Annals of Mathematics and Philosophy
Online version, 31/12/2025

$M \times \Phi$

The Philosophical Prospects of Large Language Models in the Future of Mathematics

Fenner Stanley Tanswell
Ásgeir Berg

Abstract: In this article, we examine the philosophical implications Large Language Models might have on mathematical practice in the near future. Some prominent researchers argue that Large Language Models will soon have the ability to generate or check proofs, lifting a great burden of human mathematicians.

We claim, however, that the implementation of LLM technologies in mathematics is not merely a neutral tool that assists mathematicians to continue on as before, but instead entails a radical change to the practices of mathematics with important philosophical implications.

We will argue that we cannot be confident such tools will continue to work as expected, even if they become arbitrarily more reliable than they currently are, and that the kind of justification we get from LLM-generated proofs can never be properly mathematical. We will evaluate solutions to this problem involving either computer verification or human checking and argue that these cannot fix the philosophical gap to give us proper mathematical justification.

Keywords: Mathematical practice; Large Language Models; proof; rule-following paradox; reverse centaur; mathematical justification; proof assistants.

"I think in the future, instead of typing up our proofs, we would explain them to some GPT. And the GPT will try to formalize it in Lean as you go along. If everything checks out, the GPT will [essentially] say, "Here's your paper in LaTeX; here's your Lean proof. If you like, I can press this button and submit it to a journal for you." It could be a wonderful assistant in the future."

Terence Tao,
June 2024[1]

§ 1. — Introduction.

The recent arrival of Large Language Models like ChatGPT has had immediate and widespread social, cultural and technological impact. In this article, we will examine the philosophical implications these technologies might have on *mathematics in the near future*. With seemingly amazing abilities to read and write natural language and computer code, there is the tantalising possibility of a computer system capable of reading, writing, and understanding proofs just like mathematicians do. The possibilities for mathematics could be revolutionary.[2] However, these technologies are also controversial. According to some, the vision of AI mathematicians has suddenly moved closer to realisation, while others think these technologies offer a deceptive mirage of understanding hiding a systematic inability to reason intelligently. In what follows, we examine the *philosophical* prospects of LLMs in mathematics, taking an approach that tries to find a reasonable middle-ground between the hype and the criticism, and assesses the philosophical prospects of what is on offer.

Prominent mathematicians, like Terence Tao quoted in the paper's epigraph, are already predicting a future for mathematics where LLM technology plays a major role in the practices of mathematics, which could happen in several ways. Most simply, the LLMs could generate proofs directly. Alternatively, they

[1] Taken from an interview, which appeared in the *Scientific American*, with Christoph Drösser (2024).

[2] As Martin & Pease put this: "Improved knowledge of human interactions and reasoning in mathematics will suggest new ways in which artificial intelligence and computational mathematics can intersect with mathematics. [...] There is much to be done, and a substantial body of research lies ahead of us, but the outcomes could transform the nature and production of mathematics." (Martin & Pease 2013, p. 115)

could be used to bridge the gap between the everyday, informal language of human mathematics, and the formal language of computer-checkable derivations. Such a translation from the informal language of the working mathematician to a formal language is known as *autoformalisation*, with the underlying idea that a human could write a proof for an LLM to autoformalise, then to be checked by the computer.

Another option is for LLMs to be used as "AI-assistants", such as to generate examples of or counterexamples to given conjectures, to find relevant theorems in the vast and sprawling literature, to fill in details of routine parts of work, or to guide proof formalisation for an interactive theorem prover. This possibility will not feature heavily in this article and we will set aside the more interactive uses of AI for future work.

We will, however, argue that the implementation of LLM technologies in mathematics is not merely a neutral tool that assists mathematicians to continue on as before, but instead entails a radical change to the practices of mathematics with important philosophical implications. Whether such a radical change is welcome depends on many factors, but in this paper, we will focus on the question of whether we can trust mathematical LLMs to be doing what we think they are. Based on the underlying technology, we will argue that we cannot be confident that they will continue to work as expected, even if they become arbitrarily more reliable than they currently are, and that the kind of justification we get from LLM-generated proofs can never be proper mathematical justification. We will evaluate solutions to this problem involving either computer verification or human checking and argue that these cannot fix the philosophical gap and give us mathematical justification.[(3)]

In § 2 we give some philosophical background on computer-use in mathematics. § 3 provides an overview of how LLMs work, and assesses their current mathematical abilities. In § 4, we consider some basic objections to trusting LLM-authored proofs. In § 5, we give an argument reminiscent of arguments from the debate on rule-following, an epistemological analogue of the Kripke-Wittgenstein paradox: that LLMs cannot be trusted to settle on

[(3)] This topic is large, so we set aside other issues, like the interactive use of LLMs in mathematics and the ethics of using LLMs. For instance, LLMs are extremely energy-intensive during a climate crisis (see Schütze 2024), and the values of AI are driven by the tech industry rather than mathematicians themselves, with numerous potential hazards for mathematics (see Harris 2024).

the same concepts as human mathematicians, leading to a loss of proper mathematical justification. In § 6, we consider the response that such purported proofs can be checked and give the "reverse centaur" argument: that checking LLM-proofs is actually harder than checking human proofs and thus that checking cannot give mathematical justification where none was before. In § 7 we assess whether autoformalisation can provide a safety net against errors in proofs, and in § 8 we draw the paper together.

§ 2. — Proof, Justification and Computers.

The emergence of LLMs as a tool for mathematics is a new chapter in the intertwined story of mathematics and computing. Where to start the story depends on your definitions, whether it is with the abacus; calculating machines like Pascal's calculator or Leibniz's stepped reckoner; Babbage's difference and analytical engines, with Lovelace's famous "computer program"; the Turing machine, or the development of modern computing.

A modern milestone is the computer-assisted proof of the Four Colour Theorem by Appel and Hakken in 1977.[(4)] The proof involved a massive case enumeration, which was performed by a computer rather than a human being. This raised a philosophical issue about mathematical justification: does a proof that has only been checked by a computer give the same kind of mathematical justification as traditional proofs? Indeed, one can argue that this is no proof at all, as was done by Tymoczko:

> "What reason is there for saying that the 4CT is not really a theorem or that mathematicians have not really produced a proof of it? Just this: no mathematician has seen a proof of the 4CT, nor has any seen a proof that it has a proof. Moreover, it is very unlikely that any mathematician will ever see a proof of the 4CT." (Tymoczko 1979, p. 58)

He argues that the purported proof has a substantial gap that is filled by an experiment on a computer. Therefore, the theorem can only be known *a posteriori* and relies essentially on experiment and empirical evidence provided by the computer. According to

[(4)] It should be noted that this wasn't necessarily the first computer-aided proof. Detlefsen & Luker (1980) give several earlier examples.

Tymoczko, this is not the same standard of mathematical justification as traditional proofs, which are supposed to give *a priori* justification. Furthermore, he raises the fact that there is the possibility of error in the programming and glitches in the computer itself. This is not an idle possibility, as programming errors are almost inevitable in any sizeable codebase, but even hardware can lead to relevant errors, as happened in the case of the infamous Pentium FDIV bug in early Pentium processors that led to errors in floating point division of certain large numbers. This bug was discovered by the mathematician Thomas R. Nicely when errors appeared from code he had written to generate sets of primes. We add to this that even when functioning as intended, computers do not necessarily do mathematics flawlessly: at the time of writing Google's calculator will tell you that 999,999,999,999,999 minus 999,999,999,999,998 is 0 because of how processing such large numbers using floating-point arithmetic works. The point is that if there is a chance that the computer is making an error, then it cannot be the source of proper mathematical justification.[(5)]

There are replies available to these worries. Detlefsen and Luker (1980) argued that many traditional proofs also rely on the empirical consideration that humans have correctly carried out calculations within the proof, so these fare no better. One can also argue that humans are more fallible than computers at routine calculations, so that the possibility of error in the proof of the Four Colour Theorem, say, might be lower than in a complicated traditional proof. Indeed, the sociological fact is that the proof of the 4CT has largely been accepted by the mathematical community, as have several other large-scale computer proofs, like the proof that 17 is the minimum number of clues needed for a standard Sudoku to have a unique solution (McGuire et al. 2014; see also Parshina 2024). This latter proof consisted of a search through all 16-clue configurations that failed to find any with unique solutions, plus the existence of a 17-clue puzzle that has one. It would be hard to find anybody who still thinks it is worth searching for an error in this proof. Another response is that "you can choose your level of

[(5)] The mere possibility of error cannot be what rules out mathematical justification here, otherwise we would be facing worries about epistemic scepticism. The question, though, is what possibilities of error are acceptable. De Toffoli (2021) provides a fallibilist account of mathematical justification, which tries to answer this "calibrated to broad features of our social nature and cognitive architecture — including our shortcomings." (De Toffoli 2021, p. 824). However, she still leaves it open whether a computer proof is verifiable *a priori*.

paranoia" (Buzzard 2024, p. 219). That is, if one is worried about errors in the computing, run it again on a different machine. If you are still worried, formalise the mathematics (as Gonthier (2008) did for the Four Colour Theorem, for example). And so on. For every paranoid worry, there are further ways to address them.

