Mathematical Understanding

Jeremy Avigad

January 1, 2026

> We are not trying to meet some abstract production quota of definitions, theorems and proofs. The measure of our success is whether what we do enables *people* to understand and think more clearly and effectively about mathematics.
>
> William Thurston,
> "Proof and Progress in Mathematics" [64]

> Any fool can know. The point is to understand.
>
> attributed to Albert Einstein,
> [57, epigraph to Chapter 1]

Abstract

The philosophy of mathematics has traditionally been concerned with questions about the proper justification of claims to knowledge. The goals of mathematics, however, are not limited to making true mathematical statements; ultimately, we look to mathematics for *understanding*. This chapter considers the ways we talk about mathematical understanding in our assessments of historical and contemporary mathematical results, in our discussions of computers in mathematics, and in mathematics education. It argues that a theory of mathematical understanding should be a theory of *abilities* to think and reason mathematically, and that such a theory should inform, and be informed by, practical applications. It also situates models of reasoning and understanding in the context of symbolic and neural AI, which offer complementary perspectives.

1 Introduction

The distinction between knowledge and true belief is central to Western epistemology. In Plato's dialogue, *Meno*, Socrates asks a slave a series of questions that leads him to discover a theorem of geometry. He then uses the example to argue that acquisition of knowledge is a matter of recalling *forms* that our souls encounter before we are born, a process that serves to secure our beliefs and keep them from running away.

> ... true opinions, as long as they remain, are a fine thing and all they do are good, but they are not willing to remain long, and they escape from a man's mind, so that they are not worth much until one ties them down by giving an account of the reason why. And that, Meno, my friend, is recollection, as we previously agreed. After they are tied down, in the first place they become knowledge, and then they remain in place. That is why knowledge is prized higher than correct opinion, and knowledge differs from correct opinion in being tied down. [15]

For Plato, mathematical objects and concepts are paradigm instances of forms, geometric reasoning is a paradigm of recollection, and deductive argument is the paradigm form of the *logos* that secures knowledge. The philosophy of mathematics has been concerned with the nature of mathematical objects, inference, and proof ever since. These are the focus, for example, of the *The Oxford Handbook of Philosophy of Mathematics* [56], published in 2005.

Concerns about justification seem less pressing today, however, now that the foundational crises in mathematics are a century behind us and axiomatic foundations provide adequate and uncontroversial accounts of the norms of mathematical inference. Gödel's second incompleteness theorem places strong limits on what we can do to justify our choices of axioms, and twentieth-century philosophy of mathematics encourages us to accept that no philosophical justification is necessary. According to the logical positivists, mathematical and logical axioms are true because they are part of the linguistic framework we use to interpret the world, and according to Quine, they are true because they are part of our holistic web of beliefs. Either way, it is fruitless to look for external justification; mathematics is as mathematics does, and it is not the philosopher's job to question it. What, then, is left for a philosopher of mathematics to do?

Even when mathematical considerations dictate our choices of axioms and rules, we can seek more clarity as to why we do mathematics the way we do. Mathematicians try to do mathematics well, and they are generally thoughtful and reflective about their craft. As the first epigraph suggests, the goal of mathematics is not just to rack up definitions, theorems, and proofs, but to achieve mathematical understanding. As philosophers, we should join mathematicians in thinking about what that means. As a first cut, it is helpful to take the phrase "mathematical understanding" to stand for the things that we value in our mathematical artifacts that go beyond mere knowledge, that is, what we get from mathematical definitions, proofs, and theories, beyond knowing that some mathematical statements are true. Focusing on understanding is a way of picking up where twentieth-century philosophy of mathematics left off: if the ultimate justification for our axiomatic frameworks is their role in supporting mathematics as a whole, we would do well to come to terms with the goals of mathematics and the ways our mathematical practices achieve them.

This chapter makes the case that we can be both philosophical and scientific about mathematical understanding, and that our scientific theories should inform and be informed by our philosophical ones. I will advocate for a specific approach to thinking about mathematical understanding that focuses on studying the *abilities*, or capacities, that make mathematical reasoning possible. I will argue that we can do this in a way that abstracts these abilities away from the particular cognitive architectures that implement them, whether they are found in students, expert mathematicians, symbolic algorithms, neural networks, or even social and historical processes.

We will also consider two views of mathematical understanding that emerge from computational advances in the mechanization of mathematical reasoning. From the perspective of symbolic AI and automated reasoning, understanding is a matter of having algorithms, symbolic representations, and heuristics that enable the execution of complex mathematical tasks. From the perspective of machine learning and neural networks, understanding is a matter of having distributed representations, derived from experience, that enable recognizing mathematical patterns and synthesizing reasoning strategies. We will explore these complementary perspectives and the way they can inform our discourse on understanding.

While there is a growing philosophical literature on scientific understanding, there is considerably less on mathematical understanding (e.g., [13, 24, 31, 63]). This chapter is not a survey; rather, it revisits themes I have explored over the last two decades and situates them in the context of recent developments in AI.

2 Motivating Questions

Let us start with some examples of questions that a theory of mathematical understanding might help us answer.

2.1 Understanding vs. knowledge

From the traditional point of view, the purpose of proving a theorem is to establish that it is true, thereby justifying claims to know it. However, multiple proofs of the same theorem are often found in the mathematical literature [4, 17, 18].

Proofs convey all sorts of understanding. For example, in the *Arithmetic*, Diophantus notes that $5 = 2^2 + 1^2$, $13 = 3^2 + 2^2$, and $5 \times 13 = 65 = 8^2 + 1^2 = 7^2 + 4^2$; in other words, the product of two sums of integer squares is again an integer square. This is a general phenomenon:

Theorem. *If each of x and y is the sum of two integer squares, then so is xy.*

Proof. Suppose $x = a^2 + b^2$ and $y = c^2 + d^2$. Then

$$\begin{aligned} xy &= (a^2 + b^2)(c^2 + d^2) \\ &= a^2c^2 + b^2d^2 + a^2d^2 + b^2d^2 \\ &= a^2c^2 - 2abcd + b^2d^2 + a^2d^2 + 2abcd + b^2d^2 \\ &= (ac - bd)^2 + (ad + bc)^2, \end{aligned}$$

a sum of two squares. □

This proof uses nothing more than high school algebra, yet it leaves the result mysterious. For an alternative proof, consider the *Gaussian integers*, that is, complex numbers of the form $a + bi$, where a and b are integers. In this system of numbers, -1 has two square roots, i and $-i$. As far as the operations of arithmetic are concerned, the two are indistinguishable; if we replace i by $-i$ uniformly in any calculation using addition and multiplication, the calculation remains valid. In particular, if $\alpha = a + bi$, we define the *conjugate* $\overline{\alpha}$ of α to be $a - bi$, and we have that if $\alpha\beta = \gamma$, then $\overline{\alpha}\overline{\beta} = \overline{\gamma}$. This tells us that the conjugate $\overline{\alpha\beta}$ of a product is equal to the product $\overline{\alpha}\overline{\beta}$ of the conjugates, for every α and β.

