of the computer.

3.4. Main Claim

With such examples in mind, we offer the following answer to our Main Question posed in Section 1.

> **Main Claim 1**: If we want to acquire apriori mathematical knowledge directly from the output of a computer, then what the computer is doing must be mathematically transparent to us (as in the case of 4CT). If what a computer is doing is *not* mathematically transparent to us (as in the case of typical DNNs or LLMs) then we cannot directly acquire apriori mathematical knowledge from the output of a computer, even though we may be able to gain a type of inductively justified belief from it. However, even if we do not directly acquire apriori mathematical knowledge from the output of a computer, if the computer outputs a human-checkable proof or counterexample, then upon checking it appropriately we do gain apriori mathematical knowledge (though not directly from the computer, as human checking was required.)

4. Transparent Proof Checking

The considerations of the previous section seem to entail that, while extraordinarily useful, DNNs and LMMs can ultimately only be of limited use in acquiring apriori mathematical knowledge. At best, their reports offer us inductively justified beliefs, and it is only when they output results that are human-checkable that we can acquire apriori mathematical knowledge from them.

In certain cases, however, these limits may be surpassed. We focus on the case in which a machine (perhaps an LLM) outputs not only some mathematical claim X but also something it claims to be a proof of X, and that this proof is stored somewhere on a hard drive. If what has been stored on the drive can be human-checked, then we can check it, and if it is indeed a correct proof, we thereby gain apriori knowledge of X.

Suppose however that the proof of X stored on the hard drive is so long that it cannot be human-checked. It might then appear that apriori knowledge of X is beyond our reach.

But this is not so. Let us suppose that the stored proof of X, while enormous, is systematically organized as a tree of propositions of the sort one might encounter in a mathematical logic class. We can imagine constraining the output of the machine generating the proof to demand that its outputs be formulated in this way (as in (Romera-Paredes et al., 2023) where the model outputs all results in `Python` or, as is common (Avigad, 2024), in a formal language instantiated in `Lean` (De Moura et al., 2015)). We can permit abbreviations, additional rules, and verbose articulations of steps in the proof so that this proof has roughly the form of a human-generated proof,[5] even though it was

[5] Of course, the proofs of practicing mathematicians are not like this, often involving large leaps (Kitcher, 1998).

not human-generated and is so large that it is not human-surveyable. These assumptions can easily be made to hold by adding some overhead to the original program.

Although we cannot check this proof ourselves, this does not stop us from writing a proof-checking program that can. The proof-checking program goes through the proof, verifying that it starts with genuine axioms and that each step is a legitimate application of some standard inferential rule. We can imagine a version of this proof-checker that is completely mathematically transparent, and checks the proof in exactly the way a human would. When such a program runs, at any point we can (in principle) correctly say something like 'the computer is now checking inference 15435 and is verifying that it is a correct application of *modus-ponens*.'

Let us assume that we run this proof-checker, and it reports no errors. Just as Appel and Haken acquire apriori knowledge of the 4CT from the output of their mathematically transparent program, so too we acquire apriori knowledge that there is a correct proof of X from the output of our mathematically transparent proof-checker. Because there is a correct proof of X, X is true, and thus we acquire apriori knowledge of X. This is true even though no human has (or ever could have) any sort of rational grasp on the process that led to the generation of the proof, and no human is capable of checking the stored proof.

The important point is that in this case we have apriori knowledge of X not based on the output of the LLM, whose workings are not transparent to us, but based on the output of the proof-checker, whose workings *are* transparent to us.

Of course, if the LLM 'claims' to have proven X but cannot produce and store the actual proof of X, then we cannot use a proof-checker in the way just described. In this case, we see no way to acquire anything other than inductive grounds for believing X.

This leads us to the second of the two central claims of this paper, which may be viewed as a counterpoint to Main Claim 1.

> **Main Claim 2**: We *can* (indirectly) gain apriori knowledge from the output of a computer program that is not mathematically transparent but which stores a (not necessarily human-checkable) proof of a mathematical claim. This is accomplished by employing a mathematically transparent proof-checker to evaluate the stored proof of the claim.

5. General Conclusions

Modern LLMs and DNNs are opaque to us in ways that create obstacles to obtaining mathematical knowledge from them. However, if a proof-checker transparently automating human forms of mathematical evaluation is attached to such machines, then we can obtain apriori mathematical knowledge from them. Surprisingly, this applies even when the original machines are entirely opaque to us and the proofs they output are not human-surveyable.

A different question for further consideration is to what extent we may gain scientific (Kidd and Birhane, 2023) knowledge outside of mathematics by appending analogous transparent 'checking' mechanisms to the output of otherwise opaque algorithms. This would get us closer to overcoming the perceived problems of confabulation and realizing the ambition of fully automated scientific discovery.

Acknowledgments: The authors which to thank audiences at the Philosophy of Science Association (PSA 2024) 29th Biannual Meeting, mrc2024, IACAP2024, and the University of Chicago.