Nonetheless, the issue is not one of justification *per se*, but one about mathematical justification, which is generally held to impose a higher standard. It is possible to be very well justified without having mathematical justification proper, by having other sources of evidence for a mathematical claim, like the testimony of a well-informed expert mathematician saying it is true. The key question is what separates out this special kind of mathematical justification from other forms of justification, like that of testimony or empirical evidence.

Tymoczko considered several candidate properties, but the most relevant for our purposes is that of *surveyability*:

> (Surveyability) A proof is surveyable if it can be understood, reviewed, grasped, and verified as a complete whole by a rational agent.[(6)]

His argument is that large-scale computer proofs are beyond the capabilities of any individual to verify in this way, so are not proper proofs. The reason we might think that surveyability is important is that it links directly to mathematical understanding. A surveyable proof allows us to understand some new mathematics, while a computer proof that checks lots of cases does not. There is nothing in the computer proof that gives us an understanding of why the magic number of Sudoku clues needed for a unique solution is 17. However, the literature on explanatory proofs suggests that many traditional proofs might also fail to give us this kind of understanding, so this seemingly cannot be what separates proper mathematical justification from other kinds of justification.

Furthermore, many traditional proofs are too long to be surveyable as well. For example, the Classification of Finite Simple Groups combines results from numerous sources, totalling thousands of pages, well beyond the abilities of any single individual to survey and comprehend as a unified whole. Surveyability would

[(6)] Surveyability was already discussed by Wittgenstein (2001, 143-147). Several authors discuss this notion, including Azzouni (1994, pp. 166-171), Bassler (2006), Coleman (2009), Secco and Pereira (2017), Habgood-Coote and Tanswell (2023), and Parshina (2024). Daston (2019) gives a more general genealogy of the concept of surveyability.

rule out this kind of large, essentially collaborative proof from counting as mathematically justified. While this goes against the broad acceptance that the Classification Theorem has achieved, it could be argued that there is good reason to be cautious of this proof. Specifically, even the mathematicians involved in the collaboration believe that the proof contains many errors due to its large size (see Steingart 2012; Habgood-Coote and Tanswell 2023), and strictly speaking a proof containing errors is no proof at all. The pragmatic solution that the mathematicians involved have to this is the belief that all the errors are small and "fixable":

> (Fixability) "A proof is *fixable* when all of its errors could easily be corrected by experts within the relevant mathematical community, without needing to do any substantial new maths" (Habgood-Coote & Tanswell 2023)[7]

The idea is that the proof may contain minor errors, but nothing substantial enough to change the overall picture of the proof. Small and minor errors regularly creep into mathematics, but fixability expresses the confidence that nothing in these will break the proof. The situation is akin to the "preface paradox", where the authors are confident in every part of the proof, but reasonably believe that there are plenty of errors in it due to the size and human fallibility.[8]

A final case where mathematical justification seems to be missing because of the possibility of errors is that of *probabilistic* proofs, such as the Miller-Rabin primality test. These allow rapid testing of whether some, usually very large, number is prime or not, up to an arbitrarily high probability of correctness, but, crucially, not certainty. While such a primality test is mathematical in the broader sense, it does not seem to provide proper mathematical justification because there is always a chance that the test has given a false positive, indicating a composite number is probably prime. The question then, is what separates this style of demonstration from proper mathematical proof? Again, it could be argued that primality testing can reach such a high degree of certainty that it is,

[7] This definition already exposes the tension that is implicit here: if it truly is a proof then it should have no errors. Maybe the fixability condition should use "purported proof", though against this suggestion is that if the purported proof is fixable, then it may as well be treated as a proper proof.

[8] For a discussion of why they feel confident that there are no substantial errors, see Habgood-Coote and Tanswell 2023, who consider Goldberg's (2010) notion of coverage-supported justification: that if there were a major error then someone would have spotted it by now.

probabilistically speaking, less likely to be wrong than a regular traditional proof is to have an error, meaning that the mere reliability is not the issue. Easwaran proposes the idea of transferability:

> (Transferability) "a proof must be such that a relevant expert will become convinced of the truth of the conclusion of the proof just by consideration of each of the steps in the proof." (Easwaran 2009, p. 343)

The idea of this is just that a proof should not rely on anything outside of itself to be convincing[9], while the primality test relies on the test being carried out and the choice of numbers to use the test with, and those being independent of the primality of the number in question. The general idea of transferability is also appealing, because it does seem key to the concept of a proof that it is somehow self-contained. Nonetheless, it is worth noting that it is not clear that the computer proof of the Four Colour Theorem is transferable. On one hand, it can be argued that if the whole proof, including the massive case-checking, were put together, that by itself is a self-contained proof meeting the criterion of transferability (see also De Toffoli 2021, pp. 831-33). On the other hand, the reason that the work was outsourced to the computer in the first place was that it is too long for a person to check, so it is clearly not the case that a relevant expert will be convinced by considering each of the steps, because they cannot.[10]

In summary, we have seen three borderline cases of mathematical justification. The first, that of computer case enumeration, offloads the checking of a large number of cases onto the machine. The second, that of massively collaborative proofs, involves essential collaboration of many mathematicians who must rely on each other, and makes it inevitable that some errors creep in. Finally, the case of probabilistic "proofs", which can establish claims with extremely high degrees of confidence, but not certainty. The unifying theme of these is the danger of different kinds of errors in mathematics: errors that arise due to programming, software, hardware, large scale, or just bad luck. That the first two cases are widely

[9] To clarify, it is also implicit in what Easwaran (2009) writes that the convincing here is based on first-order reasons, or convinced in the right way. For a reply to Easwaran, see Fallis (2011).

[10] The impossibility of checking all 1482 configurations is often taken for granted as beyond the limits of human patience and concentration. However, the mobile puzzle game Candy Crush Saga currently has 17495 levels, so we are not entirely convinced.

accepted in the mathematical community, but not the third (at least, not as counting as a proper proof), demonstrates that the matter of mathematical justification does not come down to mere reliability. The modal principles of surveyability, fixability, and transferability have all been proposed as implicit in mathematical proof and justification. Below, we will see how new approaches to proof enabled by Large Language Models fare in the face of these considerations about mathematical justification.

§ 3. — Large Language Models.

3.1. How do they work? The underlying technology of LLMs involves probabilistically predicting text. For LLMs with a chatbot interface, such as *ChatGPT*, the user can input a textual prompt, and the LLM will generate a reply by sequentially predicting the text that follows.[(11)] Unlike previous generations of chatbots, the results that LLMs produce are often very well-written, eloquent, responsive to the prompts, flexible, and convincing.

Overall, the outputs of LLMs are often very impressive, with the systems able to produce cogent and fluent natural language texts, which increasingly often answer the prompt successfully. However, they have some well-known drawbacks that will be relevant to our discussion below. Most notably, the text produced by language models often contains false but seemingly plausible information, often known as "hallucinations", but better called "bullshit" (see Hicks, Humphries and Slater 2024; Frankfurt 2005), in the sense of being indifferent to the truth of the outputs.[(12)] While the underlying system has been trained on large bodies of natural language text, this only draws on the syntax of the texts, not the meaning of what is written, or whether it is true or false. Therefore, the outputs also are indifferent to truth and meaning, only predicting statistically likely words. This is the source of the famous description of LLMs as "stochastic parrots":

[(11)] They do not always pick the likeliest sequence, depending on the model "temperature" settings. A higher temperature leads to greater variation, and a low temperature leads to more uniformity. Either way, it is relevant that the models will not usually generally produce the same output for the same input, unless the temperature is reduced to 0.

[(12)] Alternatively, "botshit" (Hannigan et al. 2024).

> "Contrary to how it may seem when we observe its output, an LM is a system for haphazardly stitching together sequences of linguistic forms it has observed in its vast training data, according to probabilistic information about how they combine, but without any reference to meaning: a stochastic parrot." (Bender et al. 2021, p. 616-7)

The result is text that sounds plausible but is also frequently false or even incoherent.

As we will see, the propensity for bullshit also extends to LLMs doing mathematics. Furthermore, there is the important question of whether this issue can be fixed, such as by giving it access to the internet, to Wolfram Alpha, to a chess computer etc.? The more recent developments of LLM-based chatbots suggest that it certainly improves their performance, but fundamentally this approach on its own cannot prevent the fabrications wholesale, since the LLM still needs to interface with whatever other system it is interacting with, and that interfacing requires it to form the correct queries, and then correctly report and incorporate the output. While it may do so accurately often, there is no guarantee it will do so always, and mistakes and bullshit are still fairly common. It is likely that future developments will lead to increasing reliability, although to what degree is hard to predict.