Now, for any Gaussian integer, α, we define the *norm* of α, $N(\alpha)$, by

$$N(\alpha) = \alpha\overline{\alpha} = (a + ib)(a - ib) = a^2 - i^2b^2 = a^2 + b^2.$$

So the norm of $a + ib$ is the sum of squares $a^2 + b^2$, and any sum of squares $a^2 + b^2$ can be expressed as $N(a + ib)$. It is not hard to show that the norm of a product is the product of the norms:

$$N(\alpha\beta) = (\alpha\beta)(\overline{\alpha\beta}) = \alpha \cdot \beta \cdot \overline{\alpha} \cdot \overline{\beta} = \alpha\overline{\alpha} \cdot \beta\overline{\beta} = N(\alpha)N(\beta).$$

With that little bit of theory in hand, our theorem has a short proof.

Proof. Suppose x and y are sums of squares. If $x = N(\alpha)$ and $y = N(\beta)$, then $xy = N(\alpha\beta)$, a sum of two squares. □

Seeing this proof for the first time typically induces an aha moment. It elicits a warm feeling even in seasoned experts. The ideas—the study of number systems extending the integers and their symmetries—are fundamental to number theory today, and there is a lot we can say about why we like the proof [4]. The visceral feeling of having grasped something valuable makes it compelling that there is a phenomenon here that deserves explanation.

Mark Steiner distinguished between explanatory and non-explanatory proofs [60], initiating debate on the nature of mathematical explanation [47]. However, not every assessment of proof is cast in terms of explanation; for example, proofs are assessed on grounds of simplicity, generality, depth, and insight. At least some of these virtues bear on the way the proof contributes to mathematical understanding, and a theory of mathematical understanding should help us make sense of them. This raises a fundamental question: what do we value in mathematical proofs beyond their ability to tell us that a mathematical statement is true?

2.2 Understanding proofs

There is a reciprocal relationship between understanding and proof: a good proof conveys understanding, while reading and discovering proofs *requires* understanding. Understanding a proof involves, in part, being able to explain or justify the correctness of each step, but that is not all: it is not unusual for a mathematician to feel they have verified a proof line by line without fully understanding it. Poincaré sums it up nicely:

> Does understanding the demonstration of a theorem consist in examining each of the syllogisms of which it is composed in succession, and being convinced that it is correct and conforms to the rules of the game? In the same way, does understanding a definition consist simply in recognizing that the meaning of all the terms employed is already known, and being convinced that it involves no contradiction?
>
> . . . Almost all are more exacting; they want to know not only whether all the syllogisms of a demonstration are correct, but why they are linked together in one order rather than in another. As long as they appear to them engendered by caprice, and not by an intelligence constantly conscious of the end to be attained, they do not think they have understood. [52]

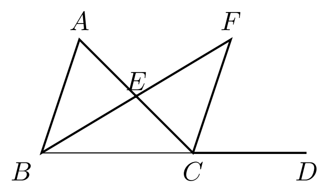
The logic of discovery and the logic of justification are thereby linked: understanding a proof involves seeing a process under which the proof could have been discovered. Rebecca Morris has argued that we consider a proof to be *motivated* when the presentation helps us see how each step could have been anticipated, given the context and goal [49]. Yacin Hamami and Morris have argued that understanding a proof requires seeing it as an instance of a rational plan, and being able to reconstruct a rational process by which it is generated [30].

Many authors have noted that formal proofs do not convey the same understanding as informal proofs, and there is a substantial philosophical literature investigating the gap between the two (see [28, 59] for overviews). A standard view holds that formal proof is an idealized standard of correctness, in the sense that an informal proof is correct if and only if it appropriately warrants the existence of a formal derivation. However, formal derivations can be inordinately long, and even a single error invalidates the conclusion. This poses a clear objection to the standard view: how can our judgments of the correctness of an informal proof possibly warrant the existence of something so complex and fragile? I have described one approach to answering this question [9]. In a nutshell: we don't gain confidence in the correctness of proofs by checking them formally; rather, we *understand* them.

2.3 Diagrammatic reasoning

Proposition 16 of Book I of Euclid's *Elements* reads "In any triangle, if one of the sides be produced, the exterior angle is greater than either of the interior and opposite angles." Euclid proves the claim for one of the interior angles and notes that the proof for the other is similar. The proof below is essentially his.

Theorem. *Let ABC be any triangle, and extend the segment BC to D. Then the external angle ACD is greater than interior angle BAC.*



Proof. Let E bisect segment AC, and extend the segment BE to F so that BE is congruent to EF. By side-angle-side, triangle ABE is congruent to CFE: AE is congruent to EC by the choice of E, BE is congruent to EF by construction, and angle AEB is congruent to angle CEF because they are vertical angles. As a result, angle BAC is congruent to the corresponding angle ACF. Since angle ACD is clearly greater than angle ACF, we have that ACD is greater than BAC, as required. □

The propositions of the *Elements* consist of *demonstrations*, which establish that a property holds of a given diagram, and *constructions*, which establish that one can construct geometric objects with specific properties with a ruler and compass. Each construction and inference in the proof above is justified explicitly by an appeal to a prior proposition or one of Euclid's axioms. There is one exception, namely, the inference marked "clearly." What justifies that?

Such diagrammatic inferences are ubiquitous in the *Elements*. Edward Dean, John Mumma, and I provide an analysis [11], building on Kenneth Manders' observation that diagrammatic inferences draw on topological relationships in the geometric data rather than exact metric relationships [48]. In this case, the construction guarantees that E and F are on the same side of line CD and that F and D are on the same side of line AC, which is what we mean when we say that F is inside angle ACD. This implies that angle ACF is smaller than angle ACD. We somehow track these relationships with some measure of reliability as we read the proof and verify its correctness. The goal of our analysis was not to explain how we do this; rather, it was to characterize the competences implicit in the practice. The point is that reading and writing Euclidean proofs require a particular geometric understanding, and we wanted to describe it. With such an analysis in hand, one can then address the cognitive question of how we can carry out such inferences (see, for example, [32]) or the question of how to train AI to write such proofs [50].

The fact that the formal axiomatic tradition fails to account for visual and diagrammatic reasoning has spawned a considerable philosophical literature (for overviews, see [26, 65]). In each use case, we can ask about the understanding a diagram conveys and the understanding required to use it effectively. There is nothing special about diagrams; the use of text-based representations in mathematics is equally complex and mysterious [10]. Questions about how we understand diagrammatic proofs and representations are, therefore, a part of the larger question of how we understand mathematical proofs and representations generally.

2.4 Historical progress

Mathematicians often claim that some mathematical advance was made possible by a prior conceptual development. For example, number theorists will tell you that Riemann's introduction of the complex zeta function and the use of complex analysis made it possible for Hadamard and de la Vallée Poussin to prove the prime number theorem in 1896. What is the sense of "possibility" here? Riemann wrote down the definition of the zeta function and derived some of its properties, which required considerable ingenuity, expertise, and insight. We need to make sense of a conceptual space in which writing a definition can be a substantial achievement, opening up new avenues of reasoning.

Manders was an advocate of the historical case studies as a means of philosophical analysis. Start with any episode in the history of mathematics that is commonly viewed as an important advance, like the invention of calculus or analytic geometry. Calling it an advance implies that something about our state of knowledge or understanding changed as a result. If we can figure out how to characterize the relevant change in epistemic state and why it was valuable, we will have learned something about what is important in mathematics. Historical analysis can thus contribute to our philosophical understanding even if we don't care about history per se.