Funding statement: Funding for this work was provided by a generous grant to Duede from the Alfred P. Sloan Foundation (G-2024-22468).

Competing Interests: The author(s) declare none

References

Appel, K. I. and Haken, W. (1989) *Every planar map is four colorable*. vol. 98. American Mathematical Soc. https://doi.org/10.1112/blms/23.1.89.

Avigad, J. (2024) Mathematics and the formal turn. *Bulletin of the American Mathematical Society*. *61*(2), 225–240.

Boge, F. J. (2022) Two dimensions of opacity and the deep learning predicament. *Minds and Machines*. *32*(1), 43–75. https://doi.org/10.1007/s11023-021-09569-4.

Burge, T. (1998) Computer proof, apriori knowledge, and other minds: The sixth philosophical perspectives lecture. *Philosophical perspectives*. *12*, 1–37. https://doi.org/10.1111/0029-4624.32.s12.1.

Creel, K. A. (2020) Transparency in complex computational systems. *Philosophy of Science*. *87*(4), 568–589. https://doi.org/10.1086/709729.

Davies, A., Veličković, P., Buesing, L., Blackwell, S., Zheng, D., Tomašev, N., Tanburn, R., Battaglia, P., Blundell, C., Juhász, A. *et al.* (2021) Advancing mathematics by guiding human intuition with ai. *Nature*. *600*(7887), 70–74. https://doi.org/10.1038/s41586-021-04086-x.

De Moura, L., Kong, S., Avigad, J., Van Doorn, F. and von Raumer, J. (2015) The lean theorem prover (system description). Automated Deduction-CADE-25: 25th International Conference on Automated Deduction, Berlin, Germany, August 1-7, 2015, Proceedings 25. Springer. pp. 378–388. https://doi.org/10.1007/978-3-319-21401-6_26.

Detlefsen, M. and Luker, M. (1980) The four-color theorem and mathematical proof. *The Journal of Philosophy*. *77*(12), 803–820. https://doi.org/10.12677/pm.2019.93054.

Duede, E. (2022) Instruments, agents, and artificial intelligence: novel epistemic categories of reliability. *Synthese*. *200*(6), 491. https://doi.org/10.1007/s11229-022-03975-6.

Duede, E. (2023) Deep learning opacity in scientific discovery. *Philosophy of Science*. *90*(5), 1089–1099. https://doi.org/10.1017/psa.2023.8.

Frigg, R. and Reiss, J. (2009) The philosophy of simulation: hot new issues or same old stew? *Synthese*. *169*(3), 593–613. https://doi.org/10.1007/s11229-008-9438-z.

Grochow, J. A. (2019) New applications of the polynomial method: the cap set conjecture and beyond. *Bull. Amer. Math. Soc*. *56*(1), 29–64. https://doi.org/10.1090/bull/1648.

Humphreys, P. (2004) *Extending ourselves: Computational science, empiricism, and scientific method*. Oxford University Press. https://doi.org/10.5860/choice.43-0272.

Humphreys, P. (2009) The philosophical novelty of computer simulation methods. *Synthese*. *169*(3), 615–626. https://doi.org/10.1093/oso/9780199334872.003.0004.

Kidd, C. and Birhane, A. (2023) How ai can distort human beliefs. *Science*. *380*(6651), 1222–1223. https://doi.org/10.1126/science.adi0248.

Kitcher, P. (1998) Mathematical change and scientific change. *New directions in the philosophy of mathematics*. pp. 215–242.

Lipton, Z. C. (2018) The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*. *16*(3), 31–57. https://doi.org/10.1145/3236386.3241340.

Marr, D. (2010) *Vision: A computational investigation into the human representation and processing of visual information*. MIT press. https://doi.org/10.1016/0022-2496(83)90030-5.

Romera-Paredes, B., Barekatain, M., Novikov, A., Balog, M., Kumar, M. P., Dupont, E., Ruiz, F. J., Ellenberg, J. S., Wang, P., Fawzi, O. *et al.* (2023) Mathematical discoveries from program search with large language models. *Nature*. pp. 1–3. https://doi.org/10.1038/s41586-023-06924-6.

Trinh, T. H., Wu, Y., Le, Q. V., He, H. and Luong, T. (2024) Solving olympiad geometry without human demonstrations. *Nature*. *625*(7995), 476–482. https://doi.org/10.1038/s41586-023-06747-5.

Tymoczko, T. (1979) The four-color problem and its philosophical significance. *The journal of philosophy*. *76*(2), 57–83. https://doi.org/10.2307/2025976.

Williamson, T. (2013) How deep is the distinction between a priori and a posteriori knowledge? In *The A Priori in Philosophy*, Casullo, A. and Thurow, J. C. (eds). Oxford University Press. pp. 291–312.

Zerilli, J. (2022) Explaining machine learning decisions. *Philosophy of Science*. *89*(1), 1–19. https://doi.org/10.1017/psa.2021.13.