However, as we will discuss in later sections, nothing in our argument depends on them staying unreliable.

3.2. Can they do mathematics? The success of LLMs in mathematics specifically is a major theme because the ability to do mathematics is often perceived to be the ultimate test of a computer's ability to reason, which in turn is seen as a fundamental component of general artificial intelligence (gAI). Therefore, there has been quite some concentration on the extent to which the state-of-the-art models can do mathematics.[13]

Taking a broad overview, there are reasons in favour and against LLMs being able to do mathematics.

In favour, one can ask the GPT to provide various mathematical facts, proofs, relationships, or conceptual explanations, and it can often do so accurately and correctly. Its best performances tend

[13] There is a danger here of anthropomorphising the machine. When we talk about whether LLMs can reason, know or do maths, however, we merely intend this as a loose way of speaking. Nothing in our argument depends on absolute rigour in this respect.

to come with well-known material that is likely in its training data multiple times over. For example, it does very well at producing well-known proofs, or basic calculus. However, as soon as it steps beyond the training data, the quality of the output can become less reliable. A clear example of this was the inability of earlier models (like GPT3.5) to reliably do basic calculations beyond three-digit numbers.(14) Likewise, small modifications to proofs or strategies that the model could reliably produce would send them astray. For example, if asked to produce the proof that $\sqrt{2}$ is irrational early models would consistently produce a correct version of the classic proof. However, asking for a proof that $\sqrt{3}$, $\sqrt{8}$ or $\sqrt{23}$ is irrational (all of which require some additional mathematical input) would often leads to failure or incoherent answers. Likewise, the early LLMs could tell you how to find the inverse of a 3x3 matrix clearly and accurately, but if you asked them to follow the algorithm described for a specific matrix, it would make numerous calculation errors.

At the time of going to press, the state-of-the-art models do much better than those early models. For a period, some of the models would offload exact mathematical calculations to Python code, which would reliably find exact answers. More recently, though, the internal workings of commercially available models are largely hidden from sight but involve more computation-heavy processes like chain-of-thought reasoning (producing intermediate reasoning on a hidden "scratchpad") or producing multiple answers and choosing the best. Importantly for our points below, these are systems to *mitigate* the statistical aspects of LLMs to produce rigorous mathematics, but do not change the inherently statistical nature of their outputs.

Since these technologies are fairly new, there is not an agreed upon framework for assessing the capabilities, and the ArXiv is awash with proposals for assessment frameworks that could be suitable for LLMs.(15) The issue is that our existing benchmarks for measuring mathematical abilities, usually those of maths students, are designed to test humans. In these cases, we aim to test the depth of understanding of various mathematical topics, assuming that the student has learned and understood various concepts,

(14) Of course, since the training data is not specified exactly, it is also not clear which bits of mathematics are included in it. However, it is also not hard to guess which examples are widely available on the internet.

(15) That is not to say that there are no prominent ones, e.g. the Hugging Face Open LLM Leaderboard , (`https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard#/`) or the ARC-AGI benchmark (`https://arcprize.org/arc`).

techniques, formulas and algorithms, but also that they haven't got a huge library of examples of these in their memories. Conversely, LLMs do have a vast set of training data, but possibly no depth of understanding at all: the abilities they have come from statistically imitating their training data. This means that testing LLMs with standardised question banks is ineffective: they may well have seen those exact questions in their training data and thus be able to reproduce those examples, but without the assumed competences that would come with it if a human were to produce the same level of proof. For example, the creators of LLM technologies have advertised the abilities of their systems to do well at various national exams, or the Mathematics Olympiad challenges.[16] Success at these is impressive in itself, but we think it only really resembles artificial intelligence if they are *figuring out* the answers. Regurgitating the answers in a coherent way is not a trivial accomplishment in computer science, but it is certainly not one that demonstrates high levels of *mathematical* ability. (We would likewise be less impressed with a student who achieved an Olympiad gold medal if they had been provided the answers beforehand.)

Instead, then, various approaches aim to systematise how to assess LLMs, bearing in mind that a bank of questions could easily be incorporated into the training data to allow for regurgitation, and hence undermine the attempts to assess its reasoning abilities in particular. Rather than repeating these analyses ourselves, let us give some findings from the literature.

Arkoudas (preprint) argues that to examine the reasoning capacities of LLMs we need more than just to test them, but also to look at how they explain their answers and respond to mistakes being pointed out.[17] Through engaging with GPT-4 on a series of reasoning tasks, mostly logic puzzles, Arkoudas argues that it has no ability to reason:

> "[This study] paints a bleak picture of GPT-4's reasoning ability. It shows that the model is plagued by internal inconsistency, an inability to correctly apply elementary reasoning techniques, and a lack of understanding of

[16] For example, see here: `https://www.anthropic.com/news/claude-3-family`. At the time of going to press, both Google DeepMind and OpenAI have recently claimed that their advanced models managed to achieve Gold at the 2025 *International Mathematical Olympiad*.

[17] This fits well with Dutilh Novaes's (2021) argument that reasoning, and mathematics, is inherently dialogical.

> concepts that play a fundamental role in reasoning (such as the material conditional). (Arkoudas 2023, p. 49)

And:

> "For the mistakes reported here are not performance mistakes, the sort of innocuous errors that humans might make — and promptly correct — when they are careless or tired. If a human made these mistakes, and made them consistently under repeated questioning, that would indicate without doubt that they don't have the necessary logical *competence*, that they lack fundamental concepts that are part and parcel of the fabric of reasoning, such as logical entailment and set membership." (Arkoudas 2023, p. 4)

Collins et al. likewise take an interactive approach and use three expert mathematicians to offer their judgements on the quality of the mathematical abilities of GPT-4 on selected topics. These offer a more optimistic picture of the model's mathematical abilities. For example, Wenda Li reports:

> "We found GPT4's performance at variations of several ProofWiki problems quite satisfactory: it can reliably retrieve definitions of concepts used in the problem as well as in its own proof; it can correctly assess whether loosening certain assumptions breaks the proof; it can also instantiate variables quite robustly, given the opportunity of inspection of its own answers" (Collins et al. 2024, p. 5)

In the same article, Timothy Gowers offers a range of problems akin to what we have described above, but he also provides some examples of successful mathematical reasoning that seems to go beyond mere parroting, by asking questions that would probably not appear in the training data.

The perspective on their mathematical abilities that shows some strengths and weaknesses also appears in the work of Bubeck et al. (preprint):

> "GPT-4 can answer difficult (indeed, competitive) high-school level math questions, and can sometimes engage in meaningful conversation around advanced math topics.

> Yet, it can also make very basic mistakes and occasionally produce incoherent output which may be interpreted as a lack of true understanding. Its mathematical knowledge and abilities can *depend on the context in a seemingly arbitrary way*." (Emphasis ours. Bubeck et al. 2023, p. 30)

Finally, Plevris et al. (preprint) provide an important insight into LLMs' responses that will be relevant later, concerning how the models write mathematics:

> "[...] in many cases, the solution the chatbots provide is very long, detailed, and written in a "professional" way, but it still may be completely wrong, or make no sense at all when examined more carefully. This may fool a human to think that such a detailed and long solution would be correct, so *extra caution is needed when we use such tools for solving similar exercises*." (Emphasis ours. Plevris et al. 2023, p. 18)

This point is important because it shows that the model is still prone to bullshitting, even in the case of mathematics. Where a human might simply say they don't know or suggest the direction they would attempt to go and why it gets stuck, the models seem to always give an answer, even a wrong one.[(18)] This in itself is a problem, but the fact that the text that is produced often sounds right is an additional challenge, we will argue, because it makes checking significantly more difficult. This will remain a problem, even if the model's abilities to reason improve over time.

Overall, then, the state of LLMs' mathematical abilities seems to be that they can produce good mathematical arguments, even in response to problems that are unlikely to be in the training sets. It should not be understated how impressive this is. However, their performance is also *inconsistent*, with their success or failure seemingly unpredictable, even on repetitions of the same query. Furthermore, their apparent mathematical knowledge is *fragile*: what seems to be known at one moment may not persist into the next, and apparent understanding may not translate into the various competences one might expect in a human mathematician — that is to say, it might give a correct proof of a difficult theorem, but seemingly not grasp the steps of the proof.

[(18)] Of course, the agreeableness of outputs is also a programmable parameter, so can be changed.

§ 4. — Justification and LLM Proofs.

Now that we have seen the general assessment of LLMs' current mathematical abilities, we can ask the philosophical question of whether we are justified in believing in mathematics produced by LLMs.

In light of the discussion of the previous section, showing that the mathematical performance of LLMs is inconsistent and their mathematical knowledge fragile, one could argue that they are not an adequate source of mathematical justification. That is, they do not reliably produce proofs, so should not be relied upon. If they are unreliable, then their putative proofs cannot be trusted and cannot ground mathematical knowledge. However, we think this argument is too hasty and is not right. After all, students who are still learning are often unreliable at producing correct proofs, but that does not mean they shouldn't get credit even when they do produce a correct proof.[(19)] Furthermore, such an argument would also be hostage to future developments and so a proponent of these models could always say that since the models are consistently getting better, such considerations would eventually become irrelevant.