When a mathematician finishes writing a proof, they often revise it substantially, breaking out definitions and lemmas, and simplifying and streamlining the argument. Software engineers call this *refactoring*. In the nineteenth century, Richard Dedekind published four versions of his proof of unique factorization of ideals in the ring of integers in any algebraic number field; considering what changed over the course of the development and why is illuminating [5]. Rebecca Morris and I studied the history of proofs of Dirichlet's celebrated theorem on primes in arithmetic progressions from Dirichlet's first publication in 1837 to a version by Landau in 1927, and we similarly found that the history taught us a lot about the nature of algebraic and set-theoretic abstraction [12].

Studying the history of mathematics can also raise new questions, such as the following. Mathematicians, philosophers, and historians often attribute the value of algebraic reasoning to its generality; for example, the axiomatization of *groups* in the nineteenth century unified instances in Galois theory, number theory, and geometry. However, although Dedekind and Weber generalized the notion of an ideal from rings of algebraic integers to rings of functions in 1882, Dedekind clearly felt that the abstraction provided a better understanding of the theory of algebraic integers even before the generalization. That raises the question of what it is about abstraction that contributes to understanding a particular body of results and, at the same time, enables generalization and application to other domains. Historical examples often provide insights that allow us to answer such questions [5, 8, 12].

2.5 Computers and understanding

Computer-assisted proofs have become increasingly common in contemporary mathematics, and there are several notable examples of results established only with the help of computation. Kenneth Appel and Wolfgang Haken used extensive computation to prove the four-color theorem in 1976 [3], and Thomas Hales announced a proof of the Kepler conjecture in 1998, again using extensive computation [27]. Propositional satisfiability solvers are being used to solve combinatorial problems, in some cases, producing proofs that are terabytes long, or even longer [36]. What are we to make of them? Thurston provides a succinct presentation of some of the key issues:

> The question is not even "How do mathematicians make progress in mathematics?" Rather, I prefer, "How do mathematicians advance human understanding of mathematics?" This question brings to the fore something that is fundamental and pervasive: that what we are doing is finding ways for people to understand and think about mathematics.
>
> The rapid advance of computers has helped dramatize this point, because computers and people are very different. For instance, when Appel and Haken completed a proof of the 4-color map theorem using a massive automatic computation, it evoked much controversy. I interpret the controversy as having little to do with doubt people had as to the veracity of the theorem or the correctness of the proof. Rather, it reflected a continuing desire for *human understanding* of a proof, in addition to knowledge that the theorem is true.
>
> On a more everyday level, it is common for people first starting to grapple with computers to make large-scale computations of things they might have done on a smaller scale by hand. They might print out a table of the first 10,000 primes, only to find that their printout isn't something they really wanted after all. They discover by this kind of experience that what they really want is usually not some collection of "answers"—what they want is understanding. [64]

We should ask about the extent to which computational methods contribute to mathematical understanding and the extent to which they fall short. The answers bear on how we use computers in mathematics and how we assess the results of computer-assisted proofs.

2.6 Mathematics education

As a society, we devote considerable resources to teaching mathematics to our children. From early childhood through high school, we subject students to countless hours of mathematics instruction and evaluate them with countless homework assignments, quizzes, and exams. If one goal of our educational practices is to convey mathematical understanding, we owe it to our students to think about what that means.

More than a quarter of a century ago, Liping Ma wrote a dissertation that was subsequently published under the title *Knowing and Teaching Elementary Mathematics: Teachers' Understanding of Fundamental Mathematics in China and the United States* [45]. Ma based her analysis on interviews in which she asked Chinese and American teachers to solve mathematical problems and explain how they would teach their students to do the same. The book argued that Chinese teachers had a deeper, more flexible understanding of elementary mathematics and sought to identify root causes. The study has been widely discussed and debated, and has influenced educational policy in the United States. The word "understanding" occurs hundreds of times in the text, often seven or eight times on a single page. The work distinguishes between different kinds of understanding, such as conceptual and procedural understanding. It also assesses the depth of understanding demonstrated in teacher interviews and recommends instructional procedures that are designed to convey understanding. Notably, the book does not attempt to give a general account of what it means to understand mathematics. We have to infer the meaning of the phrase from what Ma says about the case studies. This does not detract from the value of the work, but it makes it hard to assess the book's arguments rigorously and critically.

Contemporary philosophy of mathematics has only begun to address questions of mathematical understanding in education. Some educational researchers have begun to reflect on the nature of understanding (e.g. [23, 34, 72]); philosophers of mathematics have started publishing in the educational literature (e.g. [13, 29, 42]); and philosophers and educators have begun to collaborate (e.g. [62]). We will be limited in our ability to instill and assess students' mathematical understanding until we have a clearer sense of what it means to understand mathematics.

3 Toward a theory of mathematical understanding

The questions raised in Section 2 have several things in common. First, they all have an epistemological character; they concern our ability to discover, verify, and communicate mathematical knowledge. Second, they have a generally normative character: they are questions about how we *should* go about our mathematical activities. Third, they aren't very mysterious; if you ask a mathematician why a particular theorem or historical development is important, or ask a teacher how to convey understanding to students, they will have plenty to say. Finally, we care about the answers; we devote considerable resources to mathematical research and education, and the questions bear on the way we carry out and evaluate our work. These observations don't guarantee that there is a philosophical theory of mathematical understanding that can help us answer the questions, but they should at least encourage us to look.

3.1 The objects of understanding

The objects of mathematical understanding are varied. We can understand a definition, a theorem, or a proof. Conventional methods in mathematical logic represent these sorts of syntactic objects well. The same holds for problems, conjectures, examples, and theories; for instance, we can think of a theory as a collection of definitions, theorems, and proofs. It is not as straightforward to come up with a general definition of a diagram [26, 65], let alone explain what it means to understand one. We can often characterize the information conveyed by particular diagrams, such as Euclidean diagrams, knot diagrams, or diagrams in homological algebra. Still, it is less straightforward to make sense of the different ways diagrams may be presented to us and how that affects understanding.

We also speak of understanding more abstract entities such as methods, concepts, ideas, strategies, motivations, and goals. The philosophy of mathematics doesn't yet provide us with the means to discuss these with any rigor or precision. Sometimes, what we understand are processes rather than objects; we can understand how to prove a theorem or how to solve a problem. And just as we can understand all these sorts of things, they can, in turn, convey understanding. A diagram can help us understand a proof, and we can come to understand a definition by seeing the role it plays in proving a theorem.

3.2 Understanding as ability

The fact that we value mathematical understanding in so many different contexts suggests that we should not expect a simple, homogeneous account of what it means to understand mathematics. A common thread running through the examples above, however, is that understanding is closely tied to our mathematical *capacities*, that is, what we can do with our mathematical knowledge. Understanding a proof involves not only the ability to justify each inference but also the ability to discover the proof and others like it. Understanding Euclidean geometry includes the ability to read and write Euclidean proofs, which, in turn, relies on the ability to recognize legitimate inferences and reject illegitimate ones. Understanding Riemann's definition of the zeta function contributed to mathematicians' ability to prove the prime number theorem, and understanding elementary mathematics involves the ability to solve elementary mathematical problems across a range of variations.

These abilities are fundamentally capacities for *reasoning*. Whereas the conventional view sees mathematical knowledge as a body of definitions, theorems, and proofs, reasoning is a dynamic process of traversing a sequence of epistemic states to pursue a goal. We step through the inferences in a proof to verify the conclusion; we search for a proof to establish a new theorem; we manipulate mathematical representations of mathematical objects to solve problems; we look for patterns and formulate conjectures in the pursuit of new knowledge. Knowledge is both an outcome and a component of understanding, but at any given point in time, it is static; it is the network of definitions, theorems, and proofs we have at our disposal. Understanding is what enables us to use that knowledge effectively and to discover more.

Thinking first about what it means to reason will help us get a grip on understanding. Mathematics is hard; it calls on us to carry out complex tasks, like solving problems and proving theorems, that require long, precise sequences of steps. Each step we take changes our epistemic state: we introduce new objects, recall relevant definitions and facts, and draw conclusions. The action space is typically huge, and the combinatorial explosion of options is overwhelming. Understanding is what carries us through, taming complexity and guiding us toward our goals.

If this general view of understanding is correct, the challenge is to figure out how to talk about understanding and reasoning in a precise and rigorous way. I initially described the relevant capacities as *methods*, that is, quasi-heuristic, fallible reasoning procedures, perhaps described as a kind of cognitive-mathematical pseudocode [4]. However, thinking in procedural terms is too fine-grained. For example, to establish the correctness of a statement in number theory, it may be sufficient to recognize that the statement follows from a general result in algebra. In that case, the requisite understanding lies in the ability to identify the number-theoretic statement as an instance of the algebraic one, independent of the process that leads to that recognition. As a result, I later argued in favor of thinking about understanding in terms of the *abilities* to carry out the requisite reasoning steps [6, 7]. That will not get us entirely away from procedural thinking, since we often view some mathematical abilities as constitutive of others. For example, the ability to add multi-digit numbers presupposes the ability to add one-digit numbers and the ability to maintain place-value relationships. The natural way to explain that dependence is to view abilities as specifications of methods, or families of methods, that invoke other methods as subroutines.

Talking about procedures and abilities lands us in a terminological minefield. Cognitive scientists sometimes distinguish between ability-based accounts, which characterize cognition in terms of its behavioral manifestations, and representation-based accounts, which aim to get at the mental states that give rise to the behavior. What I propose is something more like the latter, with the caveat that we can't separate representations from the processes that act on them. Researchers in mathematics education often distinguish between students' *procedural knowledge*, by which they mean the ability of students to carry out calculations, and *conceptual knowledge*, by which they mean something more robust that can, for example, account for the ability to transfer procedural knowledge to novel settings. Once again, this distinction views abilities and procedural behavior as outward manifestations of understanding rather than explanations of how they arise. Here, I argue that, on the contrary, a suitable way of talking about abilities can provide rich functional models that explain the observable phenomena.

Although there are challenges to developing rigorous ways of talking about reasoning abilities, I am convinced it is the way to go. Others have also argued in favor of thinking about mathematical reasoning in procedural terms [24, 29, 61, 62], something that is clearly related to Ryle's distinction between knowing that and knowing how [54, 58]. Views of understanding in terms of abilities or intellectual know-how can also be found in the general epistemological literature [37] and in the epistemology of science [38]. Thinking about understanding in terms of abilities, however, is especially well-suited to mathematics and mathematical reasoning. In Section 3.3, I argue that such an approach is methodologically sound and that there are plenty of opportunities to make progress.

3.3 Methodology

We want to make sense of the various ways we talk about understanding in a manner that is clear, precise, and internally coherent, one that fits the available data and supports reasoning and debate about subjects that touch on the notion. I have argued that we should think of mathematical reasoning as a process of passage through epistemic states en route to a mathematical goal, and that we should think of mathematical understanding as possession of abilities that enable one to carry out such reasoning successfully. Some abilities are easy to characterize in behavioral terms, such as the ability to draw an appropriate inference or carry out a calculation. Others function at a higher level, for example, the ability to see an analogy between two mathematical domains or the ability to implement a high-level proof strategy. Sections 4 and 5 will explore some of the vocabulary available from computer science and cognitive science to talk about such abilities, but we should seek inspiration and guidance from all the domains of application indicated in Section 2.

Viewing understanding as possession of abilities accords well with our intuitions. If you tell me that a student in your class understands complex analysis and I ask you to explain what you mean, you will likely describe the kinds of problems the student can solve, the types of reasoning they can carry out, and the standard theorems that they can state and apply correctly. If you tell me that a particular historical result marked an advance in our understanding of a longstanding problem and I ask you what you mean, you will probably tell me what people have been able to do with it. There is no shortage of such data: we can study assessments of understanding several domains and look into the associated abilities. We can use that data to develop a general account of reasoning abilities and how they contribute to the observable results.

The key methodological claim of this section is that this is all a theory of mathematical understanding needs to do [6, 7]. In particular, we can set aside the question of how these abilities are implemented in human cognition, computer algorithms, neural networks, or historical and social processes. The latter are tasks for cognitive science, computer science, machine learning, and the history and sociology of mathematics. A theory of understanding that accounts for the historical and contemporary literature and that explains the practices and value judgments we see there can provide a philosophical foundation for all these domains of application.

To make progress, we should proceed from the bottom up, answering specific instances of the kinds of questions raised in Section 2 and seeing what the answers have in common. What are the diagrammatic inferences that are allowed in Euclid's *Elements*? How did the introduction of the zeta function lead to the solution of the prime number theorem? What kinds of understanding can we get from a brute-force computation? In what ways are the practices of Chinese mathematics instructors better at conveying mathematical understanding than those of their American counterparts? We can work towards an overarching theory by addressing such questions thoughtfully and then reflecting on what the answers tell us about mathematical understanding in general.

4 Insights from symbolic AI

Following the successes of mathematical logic and the theory of computation in the early twentieth century, and the advent of the digital computer in the 1940s, a logical, symbolic, rule-based view of reasoning became the dominant paradigm in artificial intelligence. In computer science, such a perspective is fundamental to formal methods and automated reasoning, which are used, among other things, to verify complex systems. The associated views of language and thought also shaped research in linguistics, cognitive science, philosophy of language, and philosophy of mind. In this section, we consider how the symbolic perspective can help us think about mathematical understanding.

4.1 Concepts

Talk of concepts is ubiquitous in mathematics, philosophy of mathematics, and mathematics education. We can talk about the concept of a group, the concept of a Riemannian manifold, the evolution of the function concept, or children's concept of number. Ordinary concepts like "dog" or "tree" are graded, meaning that some instances are more typical than others, and the psychological literature explains this with theories of prototypes and exemplars. However those theories don't characterize mathematical concepts well. A mathematical object is either a group or it isn't, and though some examples of groups are more common or straightforward, we would not characterize them as more group-like. To a logician, a concept is a predicate, defined in a formal language with precise semantics, but that doesn't capture the way we talk about mathematical concepts either. A student can master the concept of a cardinal number without knowing the set-theoretic definition of a number, and understanding the group concept is not the same as knowing the definition of a group.

Several additional features of mathematical concepts make the notion hard to pin down. Although falling under a concept doesn't admit degrees, the understanding of a concept does. Talk of concepts is also context-dependent; for example, a student's understanding of the group concept differs from that of an expert. We don't acquire a better understanding of the number concept by seeing more numbers, nor do we acquire a better understanding of the group concept solely by seeing more groups. Algebraic and analytic number theory deepens our understanding of the integers by embedding them in larger structures that, a priori, have nothing to do with counting. We can also talk about the collective understanding of a concept within the mathematical research community, and we can talk about the evolution of a mathematical concept over time. To think about mathematics and mathematical understanding, we need a notion of concept that is compatible with all these ways of talking about them.