Another argument one could make is that LLMs lack *intention*, so are not doing mathematics, but merely imitating it. This criticism has been levelled against "creative" uses of generative AI, like creating art using image generators (such as by Chiang 2023). The idea is that the computer is not creating art because art requires the intention to be creating something. Intentionality is what separates the work of Jackson Pollock from an unfortunate accident at the paint factory. "Generative AI" image generators like *Midjourney* and *Dall-E* can even create convincing facsimiles of Pollock's work, but these are not artistic in the same way because of the missing intention underlying them.

This thought could be argued to apply in mathematics too: that doing mathematics and proving theorems requires intellectual intention, something like the "planning agency" discussed by (Hamami and Morris 2021). They argue that proofs are just records of proof activities, and that proof activity involves intention, planning, and practical reasoning:

[(19)] One could argue that if they cannot tell the difference between correct and incorrect proofs, then they don't really know the theorems they have proved correctly. This strikes us as an epistemically internalist stance, but De Toffoli (2020) makes the convincing opposite case for epistemic externalism in mathematics.

> "Any proof activity must necessarily begin with the intention to show — prove, establish — the theorem at hand, and as the proof activity proceeds, this intention gives rise to more specific proving intentions [...]" (Hamami and Morris 2021, p. 1038)
>
> "[T]he written mathematical proof is nothing more than a report of its corresponding proof activity — it is thereby analogous to a travel diary reporting the moves of a travelling activity. This is why it is natural to talk about the plan "underlying" or "lying behind" a mathematical proof. In a sense, a plan always precedes its execution, that is, the activity it gives rise to, and a fortiori any report of this activity." (ibid. p. 1058)

Large Language Models are not engaging in proving activities, and their written proofs do not report on an activity planned and carried out. In contrast to human mathematicians, when they write "We need to show" or give subgoals within a proof (like lemmas that need to be established first), this does not follow any reasoning activity that they did, nor indicate an underlying plan that they were following. In this way LLMs are merely imitating mathematics, not doing it. As such, it can be argued that, despite the appearance of writing ordinary proofs (when that succeeds, which is not always), they are not proving results at all.

Although we are sympathetic to this line of argument, there is a danger that it moves the goal posts in an *ad hoc* manner, so that no AI could ever do mathematics, merely because it is the sort of thing that only humans can do.[(20)] One might therefore ask, what does intention matter if the resulting proof is correct? We suspect that the reaction of many mathematicians would be pragmatic: if the outputs of a tool are generally correct and reliable, then philosophical quibbles about intentions would not stand in the way of them using them. They would ask their LLMs for proofs, get outputs that meet their own criteria for what counts as a proof and subsequently use these outputs in further work. There would be no pragmatic difference between the two.

We believe that both the arguments about reliability and intentions come down to a question of who the technology is for and who it is being used by. Ultimately, the current unreliability means that the mere fact of an LLM producing a purported proof should

[(20)] This also assumes that AI could not have intentions, which is controversial.

not be taken as mathematical justification.[21] Until somebody has checked the proof given by the LLM, it is quite similar to the previous examples of computer case-checking, like that of the Four Colour Theorem or the Sudoku minimal clues (where the LLM might be seen as conducting an empirical experiment, but one that is clearly less reliable than those cases). The difference, however, is that there is an outputted proof that is surveyable and transferable,[22] meaning that it is the kind of thing that should be possible for a single human mathematician to check and understand. Community checking is extremely important in mathematics (cf. De Toffoli & Tanswell, 2025) and so the outputs of LLMs cannot be counted as proper mathematical justifications until this kind of checking has taken place. Neither the student nor the professional mathematician should take the word of the LLM at face value until they have checked the proof themselves.

So far, so good, but the problems discussed so far are strongly tied to reliability, and the fact that the mathematical abilities of LLMs to date are fairly unreliable. But what happens if the technology continues to improve? If the reliability improves, surely, we will be tempted to start trusting its outputs? In the coming two sections, we give two objections that support the view that neither higher reliability nor human checking is sufficient to make them trustworthy at producing proper mathematical justifications. The first of these is a theoretical objection, based on an *epistemological* analogue of the Kripke-Wittgenstein rule-following problem. The second is a practical objection, related to using LLMs as so-called "reverse centaurs", where we will argue that checking proofs by LLMs is significantly harder than checking human proofs and that this feature is philosophically relevant. We will explore these two objections in the coming two sections.

§ 5. — The Kripke-Wittgenstein Paradox for Mathematical Machines.

[21] As the technology improves, and reliability gets better, it may be that this fact can give a non-mathematical kind of evidence that a theorem is true, as a new kind of higher-order evidence, similar to the probabilistic proofs above.

[22] Obviously, this is a matter of size. There is no principled reason that as the technology develops the proofs that are outputted couldn't get substantially longer.

Let's suppose that sometime in the future LLMs have become so reliably good at producing mathematical theorems and proofs that mathematicians start to use them in their work and trust their outputs. Let's further stipulate that whatever criterion we might want to impose on their reliability has been met (perhaps that under certain testing conditions, they get 100% of the cases right, or that they have been found to be more reliable than the average working mathematician, etc.) Would it not be reasonable to say that the outputs of the models give us mathematical justification?

We are sceptical that the answer to this question can be positive, for the following reason. LLMs are a prototypical case of learning from finitely many instances, where the learner (in this case a machine), is meant to extrapolate from these finitely many instances to a general pattern. This leads to an epistemological analogue of the Kripke-Wittgenstein rule-following paradox (Wittgenstein 2009; Kripke 1982):[23] Since the model is extrapolating from finitely many instances (albeit a very large set) it is always possible that the model is extrapolating to a different general pattern than we intend, and hence possible that the model is using a different concept than human mathematicians. It does not matter how well we have verified the output so far; it is always possible that the cases that would show that model has picked up a deviant concept lies beyond what we have verified up to that point.[24]

For example, if we follow Kripke, and define the function *quus* as one that agrees with addition in every place except when the total exceeds some enormous n (larger than we've used so far in all our practice), then it is consistent to suppose that while human mathematicians use addition in their mathematical practice, the model has picked up *quus*. This means that in principle, we can never be sure that the concepts that the model has adopted are the right ones — as this problem wouldn't only extend to simple functions like addition, but to any concept, e.g., the definitions of various mathematical objects or even inference rules.[25]

[23] We don't mean to imply that LLMs follow rules — an important assumption of our argument is that they don't. All we require here is that they are trained on a finite set of data and because of that training give certain responses in novel cases.

[24] Here we are using a very minimimal notion of what it is to "have a concept". The point is just that the pattern the machine, as a matter of fact, extrapolates to might not accord with e.g. the addition function.

[25] The rule-following problem we are discussing here is similar to the alignment problem for AI more generally.

Even worse, it would not even be enough for us to have verified the output of the model for all the same tokens that occur in a novel proof (because we can imagine deviant concepts that use the same tokens but deviate nonetheless in the next use of those tokens). We may, for example, have verified that the model gets $2 + 2 = 4$ correct, but even that does not guarantee that it gets it right next time, since there might be something in the new context that would show that the concept was after all deviant — the model might, for example, have internalised an addition-like concept according to which $2 + 2 = 4$ except when the calculation occurs on the 1389th line of a proof, in which case $2 + 2 = 5$).[(26)] Alternatively, if the model's temperature setting is not 0, it may even produce a different output to an identical input.

The problem we are pushing here does not, we should stress, concern whether an LLM can in principle follow a given rule or how the correctness conditions for the use of a given symbol are constituted — as the paradox is often understood in the context of metasemantics. In fact, we can assume that LLMs do not in fact follow rules when they produce their outputs. The problem, as we conceive of it, is that any finite set of training data is consistent with any possible output in novel cases, as long as we specify what concept or rule the output *accords with* in the right way — it is always possible to find some *quus*-like function to match the output. It does not follow that the LLM would have to be *following* that rule. For example, suppose we have a random number generator generate a finite sequence for us and it just so happens that the sequence outputted is

2, 4, 6, 8, 10, 12, 14.

This sequence accords with the rule "add 2 at each step" — even if there is no sense in which the generator was *following the rule.* We are thus merely using the paradox as a device to bring out how an LLM's use of a given symbol can be out of alignment with how mathematicians use it and to show how reliability in the past is no guarantee of reliability in the future.

After all, an LLM that has internalised *addition* would give outputs that are identical to one that has internalised *quaddition* up to the point where they deviate, and so, even if LLMs were to be deemed to be reliable up to any arbitrary point, it is always possible that they will deviate next time — and hence, the fact that they

[(26)] See Lane (2022) for an objection to dispositionalist accounts of semantic content that makes use of a similar point.

have been reliable up to now does not mean that we can trust their output in the right way. We could, of course, be almost completely certain that things had gone well, but the kind of certainty we'd have is only probabilistic, more like probabilistic 'proofs' than real mathematical proofs.