The dynamical perspective presented in Section 3 offers a promising way forward [4]. In ordinary circumstances, if I tell you that someone understands the group concept and you ask me to expand on what I mean, I will likely expand on the abilities that I take the ascription to entail: the ability to state the definition of a group, come up with or recognize instances of groups, distinguish between different types of groups, apply theorems about groups, and so on. If I tell you that an elementary school student has a solid understanding of the concept of a cardinal number and you ask me what I mean, I am likely to expand on what the student can do. If I try to explain how a body of historical results advanced our understanding of groups, I am apt to do something similar, namely, explain what the advance enabled the research community to do. This suggests that, in each case, when we talk about understanding a concept, we have a bundle of abilities in mind. We can therefore clarify the meaning of a phrase "understanding concept X" by specifying the relevant abilities.

This view makes the concepts of a group and a cardinal number vague, context-dependent, and open-ended, but that is the point: our talk of concepts in informal discourse is vague, context-dependent, and open-ended in just this way. The philosophy of mathematics can help us talk about concepts more precisely by helping us be more rigorous in talking about the attendant abilities.

4.2 Representations

In cognitive science and philosophy of mind, one often explains problem-solving behavior in terms of mental *representations*, each with its own *affordances*, or ways of being manipulated and transformed, that make it suitable for the relevant problem-solving task. For example, a diagram may be well-suited for tracking geometric relationships, while an algebraic expression may be better suited for calculations. We often attribute the ability to solve a problem or carry out a task to having the right representations.

In the philosophy of mathematics, instead of thinking about mental representations, it is more common to focus on external representations, such as written expressions and diagrams, or on abstract mathematical representations, like the representation of group as a set of symmetries, the representation of a partial order as a system of sets under the inclusion relation, or the representation of a graph by its incidence matrix. In Section 2.3 considered the role of diagrammatic representations in Euclidean geometry [48], and de Toffoli and Giardino have considered representations used in knot theory [19]. These works, and others (e.g. [26, 63, 65]), tend to focus on visual and diagrammatic representations, but often touch on other representational forms, such as algebraic expressions and symbolic notations. Manders prefers the word "artifacts" over "representations" to focus attention on the role they play in mathematical reasoning rather than their semantics. Waszek [71] favors the word "sign" for similar reasons and makes a case that they should be taken more seriously in the philosophy of mathematics.

4.3 Modularity and abstraction

Mathematicians know that one way to make a complex proof intelligible is to break it into small, manageable lemmas, and that the best way to develop a complex theory is to introduce networks of definitions and theorems that can be read and understood independently. Computer scientists and software engineers know that a complex system is easier to understand, develop, and maintain if it is similarly decomposable into smaller, self-contained parts that can be understood, developed, and maintained independently. Such systems are said to be *modular*, which is to say, they can be broken down into smaller *modules*, each of which *encapsulates* its implementation behind a well-defined *interface*. That encapsulation is a form of *abstraction*, in that it characterizes general properties and behaviors of interest, independent of particular instantiations.

The notion of modularity is used to make sense of complex systems across a range of natural and applied sciences. The modular design of an engineered system is intended to make it easier to understand, develop, maintain, and reuse. The modular design of mathematical artifacts can, similarly, explain how they acquire these virtues [8, 9]. A recent essay by Heather Macbeth [46] explores how computational proof assistants support new forms of abstraction in mathematical reasoning.

Modularity and abstraction provide means of managing complexity by limiting the amount of information at play at any point in a reasoning task. Imagine a worker assembling a complex machine, with tools and supplies on the table in front of them, more on nearby shelves, and others available from a warehouse. A modular design allows the worker to assemble the machine in parts, fetching components as needed and moving objects on and off the table to keep the workspace manageable. This metaphor supports several disciplinary approaches to thinking about reasoning and computation. Cognitive scientists distinguish between *short-term memory*, also known as *active memory*, and *long-term memory*. In the context of problem-solving or reasoning tasks, the former is sometimes identified as *working memory*. In a computer, accessing CPU registers is much faster than accessing main memory, which is much faster than accessing data on disk; compiler design focuses on efficient use of registers, and system design focuses on efficient use of memory hierarchies. In an interactive theorem prover, the *local context* contains facts and hypotheses that are immediately available for inference, in contrast to the larger *environment*, which contains definitions and theorems that can be accessed when needed. In automated reasoning, first-order theorem provers use heuristics to transfer facts from a *passive* database of assumptions to an *active* set of facts currently used in the proof search. Understanding how mathematics enables us to move information in and out of a local reasoning context is an important part of understanding how mathematical reasoning is implemented in human and computational systems.

4.4 Cognitive difficulty

Our discussion of mathematical understanding is framed by the recognition that mathematics is hard. As cognitive agents, we have limited time, energy, memory, and processing capacity. We can only keep so much information in mind at once, we can only carry out so many inferences in a given amount of time, and we can only carry out so many reasoning steps before we lose track of where we are. We therefore value mathematical methods and artifacts that are cognitively efficient, in other words, that make the best use of our limited cognitive resources. We value methods that are easy to apply, definitions that are easy to use, and proofs that are easy to follow.

Coming to terms with mathematical understanding, therefore, requires us to come to terms with mathematical ease and difficulty [8, 9, 71]. Once again, the various disciplines that draw on notions of mathematical understanding differ in how they think about the complexity of mathematical tasks. In experimental psychology, we can measure the difficulty of a task by timing how long it takes to complete it or by counting the number of errors made. In logic, we can measure the complexity of a proof by counting the number of steps or symbols. Computer science gives us complexity classes. Note, however, that complexity classes measure the complexity of a family of problems, whereas, in assessing the value of a historical development, we are often interested in measuring the role of a new method or insight in simplifying a specific problem or proof.

To some extent, we can abstract a measure of complexity from any particular system implementation. Bubble sort requires $O(n^2)$ steps whether it is carried out by a computer or by a human. When we talk about mathematics, however, we don't descend to the level of a specific programming model or formal proof system, nor do we expect our analysis to be cast in terms of psychological experiments. We need better ways of talking about cognitive difficulty in mathematics, and the strategies and methods we use to manage complexity and transcend cognitive limitations.

5 Insights from machine learning

Both logic-based and machine-learning approaches to AI were represented at the 1956 Dartmouth workshop that gave the field its name. Initially, symbolic approaches dominated, but machine learning became more prominent in the 1990s with the rise of big data, and the twenty-first century brought dramatic advances in deep learning, large language models, and reinforcement learning. The symbolic and neural traditions embody different perspectives on reasoning and understanding. Whereas symbolic AI views reasoning as a matter of performing logical inferences, neural AI sees it as a matter of drawing inferences from experience and data. Whereas symbolic AI uses hand-crafted, explicit rules and representations, neural AI uses implicit rules and representations distributed across the parameters of a neural network. Remarkably, despite these differences, machine learning approaches to mechanizing mathematics draw on many of the same paradigms as symbolic approaches.

Machine learning is not yet commonly used in mathematical research, but it is making inroads, with several notable successes in recent years. Neural networks can be trained on vast amounts of computational data, enabling them to detect subtle patterns and relationships [16, 35] or make highly educated guesses of symbolic expressions that can be checked independently to satisfy computational constraints [41, 2]. They can be used to compute solutions to partial differential equations efficiently and find regions of interest [69, 70]. Systems based on reinforcement learning can be trained to construct mathematical objects, learning from their successes and failures [66, 25]. Large language models can be trained on the mathematical literature to synthesize answers to mathematical questions and write mathematical programs and proofs. The annual International Mathematics Olympiad (IMO) tests the most talented and capable high school students in the world on their ability to solve novel problems, requiring creative thought and ingenuity; neural systems can now perform at the level of gold medalists, producing either formal proofs that can be checked by a computational proof assistant [20, 1, 14] or informal proofs that can be checked by ordinary referees [73, 44]. As I write, the state of the art of neural and neurosymbolic theorem proving has been advancing at a dizzying pace; systems are now producing new mathematical results both informally and formally, with only modest user interaction [55]. It's hard to predict what such systems will be capable of doing even a few months from now.

5.1 The role of experience

Two decades ago, in "Mathematical method and proof," I wrote:

> At one end of the spectrum of mathematical activity, there is routine calculation and verification, where the appropriate means of proceeding is clear and straightforward; at the other, there is blind search and divine inspiration. Mathematical methods are designed to shift as much as possible to the first side, so that serious thought and hard work can be reserved for tasks that are truly difficult. The project proposed here is to better understand how they do this. [4]

At the time, I did not see how to say anything substantial about mathematical intuition and insight, so I focused on more straightforward components of mathematical reasoning and on how mathematical methods, representations, and heuristics guide us. In other words, I paid attention to the sorts of things that symbolic methods can account for.

The problem with symbolic search is that it gets lost all too quickly. Most approaches to automating first-order reasoning search for proofs in restricted calculi, such as resolution or tableaux systems, which are known to be inefficient. Typical "speedup" results show that proofs can be much shorter when they invoke auxiliary definitions and lemmas, but symbolic methods are ill-equipped to find them [53]. In the article mentioned above, I recognized this limitation of the approach and acknowledged that my focus on the symbolic "may simply miss the conceptual forest for the syntactic trees."

Machine learning provides a view of the conceptual forest. If one asks ChatGPT or Google Gemini for help in proving a theorem, one often gets answers that are not quite correct but nonetheless helpful. AI is good at suggesting promising directions and strategies, even when it is not as good at carrying out the details correctly. This suggests a promising approach to automating mathematical reasoning: use machine learning to get close to a correct result and then use symbolic methods to finish the job. A paper called "Draft, sketch, and prove: guiding formal theorem provers with informal proofs" [39] represented an early implementation of such a strategy, using a large language model to sketch proofs in a formal proof assistant and symbolic automation to fill in the details. Another early study [33], which trained a model to produce formal proofs in the Lean proof assistant, found that pretraining on a large corpus of informal data scraped from the internet improved the model's ability to construct formal proofs. Results like these underscore that even rule-based tasks often require the kind of insight and understanding that come from a broader range of experiences.

Thus, where symbolic provers fall to combinatorial explosion, neural methods offer new ways forward. Research on machine learning for mathematics involves crafting learning experiences rather than algorithms and rules, and understanding is seen as the result of appropriate training rather than the product of suitable symbolic algorithms. Reconciling these views poses a philosophical challenge: we need to understand how our precise symbolic rules and representations emerge from our individual and collective experiences, and even, possibly, how our symbolic methods are justified by experience. In machine learning, one asks questions such as: What kinds of neural and system architectures give rise to expert behavior? What sorts of training regimes are useful? What opportunities do systems have for learning and exploration? We can ask analogous questions about human agents and communities. Although the answers to the two types of questions will differ in the details, the commonalities will enable us to hone in on the essential features of mathematical thought, rather than artifacts of its various implementations.

5.2 Neural perspectives on reasoning

Even when computer scientists work on purely neural approaches to perform mathematical reasoning and proof, the paradigm of reasoning as step-wise inference, as described in Section 3.2, still guides their work. Around 2022, researchers made significant progress on getting AI models to solve mathematical problems with the realization that performance improves dramatically when models are prompted to reason step by step, resulting in what is now known as *chain-of-thought* (CoT) reasoning [74, 68]. Even something as simple as prompting a model with the instruction "let's think step by step" [40] or "take a deep breath and work on this problem step by step" [76] makes a difference. OpenAI's GPT o1, released in 2024, famously inaugurated a new generation of models that used chain of thought to obtain better performance on reasoning tasks [51]. A few months later, DeepSeek released an open-source model, DeepSeek R1, which had similar performance [21].

DeepMind's *AlphaProof*, the first prover to succeed at solving IMO problems, used reinforcement learning to train a model to construct formal proofs in the Lean proof assistant. The value of the proof assistant is that it provides a clear signal when a proving task has been carried out correctly, allowing a system to explore and learn without human judgment or intervention. Other provers have achieved positive results by combining symbolic theorem proving with chain-of-thought reasoning in natural language [67, 75, 14, 43]. Such provers unify the view of reasoning as a symbolic, rule-based process with that of reasoning as a process guided by experience and training.

Neural proving is guided by further analogies to human learning and reasoning. *Curriculum learning* is a method of training systems by proceeding from easy problems to increasingly difficult ones. DeepMind's AlphaProof was trained in this fashion, using one model to generate problems while another one learned to solve them [44]. The approach even extends to test-time performance: given a particular problem to solve, a system often does better by first guessing easier versions of the problem and solving those. The technical term for guessing is *conjecturing*, and it is now commonly accepted that the ability to conjecture results is an integral part of both training and test-time performance [22, 14].

6 Conclusions

As mathematical reasoning becomes increasingly mechanized, we need to address the question as to what is distinctly important about *our* understanding. Technological advances compel us to think about our role in the mathematical process, to determine why our mathematical understanding matters, and to decide how we ought to engage with the emerging technologies. Coming to terms with our mathematical values requires us to reflect on the nature and purpose of mathematics itself. For better or worse, the new developments are bound to disrupt mathematical practice. From ancient Greece to the scientific revolution and the foundational crises of the early twentieth century, philosophy of mathematics has been there to guide us; once again, it has a vital role to play in helping us navigate the changes ahead.

We have seen that AI offers two broad frameworks for situating a theory of understanding. In logic-based, symbolic approaches, the goal is to understand the symbolic representations and processes that give rise to understanding, and the way mathematical knowledge is structured to support them. In machine learning and neural approaches, the goal is to understand the experiences and training regimes that give rise to understanding, and the way mathematical knowledge is structured to support learning from experience and exploration. Reconciling these two ways of thinking about understanding will have both philosophical and practical benefits.

References

[1] Tudor Achim, Alex Best, Alberto Bietti, Kevin Der, Mathïs Fédérico, Sergei Gukov, Daniel Halpern-Leistner, Kirsten Henningsgard, Yury Kudryashov, Alexander Meiburg, Martin Michelsen, Riley Patterson, Eric Rodriguez, Laura Scharff, Vikram Shanker, Vladmir Sicca, Hari Sowrirajan, Aidan Swope, Matyas Tamas, Vlad Tenev, Jonathan Thomm, Harold Williams, and Lawrence Wu. Aristotle: IMO-level automated theorem proving, 2025. arXiv: 2510.01346.

[2] Alberto Alfarano, François Charton, and Amaury Hayat. Global Lyapunov functions: a long-standing open problem in mathematics, with symbolic transformers. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Neural Information Processing Systems (NeurIPS)*, 2024.

[3] Kenneth Appel and Wolfgang Haken. Every planar map is four colorable. *Bulletin of the American Mathematical Society*, 82:711–712, 1976.