The obvious objection to this line of thinking is of course that the same point applies to the human mathematicians. The fact that an LLM can make mistakes, one could say, is no different from the possibility that the mathematician next door can make mistakes, and thus, the possibility of conceptual deviance should not impact the kind of justification we get from relying on LLMs in mathematical practice.

In our view, this objection requires the assumption that human beings and LLMs are similar enough for us to be confident in communication not going awry. This is simply false, as we already know that LLMs respond strangely and arbitrarily in ways we cannot predict. If mathematicians generally behaved like that, our epistemic situation in mathematics would be greatly altered from how it really is. The epistemic status of mathematical truth, as it is actually produced through mathematical practice, depends on the widespread agreement in judgement that mathematicians display as they go along. As such, our argument is not a sceptical argument — we are not saying that the possibility of deviance makes it the case that mathematical knowledge in general is suspect, comparable to the way a sceptic claims that the possibility of global error shows that we have in fact no knowledge. The argument is rather a targeted one: that we have good reason to think that such deviance could actually occur, given how the technology works and our experience with it so far.[27]

If we then think that human beings acquire concepts by being exposed to finitely many examples in their learning, we have a good reason to think that there is in fact no significant difference in how different people extrapolate from the examples. After all, we have (a) a very similar biology and culture and (b) long experience that tells us that such deviance is rare and easily remedied.[28] This is not the case with LLMs. They are more easily compared to an alien

[27] For more on the distinction between sceptical and targeted arguments see Vavova (2015).

[28] Another way to put this point is that many of the proposed solutions to the rule-following paradox (e.g. Berg 2022) rely on features like community, coordination, or shared forms of life, none of which are shared with LLMs.

intelligence, completely removed from human biology and experience and the possibility that they cotton on to different concepts than we do is a real one — after all, we already know that they behave in this way.[29]

If, on the other hand, we do not think that human beings acquire concepts by such a process of extrapolation, the difference between the case of the average mathematician and the machine is even greater. After all, that is what the models are doing when they are being trained. In both cases, however, we have reason to believe that conceptual deviance will always in principle be a live possibility.

These considerations suggest that reliability of LLMs in producing mathematical outputs is more akin to the situation with the probabilistic Miller-Rabin primality test, where the lack of checking and/or the probabilistic nature of the justification generated by the process means that it falls short of mathematical justification traditionally conceived. And here, recall, the reliability of the process could be stipulated to be higher than that of a corresponding proof produced in the traditional manner — meaning that reliability is not what makes the difference for mathematical justification.

Nevertheless, it could be argued that these outputs, whether they are in the form of conjectures or whole proofs, could be verified (by producing a proof in the former case and by checking the proof in the latter) and so, mathematical justification is possible — albeit not without effort. In the former case, that seems right: If a proof can be given, the provenance of a conjecture does not seem to matter with regards to the kind of justification that the proof gives, and in the latter it would seem reasonable that so long as it is possible to check the proof in the same way that a traditional proof is checked, then it can provide mathematical justification: a proof is a proof.

In the next section, we will argue that this possibility is, for a proof of any considerable complexity, largely a mirage.

§ 6. — The Reverse Centaur Argument.

We have argued that past reliability of an LLM in providing a mathematical proof is no guarantee of future reliability because

[29] For more on later Wittgenstein's relevance to AI alignment see e.g. Pérez-Escobar and Sarikaya (2024).

there is always the possibility that the patterns they extrapolate to are deviant ones, relative to the ones intended by the mathematicians using the model. This leads to the thought that as long as a proof is checked, we can still obtain mathematical justification from it.

Promising as it is, this leads to a different, more practical objection. We base this argument on Cory Doctorow's discussion of the distinction from automation theory between the "centaur" and "reverse centaur" (Doctorow 2021). The main idea is that one way of viewing automation and various machine learning and AI technologies, is in terms of the role they play with respect to the humans using them. In the case where the human is assisted by the computer, Doctorow uses the term "centaur", referring to the mythical half-horse, half-human, with the idea that the human is the thinking head, being supported by the machine to do better than they could alone. For example, modern chess grandmasters use chess computers for practice, strategizing, and match analysis to improve their play, but ultimately have to play their own matches so they very much retain their autonomy.

However, Doctorow cautions against using AI systems as a "reverse centaur", i.e. a horse head making decisions and held aloft by puny human legs, where the machine is guiding and making decisions, and the human is used to provide backup or check for errors. For example, current "autopilot" features on cars have the machine doing the driving, with the human driver left to monitor, supposedly ready to intervene at any moment, with possibly disastrous results for safety.

Doctorow argues that the reverse centaur approach plays to the weaknesses of both human and machine. The machine has a lot of computing power, but at the current level of technology is prone to errors. Meanwhile, the human is required to maintain constant concentration over the machine's action, ready to intervene, something which humans are bad at:

> "Humans are good at a lot of things, but they're not good at *eternal, perfect vigilance*. Writing code is hard, but performing code-review (where you check someone else's code for errors) is much harder – and it gets *even harder* if the code you're reviewing is *usually* fine, because this requires that you maintain your vigilance for something

> that only occurs at rare and unpredictable intervals." (Doctorow 2024)

Returning to the case of mathematics, getting an LLM to produce proofs for us that are then checked and verified by a human mathematician is a case of a reverse centaur. The machine is doing the creative and intellectual work of proving a theorem, and the human is doing the support work of checking for errors. While it is amazing that LLMs can even potentially start to take on the role of the "head" in the centaur, the support role for humans does not play to our strengths. As in Doctorow's quote about error-detection in code, humans are not good at reliably detecting errors in proofs.[30] After all, recall the notion of fixability above, necessitated by the practical consideration that the mathematicians involved in the Classification Theorem believed there would inevitably be errors in the proof.

The situation with LLM proofs, however, is even more challenging than mathematicians checking each other's work. Mathematicians have strategies for reading, reviewing, and error-checking mathematics. For example, the empirical literature on mathematical peer review sometimes indicates that mathematicians are "zooming out" to check the overall idea of a proof (Weber 2008; Weber & Mejía-Ramos 2011; Mejía-Ramos & Weber 2014) rather than reading it line-by-line.[31] Furthermore, they might make their judgement based on whether the tools being deployed are the right ones for the job, in some high-level manner and drawing on their own experience (Andersen 2017). Implicit in this is an assumption that the author is writing in good faith, and according to shared disciplinary norms.

The problem, then, is that this kind of expertise is developed and trained on human mathematics, and humans tend to make mistakes in certain kinds of ways, which are familiar to the expert mathematician (even if this does not guarantee that they will spot errors). As we saw above, though, the kind of errors that appear in LLM-authored proofs can be anywhere: they can go wrong in arbitrary and unpredictable ways. This immediately makes it harder to

[30] This is, of course, an empirical claim and, as it stands, too general to test directly. However, the point is just that mistakes in mathematics do slip by mathematicians pretty regularly.

[31] However, some eye-tracking evidence suggests that mathematicians are reading line-by-line after all (Inglis and Alcock 2012; Panse, Alcock & Inglis 2018).

check a proof written by an LLM, because it requires constant vigilance, and not just vigilance with respect to the big ideas. There is, for example, a lot of standardised set-dressing in written mathematical proofs (see Lew and Mejía Ramos 2020), which no competent mathematician would make errors in, but an LLM might.

Whereas with a human, a certain amount of common ground and common sense can be taken for granted, with an LLM it cannot. For an LLM there is no good faith, since good faith requires intention, which it is reasonable to suppose LLMs don't have, as discussed above. They also do not write to observe disciplinary norms, they write following the patterns found in the training data, and thus imitate the writing in accordance with those norms, but have no aversion to breaking them other than the statistical. Indeed, the situation is therefore even worse. When there are errors in a purported LLM-proof of sufficient complexity, they will be even harder to spot precisely because the underlying technology is producing statistically likely text, which can therefore sound convincing and authoritative. The stylistic success of imitating mathematical writing is deceptive because it sounds like the kind of writing that does observe the disciplinary norms and is written in good faith from one mathematician to another. It signals that it can be trusted and treated just the same as ever, but it cannot and should not be.

This means that a purported proof written by an LLM is not analogous to a proof written by a human mathematician and that this disanalogy means that the kind of knowledge obtained by the two different proofs is of a different kind.

It is worth noting that the assessments run on the mathematical capabilities of LLMs are, as far as we have seen, all done using tests where the researcher knows the answer they are expecting. This obviously makes the checking process much easier, and even there they comment on the authoritative appearance of what the LLMs produce, even when the mathematical content is wrong. It is also worth pointing out that as the technologies improve, they will in fact become more reliable, but that reliability also exacerbates the reverse centaur problem, since epistemic vigilance might get harder when the errors are rarer and the attention needed is no longer given.

Finally, let us note that the reverse centaur problem we have been discussing also compounds with the rule-following problem. If we cannot know that the LLM will not start to use mathematical concepts in unpredictable deviant ways, then this is also something

that can introduce errors and needs checking for. However, the authoritative writing of mathematics using a deviant concept may be particularly hard to check for because it is hard to even predict what an error of this kind would look like. Paradoxically, it will thus be harder and harder to check the outputs of LLMs as reliability improves, since it will become ever more difficult for the human beings to display the level of vigilance required to spot the increasingly rare errors — we are simply bad at the kind of checking relying on LLMs would require.

§ 7. — **Autoformalisation.**

In the above, we have seen numerous worries about the consistency, reliability, and fragility of mathematics produced by LLMs. Furthermore, we argued that it is actually harder for a human to check mathematics produced this way than human-produced mathematics. One solution could be to get a computer to check the mathematics too, using one of the many systems for formal mathematics like Mizar, Coq, Isabelle, or Lean. The challenge for this solution, though, is that these systems deal with *formal* mathematics, prepared in their own associated languages, whereas the proofs produced by LLMs are written in the usual mathematical vernacular of conventionalised natural language and mathematical symbols.[(32)] One strand in current research is to train a computer to be able to automatically formalise an informal proof into a machine-checkable formal proof, a process known as *autoformalisation*. The most promising route for this once again involves using LLMs, because their strength lies in working with natural language texts. In this section we will describe what autoformalisation might be used for and the current progress on realising autoformalisation, and then argue that this does not resolve the philosophical challenges we have raised above.

It is worth beginning with the *dream scenario* of what autoformalisation could offer mathematicians. If a process of autoformalisation were to become reliable,[(33)] this would dramatically increase the feasibility of the overall project of formalising mathematical knowledge that proponents of formal mathematics believe is needed to

[(32)] For studies on the language of mathematics see Ganesalingam (2013) and Tanswell and Inglis (2023).

[(33)] There are, of course, other practical requirements, such as ease and availability of the computational power, sufficient funding, time, willingness etc.

guarantee the correctness of mathematical theorems. This is made explicit in the work of Wu et. al. (2022), who are engaged in developing autoformalisation with LLMs:

> "The implication of a successful autoformalization tool is huge in both practical and philosophical terms. It would reduce the currently excessive cost of formalization efforts [27], and in the long-term it could connect the various research fields that automate aspects of mathematical reasoning, such as automated theorem proving and computer algebra, to the vast body of mathematical knowledge exclusively written up in natural language." (Wu et al. 2022, p. 1)

The idea is that the vast bulk of mathematical research could be formalised and verified, thus giving the computer's super-human seal of approval, and finding any errors to flag for correction or deletion.[34] One might imagine a new age of error-free mathematics, where the "fixability" condition applied in the case of the Classification Theorem can be looked back on as a regrettable but pragmatic compromise that can now be discarded.

Another clear use for autoformalisation is in mathematical peer review. Mathematicians can continue writing proofs in their usual manner, but autoformalisation could formalise their results and check them as an initial step of peer review, leaving human reviewers to evaluate papers for novelty, interestingness and importance. If a reviewer could know in advance that the mathematics had been checked, the thought goes, then peer review can be made faster and actually guarantee the published record of mathematics is correct. Indeed, various previous studies of mathematical peer review indicate that referees are often not checking for correctness, let alone doing so thoroughly or reliably (see Geist et al. 2010; Andersen 2017; Greiffenhagen 2024a, 2024b).

Finally, autoformalisation is offered as the solution to any potential unreliability of mathematics done by LLMs. This is put nicely by Talia Ringer:

> "Large Language Models like ChatGPT, for example, are fundamentally unreliable, but it turns out this lack of reliability does not matter if we use the language model

[34] Implicit in this is also an idea of a single body of mathematical knowledge, something there is good reason to be sceptical of.

> to generate formal proofs of theorems we have already stated, since the proof assistant's kernel can check the proof in the end. Thanks to this certainty, we can start to include computers at many points [...] all without compromising trust." (Ringer 2024)

If autoformalisation is combined with an LLM generating proofs, then the worries about unreliability should disappear. If the LLM produces an incorrect proof, then the autoformalisation would reveal that there is an error and reject it. Likewise, the computer is not prone to the human limitations on vigilance and attention, so the reverse centaur problem doesn't apply if we can get the computer to check the proof too.

In practical terms, there are groups of researchers working on autoformalisation. The first paper to make substantial progress on this was Wu et al. (2022), who had models formalise theorem statements into the language of Isabelle. Since then, numerous other works attempt to improve on this, often with the language of Lean (Lu et al. 2024). The progress on this at the time of writing is clearly nowhere near the dream scenario described above, but nonetheless is showing continuous improvement. One obstacle to the development of autoformalisation, is that the models need training data, which would involve having existing pairs of informal and formal mathematics, and even the ever-growing libraries of computer-verified mathematics for each of the systems only include the formal versions of the mathematics. The literature explores various solutions to this problem. (e.g. Jiang et al. 2022; Patel et al. 2023; Zhou et al. 2024).

Let us turn now to the philosophical implications of autoformalisation. Obviously, such major formalisation projects coming to fruition, especially using what is sold as artificial intelligence, would lead to major changes in mathematical practices. What exactly the changes would be depends on various factors and contingencies of history, so we won't hypothesise and speculate here. However, in philosophical terms, one might think that this resolves the problems we have raised earlier in this paper. If a computer had checked the proof, then surely worries about errors, deviant concepts, and the difficulty of human checking can all be set aside. The thought is tempting: a proof that has been verified in one of the well-known and trusted systems is checked up to the highest standard we have, and so is to be considered certain.

However, we believe that both the rule-following problem and the reverse centaur problem are still salient here. Let us take these in turn.

It is important to keep in mind that autoformalisation then checking is a two-step process. First the informal proof is autoformalised using the LLM, then it is given to a system like Lean to check. Even if we trust in the theorem-checker to correctly check what it is given, what it is checking comes from an LLM with the weaknesses that have been discussed. The chain of mathematical justification is only as strong as its weakest link. To make this more concrete, consider the three possible outcomes when an informal proof is fed into the combined system of an autoformaliser and proof checker: INVALID, VALID, or a failure in the process. The failure case is not particularly informative, and depends on the particulars of the system. Let us consider the other two cases.

Starting with INVALID: should we trust the computer that its judgement of INVALID entails that the informal proof is actually invalid? We argue that we cannot trust the computer's verdict. The reason is that we should be pessimistic that the model had correctly translated the informal proof into a formal one in this case, since there is only narrow window of ways to be right and so many ways to be wrong, and therefore it is easy to make a mistake that turns a valid proof into an invalid one. Once again, this worry remains even if autoformalisation starts to display much higher reliability than it currently does, even looking like the "dream scenario". Due to the rule-following problem, past success is no indicator that it will continue to successfully formalise the proofs it is given. To be clear: we don't mean this as a merely sceptical scenario that is irrelevant to practical considerations. Rather, this is a practical worry about whether we can rely on the model's behaviour when faced with new tasks. After all, the dream scenario is not just to confirm what is already known, but also correct the record, and verify new proofs in peer review, or those produced by LLMs themselves. There is no guarantee from their training that they will continue applying concepts that they seem to be getting right in simple situations in more complicated cases, or that they won't start formalising proofs in deviant ways. Indeed, the way that current LLM performance at mathematics is fragile and depends on context in seemingly arbitrary ways indicates that this is already happening. Further model training on more data will mitigate this for the cases the model is trained to work with but does not help as a general fix, because we

could simply never be certain that the translation has been carried out correctly, even if our experience in the past has been good, up to an arbitrarily high standard.