[4] Jeremy Avigad. Mathematical method and proof. *Synthese*, 153:105–159, 2006.

[5] Jeremy Avigad. Methodology and metaphysics in the development of Dedekind's theory of ideals. In *The Architecture of Modern Mathematics*, pages 159–186. Oxford University Press, Oxford, 2006.

[6] Jeremy Avigad. Understanding proofs. In Paolo Mancosu, editor, *The Philosophy of Mathematical Practice*, pages 317–353. Oxford University Press, Oxford, 2008.

[7] Jeremy Avigad. Understanding, formal verification, and the philosophy of mathematics. *Journal of the Indian Council of Philosophical Research*, 27:161–197, 2010.

[8] Jeremy Avigad. Modularity in mathematics. *The Review of Symbolic Logic*, 13:47–79, 2020.

[9] Jeremy Avigad. Reliability of mathematical inference. *Synthese*, 198:7377–7399, 2021.

[10] Jeremy Avigad. The design of mathematical language. In Bharath Sriraman, editor, *Handbook of the History and Philosophy of Mathematical Practice*, pages 3151–3189. Springer, Cham, 2024.

[11] Jeremy Avigad, Edward Dean, and John Mumma. A formal system for Euclid's *Elements*. *Review of Symbolic Logic*, 2:700–768, 2009.

[12] Jeremy Avigad and Rebecca Morris. Character and object. *The Review of Symbolic Logic*, 9:480–510, 2016.

[13] Jessica Carter. Mathematical understanding – common themes in philosophy and mathematics education. *The Journal of Mathematical Behavior*, 76:101202, 2024.

[14] Luoxin Chen, Jinming Gu, Liankai Huang, Wenhao Huang, Zhicheng Jiang, Allan Jie, Xiaoran Jin, Xing Jin, Chenggang Li, Kaijing Ma, Cheng Ren, Jiawei Shen, Wenlei Shi, Tong Sun, He Sun, Jiahui Wang, Siran Wang, Zhihong Wang, Chenrui Wei, Shufa Wei, Yonghui Wu, Yuchen Wu, Yihang Xia, Huajian Xin, Fan Yang, Huaiyuan Ying, Hongyi Yuan, Zheng Yuan, Tianyang Zhan, Chi Zhang, Yue Zhang, Ge Zhang, Tianyun Zhao, Jianqiu Zhao, Yichi Zhou, and Thomas Hanwen Zhu. Seed-prover: Deep and broad reasoning for automated theorem proving, 2025. arXiv:2507.23726.

[15] John M. Cooper, editor. *Plato: Complete Works*. Hackett Publishing Company, Indianapolis, 1997.

[16] Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis, and Pushmeet Kohli. Advancing mathematics by guiding human intuition with AI. *Nature*, 600:70–74, 2021.

[17] John W. Dawson, Jr. Why do mathematicians re-prove theorems? *Philosophia Mathematica*, 14:269–286, 2006.

[18] John W. Dawson, Jr. *Why prove it again?* Springer, Cham, 2015.

[19] Silvia De Toffoli and Valeria Giardino. An inquiry into the practice of proving in low-dimensional topology. In Gabriele Lolli, Marco Panza, and Giorgio Venturi, editors, *From Logic to Practice: Italian Studies in the Philosophy of Mathematics*, pages 315–336. Springer, Cham, 2015.

[20] DeepMind. AI achieves silver-medal standard solving International Mathematical Olympiad problems: AlphaProof and AlphaGeometry 2. https://deepmind.google/blog/ai-solves-imo-problems-at-silver-medal-level/, 2024.

[21] DeepSeek AI. Deepseek R1, 2025. https://github.com/deepseek-ai/DeepSeek-R1.

[22] Kefan Dong and Tengyu Ma. STP: self-play LLM theorem provers with iterative conjecturing and proving. In *International Conference on Machine Learning (ICML)*. OpenReview.net, 2025.

[23] Elizabeth Fennema and Thomas A. Romberg, editors. *Mathematics Classrooms That Promote Understanding*. Lawrence Erlbaum Associates, Mahwah, NJ, 1999.

[24] Janet Folina. Towards a better understanding of mathematical understanding. In Mario Piazza and Gabriele Pulcini, editors, *Truth, Existence and Explanation*, pages 121–146. Springer, Cham, 2018.

[25] Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao, and Adam Zsolt Wagner. Mathematical exploration and discovery at scale, 2025. arXiv:2511.02864.

[26] Marcus Giaquinto. *Visual Thinking in Mathematics: An Epistemological Study*. Oxford University Press, Oxford, 2007.

[27] Thomas C. Hales. A proof of the Kepler conjecture. *Annals of Mathematics*, 162:1065–1185, 2005.

[28] Yacin Hamami. Mathematical rigor and proof. *The Review of Symbolic Logic*, 15:409–449, 2022.

[29] Yacin Hamami and Rebecca Lea Morris. Philosophy of mathematical practice: a primer for mathematics educators. *ZDM Mathematics Education*, 52:1113–1126, 2020.

[30] Yacin Hamami and Rebecca Lea Morris. Plans and planning in mathematical proofs. *The Review of Symbolic Logic*, 14:1030–1065, 2021.

[31] Yacin Hamami and Rebecca Lea Morris. Understanding in mathematics: the case of mathematical proofs. *Noûs*, 58:1073–1106, 2024.

[32] Yacin Hamami, John Mumma, and Marie Amalric. Counterexample search in diagram-based geometric reasoning. *Cognitive Science*, 45:1–25, 2021.

[33] Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward W. Ayers, and Stanislas Polu. Proof artifact co-training for theorem proving with language models. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2022.

[34] Guershon Harel. Students' understanding of proofs: a historical analysis and implications for the teaching of geometry and linear algebra. *Linear Algebra and its Applications*, 302/303:601–613, 1999.

[35] Yang-Hui He, Kyu-Hwan Lee, Thomas Oliver, and Alexey Pozdnyakov. Murmurations of elliptic curves. *Experimental Mathematics*, 34:528–540, 2025.

[36] Marijn J. H. Heule and Oliver Kullmann. The science of brute force. *Communications of the ACM*, 60:70–79, 2017.

[37] Allison Hill. Understanding why. *Noûs*, 50, 2016.

[38] Anna Elisabeth Höhl. *Scientific Understanding: What It Is and How It Is Achieved*. Transcript Verlag, Bielefeld, Germany, 2024.

[39] Albert Qiaochu Jiang, Sean Welleck, Jin Peng Zhou, Timothée Lacroix, Jiacheng Liu, Wenda Li, Mateja Jamnik, Guillaume Lample, and Yuhuai Wu. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2023.

[40] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Neural Information Processing Systems (NeurIPS)*, 2022.

[41] Guillaume Lample and François Charton. Deep learning for symbolic mathematics. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2020.

[42] Marc Lange. What are explanatory proofs in mathematics and how can they contribute to teaching and learning? *The Journal of Mathematical Behavior*, 76:101191, 2024.

[43] Yong Lin, Shange Tang, Bohan Lyu, Ziran Yang, Jui-Hui Chung, Haoyu Zhao, Lai Jiang, Yihan Geng, Jiawei Ge, Jingruo Sun, Jiayun Wu, Jiri Gesi, Ximing Lu, David Acuna, Kaiyu Yang, Hongzhou Lin, Yejin Choi, Danqi Chen, Sanjeev Arora, and Chi Jin. Goedel-prover-v2: Scaling formal theorem proving with scaffolded data synthesis and self-correction, 2025. arXiv:2508.03613.