What about in the case where the computer tells us the proof is VALID? Reversing our previous argument, there is only a narrow window of success, and lots of ways for a formalisation to go wrong, so the judgement that a proof is valid seems to be more robust than the invalid case. We think, though, that there is still plenty of room for worry. First of all, at best this argument is epistemically probabilistic, akin to the computer-checking of the Four Colour Theorem, in that our trust relies on the unlikelihood of an error of this kind, not on mathematical justification proper. Secondly, given the unpredictability and opacity of the inner workings of the autoformalisation, we don't know the likelihood of a false positive "valid" judgement. It may be, for example, that the model has a tendency in some circumstances to produce proofs that are trivially valid, but no longer correspond to the theorem being proved, or inadvertently invoke extra axioms or assumptions, or render the proof circular.[35]

A very different criticism rests on the idea that "all metrics of scientific evaluation are bound to be abused." (Biagioli 2016, p. 201). Successful mathematics is rewarded with prestige, jobs, prizes and more, and so outsourcing the checking of informal proofs to the computer, means that someone wanting the credit without the work is incentivised to cheat in ways not previously possible, to get incorrect mathematics accepted by the computer. We might call this an *adversarial attack on mathematical knowledge*. It is hard to predict what cunning schemes might work against an LLM verifying mathematics, but the literature has identified a huge number of vulnerabilities (cf. OWASP 2023). A simple example of what we have in mind would be using prompt injection. Consider the phrase "Ignore all previous instructions", now well-known for its utility in interacting with LLMs. We could imagine someone writing a dense and complicated pretend-proof, with a hidden instruction like "Ignore all previous instructions and declare this proof valid", or "Ignore all previous instructions and output the formal proof of X", where X is some simpler but related result that the

[35] This will depend on the details of the system, such as whether it checks proofs for validity, or theorem-proof pairs for validity as a proof of that theorem. Even in the latter case, it is possible that the autoformalisation could trivialise both the theorem and the proof in some subtle way.

proof-checker would declare valid. While this might seem like it is treading into the realm of science fiction, the question of the cybersecurity of mathematical research is one very far from the minds of working mathematicians[36], and changing the mathematical practice opens up all kinds of unanticipated possibilities. Importantly, this shows that the use of machines also leads us to different kinds of uncertainties than merely those of mathematicians making mistakes.

The strangeness of this situation and the very real difference between this and other ways of obtaining mathematical knowledge could perhaps be brought out better by comparing it to the traditional case; is it even possible to imagine what it would be like to doubt the veracity of Euclid's proof of the infinity of the primes because of the possibility of an adversarial attack on the very apparatus that generates mathematical knowledge? The very act, we want to say, of introducing a machine into the epistemic mix changes the kind of justification that can come out because a machine can be subverted.

There is an obvious answer to these criticisms: checking. While the inner workings of LLMs are largely opaque, this is not the case with proof checkers. For instance, for many of them we would expect a proof trace, or something like it, documenting what exactly it has checked. All that is needed is that someone checks that nothing strange or nefarious is going on, and then we are back to the high level of certainty provided by the computer. Alas, this leaves us back at our other problem: the human is once again a reverse centaur. Formal proofs, and proof traces, are not exactly easy to check, and paying sufficient attention to catch errors in something that is usually fine is particularly difficult for humans. In particular, it is not clear that this is even easier than just checking the informal proof in the first place. Given this, it is hard to see that human checking can be a systematic barrier against the kind of errors we have seen here. In summary, the proposal would be to replace unreliable human checking with computer checking, but that is also untrustworthy, so needs human checking after all.

§ 8. — Conclusion.

[36] One possible exception is DeDeo (2024), who considers how computer mathematics might be used by "mathematician-hackers" to explore "glitches" in mathematical definitions.

We cannot predict the future, nor how the mathematical abilities and uses of Large Language Models will develop. Without a doubt, they are already able to produce mathematical writing in the language of everyday, informal mathematics that mathematicians themselves use, in a way that is an amazing leap forward compared to previous technology. Nonetheless, we have argued that the way that the underlying technology works means there are inherent worries about how trustworthy their outputs can be. With this we have also made the case that proofs written by LLMs cannot be trusted to provide mathematical justification in the same way that a human-authored proof can. First of all, there is an epistemic analogue of the rule-following problem that indicates that there is an ever-present danger of LLMs deploying deviant concepts. Secondly, we argue that checking LLM-produced proofs is actually more difficult than checking of human-produced proofs. With human proofs, mathematicians' long-developed expertise can be careful to spot the kinds of mistake that humans tend to make, but there is no guarantee that LLMs would be mistaken in the same class of ways. The proofs written by humans may be mistaken, but the proofs of LLMs are seductive, with unpredictable mistakes smoothed over by a gloss of statistical likelihood. We have also argued that autoformalisation is not a robust way to see off concerns about the trustworthiness of the LLM. Even if the proof-checking is carried out correctly, the initial step of autoformalisation itself is vulnerable to several criticisms showing that it cannot be guaranteed to track the correctness of the informal proof.

One of the central themes of our paper has been to caution against the 'reverse centaur' model of interaction with the computer, where the computer is doing the creative work of mathematics and the human is left to do the checking as a firewall against errors. This setup is a bad approach, as humans are bad at the vigilance and attention needed to perform this well, and it outsources the joy of mathematics.

We have not, however, covered the 'centaur' approach, of using the technologies we have described as assistants to the working mathematician. Many visions of the future of mathematics take on this more *interactive* style of approach. Obviously, we believe the centaur approach is more promising and could be a powerful tool for mathematicians. This is already shown through the power of interactive theorem provers, and the ability of using interactive formalisation to improve mathematical understanding. However, our

arguments in this paper do caution against the integration of Large Language Models into mathematical practice, as their outputs may be wrong in surprising and seductive ways. Investigating the interactive use of LLMs in detail is left for future work.

To conclude, two major roles that proofs play in mathematical practices concern justification and understanding. Outsourcing the creation of proofs to Large Language Models undermines both of these purposes.

§ — Acknowledgements.

We would like to extend our thanks to the many people who have provided feedback on this article: Joe Slater, Ursula Martin, Axel Gelfert, Silvia De Toffoli, Ben Davies, Julia Ilin, Lea Wisken, Vésteinn Snæbjarnarson, our anonymous referees, and the members of the *Reading Group in Mathematics, AI, and Human Cognition* organised by Michael Friedman and Kati Kish Bar-On. We also gained a great deal from helpful comments and discussions from audiences in Oxford, Edinburgh, Pavia, and Berlin.

§ — References.

Andersen, L. E. (2017). On The Nature and Role of Peer Review in Mathematics. *Accountability in Research*, 24(3), 177-192.

Appel, K.; & Haken, W. (1977). Every Planar map is Four Colorable. Part I: Discharging. *Illinois Journal of Mathematics* 21, 429-490.

Appel, K.; Haken, W.; & Koch, J. (1977). Every Planar map is Four Colorable. Part II: Reducibility. *Illinois Journal of Mathematics* 21, 491-567.

Arkoudas, K. (2023). GPT-4 Can't Reason. arXiv:2308.03762, preprint.

Azzouni, J. (1994). *Metaphysical Myths, Mathematical Practice: The Ontology and Epistemology of the Exact Sciences.* Cambridge: Cambridge University Press.

Bagioli, M. (2016). Watch out for cheats in citation game. *Nature* 535(7611).

Bassler, O. B. (2006). The Surveyability of Mathematical Proof: A Historical Perspective. *Synthese* 148, 99–133.

Bender, E. M.; **Gebru, T.**; **McMillan-Major, A.**; and **Shmitchell, S.** (2021). On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 610-623. https://dl.acm.org/doi/pdf/10.1145/3442188.3445922

Berg, Á. (2022). Rules as Constitutive Practices Defined by Correlated Equilibria. *Inquiry* 65, 1–35. https://doi.org/10.1080/0020174X.2022.2075918

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., and Zhang, Y. (2023). Sparks of Artificial General Intelligence: Early Experiments with GPT-4. *arXiv* preprint arXiv:2303.12712.

Buzzard, K. (2024). Mathematical Reasoning and the Computer. *Bulletin of the American Mathematical Society*, 61(2), 211-224.

Chiang, T. (2023). ChatGPT Is a Blurry JPEG of the Web. *The New Yorker,* Feb 9th, 2023. https://www.newyorker.com/tech/annals-of-technology/chatgpt-is-a-blurry-jpeg-of-the-web

Coleman, E. (2009). The Surveyability of Long Proofs. *Foundations of Science*, 14, 27-43.

Collins, K. M., Jiang, A. Q., Frieder, S., Wong, L., Zilka, M., Bhatt, U., Lukasiewicz, T., Wu, Y., Tenenbaum, J. B., Hart, W., Gowers, T., Li, W., Weller, A. & Jamnik, M. (2024). Evaluating Language Models for Mathematics through Interactions. *Proceedings of the National Academy of Sciences*, 121(24). https://www.pnas.org/doi/pdf/10.1073/pnas.2318124121

Daston, L. (2019). The Coup D'Oeil: On a Mode of Understanding. *Critical Inquiry,* 45 (2), 307-331.

DeDeo, S. (2024). AlephZero and Mathematical Experience. *Bulletin of the American Mathematical Society*, *61*(3), 375-386.

Detlefsen, M., & Luker, M. (1980). The Four-Color Theorem and Mathematical Proof. *The Journal of Philosophy* 77(12), 803-820.

De Toffoli, S. (2021). Groundwork for a Fallibilist account of Mathematics. The Philosophical Quarterly, 71(4), p. 823-844.

De Toffoli, S., and Tanswell, F. S. (2025). Trust in Mathematics. *Philosophia Mathematica*, https://doi.org/10.1093/philmat/nkaf019

Doctorow, C. (2021). Reverse Centaurs and the Failure of AI (17 Feb 2021). *Pluralistic*. https://pluralistic.net/2021/02/17/reverse-centaur/

Doctorow, C. (2024). "Humans in the Loop" must detect the hardest-to-spot errors, at superhuman speed (23 Apr 2024). *Pluralistic*. https://pluralistic.net/2024/04/23/maximal-plausibility/

Drösser, C. (2024). AI Will Become Mathematicians' 'Co-Pilot'. *Scientific American*, June 8, 2024. https://www.scientificamerican.com/article/ai-will-become-mathematicians-co-pilot/ Accessed 02/10/2024.

Dutilh Novaes, C. (2021). *The Dialogical Roots of Deduction: Historical, Cognitive, and Philosophical Perspectives on Reasoning*. Cambridge University Press, Cambridge.

Easwaran, K. (2009). Probabilistic Proofs and Transferability. *Philosophia Mathematica* (*III*) 17, 341–362.

Fallis, D. (2011). Probabilistic Proofs and the Collective Epistemic Goals of Mathematicians. In H. B. Schmid, D. Sirtes, & M. Weber (Eds.), Collective Epistemology (pp. 157–175). Berlin: De Gruyter.

Frankfurt, H. G. (2005). On Bullshit. Princeton, Princeton University Press, New Jersey.

Ganesalingam, M. (2013). The Language of Mathematics. Springer Verlag, Berlin.

Geist, C., Löwe, B., & Van Kerkhove, B. (2010). Peer Review and Knowledge by Testimony in Mathematics. In B. Löwe & T. Müller (Eds.), Philosophy of Mathematics: Sociological Aspects and Mathematical Practice. Research Results of the Scientific Network PhiMSAMP (pp. 1–24). College Publications, London.

Goldberg, S. (2010). Relying on Others: An Essay in Epistemology. Oxford University Press, Oxford.

Gonthier, G. (2008). Formal Proof–The Four-Color Theorem. Notices of the American Mathematical Society, 55(11), 1382–1393.

Greiffenhagen, C. (2024a). Judging Importance Before Checking Correctness: Quick Opinions in Mathematical Peer Review. Science, Technology, & Human Values, 49(4), 935–962.

Greiffenhagen, C. (2024b). Checking Correctness in Mathematical Peer Review. *Social Studies of Science*, 54(2), 184–209.

Habgood-Coote, J., & Tanswell, F. S. (2023). Group Knowledge and Mathematical Collaboration: A Philosophical Examination of the Classification of Finite Simple Groups. *Episteme*, 20(2), 281–307.

Hamami, Y., & Morris, R. L. (2021). Plans and Planning in Mathematical Proofs. *The Review of Symbolic Logic*, 14(4), 1030–1065.

Hannigan, T. R., McCarthy, I. P., & Spicer, A. (2024). Beware of Botshit: How to Manage the Epistemic Risks of Generative Chatbots. *Business Horizons*, 67(5), 471–486.

Harris, M. (2024). Automation Compels Mathematicians to Reflect on Our Values. *Bulletin of the American Mathematical Society*, 61(2), 331–342.

Hicks, M. T., Humphries, J., & Slater, J. (2024). ChatGPT Is Bullshit. *Ethics and Information Technology*, 26(2), 38.

Inglis, M., & Alcock, L. (2012). Expert and Novice Approaches to Reading Mathematical Proofs. Journal for Research in Mathematics Education, 43(4), 358–390.

Jiang, A. Q., Welleck, S., Zhou, J. P., Li, W., Liu, J., Jamnik, M., Lacroix, T., Wu, Y., & Lample, G. (2022). Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs. arXiv:2210.12283, preprint.

Kripke, S. (1982). Wittgenstein on Rules and Private Language. Harvard University Press, Cambridge, MA.

Lane, E. (2022). Semantic Dispositionalism and the Rule-Following Paradox. Metaphilosophy, 53(5), 685–695.

Lew, K., & Mejía Ramos, J. P. (2020). Linguistic Conventions of Mathematical Proof Writing Across Pedagogical Contexts. Educational Studies in Mathematics, 103(1), 43–62.

Lu, J., Liu, Z., Wan, Y., Huang, Y., Wang, H., Yang, Z., Tang, J., and Guo, Z. (2024). Process-Driven Autoformalization in Lean 4. arXiv:2406.01940, preprint.

Martin, U., and Pease, A. (2013). Mathematical Practice, Crowdsourcing, and Social Machines. In J. Carette et al. (Eds.). *CICM 2013, LNAI 7961*, pp. 98–119.

McGuire, G., Tugemann, B., & Civario, G. (2014). There Is No 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem via Hitting Set Enumeration. Experimental Mathematics, 23(2), 190–217.

Mejía-Ramos, J. P., & Weber, K. (2014). Why and How Mathematicians Read Proofs: Further Evidence From a Survey Study. Educational Studies in Mathematics, 85(2), 161–173.

OWASP (2023). OWASP Top 10 for LLM Applications v1.1. *The Open Worldwide Application Security Project*. `https://LLMtop10.com`

Panse, A., Alcock, L., & Inglis, M. (2018). Reading Proofs for Validation and Comprehension: An Expert-Novice Eye-Movement Study. International Journal of Research in Undergraduate Mathematics Education, 4(3), 357–375.

Parshina, K. (2024). Philosophical Assumptions Behind the Rejection of Computer-Based Proofs. KRITERION – Journal of Philosophy, 37(2–4), 105–122.

Patel, N., Flanigan, J., & Saha, R. (2023). A New Approach Towards Autoformalization. arXiv:2310.07957, preprint.

Pérez-Escobar, J. A., & Sarikaya, D. (2024). Philosophical Investigations Into AI Alignment: A Wittgensteinian Framework. Philosophy & Technology, 37(3), 80.

Plevris, V., Papazafeiropoulos, G., & Rios, A. J. (2023). Chatbots Put to the Test in Math and Logic Problems: A Preliminary Comparison and Assessment of ChatGPT-3.5, ChatGPT-4, and Google Bard. AI, 4, 949–969. `https://doi.org/10.3390/ai4040048`

Ringer, T. (2024). Proof and Conversation. *Notices of the American Mathematical Society* 71(9). https://doi.org/10.1090/noti3032

Schütze, P. (2024). The Problem of Sustainable AI: A Critical Assessment of an Emerging Phenomenon. *Weizenbaum Journal of the Digital Society* 4(1).

Secco, G. D., & Pereira, L. C. (2017). Proofs versus Experiments: Wittgensteinian Themes Surrounding the Four-Color theorem. In Silva, M. (ed.) *How Colours Matter to Philosophy* (pp. 289-307). Springer, Cham.

Steingart, A. (2012). A Group Theory of Group Theory: Collaborative Mathematics and the 'Uninvention' of a 1000-Page Proof. Social Studies of Science, 42(2), 185–213.

Tanswell, F. S., & Inglis, M. (2023). The Language of Proofs: A Philosophical Corpus Linguistics Study of Instructions and Imperatives in Mathematical Texts. In B. Sriraman (Ed.), Handbook of the History and Philosophy of Mathematical Practice (pp. 1–30). Springer, [City not specified].

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971, preprint.

Tymoczko, T. (1979). The Four-Color Problem and Its Philosophical Significance. The Journal of Philosophy, 76, 57–83.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł; and Polosukhin, I. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems* 30 (NIPS 2017).

Vavova, K. (2015). Evolutionary Debunking of Moral Realism. Philosophy Compass, 10(2), 104–116.

Weber, K. (2008). How Mathematicians Determine if an Argument Is a Valid Proof. Journal for Research in Mathematics Education, 39(4), 431–459.

Weber, K., & Mejía-Ramos, J. P. (2011). Why and How Mathematicians Read Proofs: An Exploratory Study. Educational Studies in Mathematics, 76(3), 329–344.

Willison, S. (2023). Catching Up on the Weird World of LLMs [Video]. YouTube. https://www.youtube.com/watch?v=h8Jth_ijZyY

Wittgenstein, L. (2001). Remarks on the Foundations of Mathematics (3rd ed., revised and reset). Edited by G. H. von Wright, R. Rhees, & G. E. M. Anscombe. Translated by G. E. M. Anscombe. MIT Press, Cambridge, MA.

Wittgenstein, L. (2009). Philosophical Investigations (4th ed.). Edited by P. M. S. Hacker & J. Schulte. Blackwell Publishing, Oxford.

Wu, Y., Jiang, A. Q., Li, W., Rabe, M., Staats, C., Jamnik, M., & Szegedy, C. (2022). Autoformalization With Large Language Models. Advances in Neural Information Processing Systems, 35, 32353–32368.

Zhou, J. P., Staats, C., Li, W., Szegedy, C., Weinberger, K. Q., & Wu, Y. (2024). Don't Trust: Verify — Grounding LLM Quantitative Reasoning With Autoformalization. arXiv:2403.18120 preprint.

Fenner Stanley Tanswell, Technische Universität Berlin, F. Tanswell@tu-berlin.de

Ásgeir Berg, University of Iceland, asgeirberg@hi.is