[44] Thang Luong and Edward Lockhart. Advanced version of Gemini with Deep Think officially achieves gold-medal standard at the international mathematical olympiad. https://deepmind.google/en/blog/advanced-version-of-gemini-with-deep-think-officially-achieves-gold-medal-standard-at-the-international-mathematical-olympiad/.

[45] Liping Ma. *Knowing and Teaching Elementary Mathematics: Teachers' Understanding of Fundamental Mathematics in China and the United States*. Routledge, New York, 1999.

[46] Heather Macbeth. Algorithm and abstraction in formal mathematics. In Kevin Buzzard, Alicia Dickenstein, Bettina Eick, Anton Leykin, and Yue Ren, editors, *Mathematical Software – ICMS 2024*, pages 12–25, Cham, 2024. Springer Nature Switzerland.

[47] Paolo Mancosu, Francesca Poggiolesi, and Christopher Pincock. Mathematical Explanation. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2023 edition.

[48] Kenneth Manders. The Euclidean diagram. In Paolo Mancosu, editor, *The Philosophy of Mathematical Practice*, pages 80–133. Oxford University Press, Oxford, 2008.

[49] Rebecca Lea Morris. Motivated proofs: what they are, why they matter and how to write them. *The Review of Symbolic Logic*, 13:23–46, 2020.

[50] Logan Murphy, Kaiyu Yang, Jialiang Sun, Zhaoyu Li, Anima Anandkumar, and Xujie Si. Autoformalizing euclidean geometry. In *International Conference on Machine Learning (ICML)*. OpenReview.net, 2024.

[51] OpenAI. Learning to reason with LLMs. `https://openai.com/index/learning-to-reason-with-llms/`, 2024.

[52] Henri Poincaré. *Science et Méthode*. Flammarion, Paris, 1908. Translated by Francis Maitland as *Science and method*, Dover Publications, New York, 1952.

[53] Pavel Pudlák. The lengths of proofs. In *Handbook of Proof Theory*, pages 547–637. North-Holland, Amsterdam, 1998.

[54] Gilbert Ryle. *The Concept of Mind*. University of Chicago Press, Chicago, 1949.

[55] Johannes Schmitt. Extremal descendant integrals on moduli spaces of curves: An inequality discovered and proved in collaboration with ai, 2025. `arXiv:2512.14575`.

[56] Stewart Shapiro, editor. *The Oxford Handbook of Philosophy of Mathematics and Logic*. Oxford University Press, Oxford, New York, 2005.

[57] George F. Simmons. *Precalculus Mathematics in a Nutshell: Geometry, Algebra, Trigonometry*. Janson Publications, 1987.

[58] Jason Stanley. Knowing (how). *Noûs*, 45:207–238, 2011.

[59] Fenner Stanley Tanswell. *Mathematical Rigour and Informal Proof*. Cambridge University Press, 2024.

[60] Mark Steiner. Mathematical Explanation. *Philosophical Studies*, 34:133–151, 1978.

[61] Fenner Stanley Tanswell and Matthew Inglis. The language of proofs: a philosophical corpus linguistics study of instructions and imperatives in mathematical texts. In Bharath Sriraman, editor, *Handbook of the History and Philosophy of Mathematical Practice*, pages 2925–2952. Springer, Cham, 2024.

[62] Fenner Stanley Tanswell and Keith Weber. Instructions and recipes in mathematical proofs. *Educational Studies in Mathematics*, 111, 2022.

[63] Jamie Tappenden. Proof style and understanding in mathematics I: Visualization, unification and axiom choice. In Paolo Mancosu, Klaus Frovin Jørgensen, and Stig Andur Pedersen, editors, *Visualization, Explanation and Reasoning Styles in Mathematics*, pages 147–214. Springer Netherlands, Dordrecht, 2005.

[64] William P. Thurston. On proof and progress in mathematics. *Bulletin of the American Mathematical Society*, 30:161–177, 1994.

[65] Silvia De Toffoli. *How to Prove Things with Diagrams*. Oxford University Press, Oxford, to appear.

[66] Adam Zsolt Wagner. Constructions in combinatorics via neural networks, 2021. arXiv:2104.14516.

[67] Haiming Wang, Mert Unsal, Xiaohan Lin, Mantas Baksys, Junqi Liu, Marco Dos Santos, Flood Sung, Marina Vinyes, Zhenzhe Ying, Zekai Zhu, Jianqiao Lu, Hugues de Saxcé, Bolton Bailey, Chendong Song, Chenjun Xiao, Dehao Zhang, Ebony Zhang, Frederick Pu, Han Zhu, Jiawei Liu, Jonas Bayer, Julien Michel, Longhui Yu, Léo Dreyfus-Schmidt, Lewis Tunstall, Luigi Pagani, Moreira Machado, Pauline Bourigault, Ran Wang, Stanislas Polu, Thibaut Barroyer, Wen-Ding Li, Yazhe Niu, Yann Fleureau, Yangyang Hu, Zhouliang Yu, Zihan Wang, Zhilin Yang, Zhengying Liu, and Jia Li. Kimina-prover preview: Towards large formal reasoning models with reinforcement learning, 2025. arXiv:2504.11354.

[68] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2023.

[69] Y. Wang, C.-Y. Lai, J. Gómez-Serrano, and T. Buckmaster. Asymptotic self-similar blow-up profile for three-dimensional axisymmetric Euler equations using neural networks. *Physical Review Letters*, 130:Paper No. 244002, 6, 2023.

[70] Yongji Wang, Mehdi Bennani, James Martens, Sébastien Racanière, Sam Blackwell, Alex Matthews, Stanislav Nikolov, Gonzalo Cao-Labora, Daniel S. Park, Martin Arjovsky, Daniel Worrall, Chongli Qin, Ferran Alet, Borislav Kozlovskii, Nenad Tomašev, Alex Davies, Pushmeet Kohli, Tristan Buckmaster, Bogdan Georgiev, Javier Gómez-Serrano, Ray Jiang, and Ching-Yao Lai. Discovery of unstable singularities, 2025. arXiv:2509.14185.

[71] David Waszek. Signs as a theme in the philosophy of mathematical practice. In Bharath Sriraman, editor, *Handbook of the History and Philosophy of Mathematical Practice*, pages 2971–3001. Springer, Cham, 2024.

[72] Keith Weber and Matthew Inglis. Mathematics education research on mathematical practice. In Bharath Sriraman, editor, *Handbook of the History and Philosophy of Mathematical Practice*, pages 2637–2663. Springer, Cham, 2024.

[73] Alexander Wei. Our latest experimental reasoning model has achieved a longstanding grand challenge in AI: gold-medal level performance on the world's most prestigious math competition — the International Mathematical Olympiad. https://www.scientificamerican.com/article/openai-model-earns-gold-medal-score-at-international-math-olympiad-and/, July 2025.

[74] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Neural Information Processing (NeurIPS)*, 2022.

[75] Huajian Xin, Z. Z. Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, Wenjun Gao, Haowei Zhang, Qihao Zhu, Dejian Yang, Zhibin Gou, Z. F. Wu, Fuli Luo, and Chong Ruan. Deepseek-prover-v1.5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2025.

[76] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2024.