The Byzantine Computer Scientists Problem (DRAFT)

Eric Purdy

November 9, 2023

Abstract

What would it take to move beyond the P vs. NP problem? When can computer scientists stop receiving lengthy proofs that P = NP, or P != NP? One approach would be to prove that neither P = NP nor P != NP can be proven with certainty in polynomial time by computationally bounded agents; then we could all collectively go about our lives, achieving a Pareto-optimal outcome.

Unfortunately, verifying the proof of such a result creates a massive set of coordination problems that we term the Byzantine Computer Scientists Problem. Specifically, it is not in the self-interest of any single computer scientist to engage with a proof on the topic of P vs. NP. The only way such a proof could be created and verified without any single computer scientist acting against their own interests, would be for a single computer scientist to formulate a correct and almost entirely complete proof, and then establish a verification protocol with trusted confederates over public channels, and then coax a sufficiently large coalition of eminent computer scientists to engage with the verification protocol. Ultimately, if the coalition was large enough and/or eminent enough, we prove that the theorem can be verified to any desired level of certainty by expending polynomially many computational resources.

The primary cause of the coordination problems is that the relevant game theoretic structure is a massive multiplayer game of imperfect information. Since information is imperfect, many important and decision-relevant quantities are unobserved. Since computational resources are bounded, efficient heuristics must be developed to allow these quantities to be estimated to the necessary level of precision to allow coordination. We provide efficient, well-validated heuristics for the necessary inference tasks in this paper, based on the Elo and Glicko systems for chess ratings, and based on the PageRank algorithm for determining document importance in a web of linked documents.

We next discuss the cut method for proving the limits of computational paradigms. We prove the limits of an extremely simple computational model, that of weighing coins with a balance.

We next discuss the known literature in deep reinforcement learning, showing that the cut method cannot easily be applied to continuously learning Transformer-based neural networks. This is because of the extremely non-local nature of the attention pattern in a Transformer, and because the continuous learning makes the resulting graph that one would have to cut obscenely huge.

We then survey the available literature on the phenomenon of "grokking" in transformer models solving algorithmic tasks. We thus conclude that it is plausible that Transformer-based neural networks have no particular limits, if one is willing to train them for infeasibly long periods.

We then discuss the computational alignment problem: how do we allow distributed computation of the answers to potentially difficult problems, without also allowing nefarious uses of that distributed computation? We provide theoretical results that, given certain reasonable assumptions, we can probably be approximately safe, i.e., that we can make the probability of large-impact events quite small.

We then attempt to validate these assumptions by studying various subproblems of the computational alignment problem. Specifically, we provide a proposed scalable solution to the problems of eliciting latent knowledge and performing mechanistic interpretability in large language models. We also provide a variant of model-based reinforcement learning with provable safety guarantees. We provide an algorithm for learnable heuristic caching for real-time intelligence tasks based on the mammalian cerebellum, capable of implementing one of the core components of the safe model-based reinforcement learning algorithm.

We conclude with a list of open questions related to the structure and function of the components of the brain in humans, mammals, and other animals.

1 Note to the Reader

This document is a work in progress. Significant work remains to be done.

This document assumes significant amounts of familiarity with the existing literature in three fields: complexity theory, deep reinforcement learning, and the alignment problem. We would have to write a whole large textbook to make this document self-contained, and we have not had time to do so. We apologize to the reader for this situation.

For an introduction to complexity theory, we recommend Computational Complexity: A Modern Approach, by Sanjeev Arora and Boaz Barak. It is available online at https://theory.cs.princeton.edu/complexity/book.pdf

For an introduction to deep reinforcement learning, we suggest that the reader consult the first google hit for "deep rl textbook", which is Deep Reinforcement Learning, a textbook, by Aske Plaat, available online at https://arxiv.org/abs/2201.02135

For an introduction to the alignment problem, we suggest that the reader consult the first google hit for "ai alignment textbook", which is The Alignment Problem, by Brian Christian, available for purchase online at https://brianchristian.org/the-alignment-problem/

Alternatively, the reader may consult the first google hit for "ai safety fundamentals", which is the following website: https://aisafetyfundamentals.com/

This document also assumes familiarity with my previously published work in complexity theory. This includes:

1. Lower Bounds for Coin-Weighing Problems, published in ACM TOCT 2011, and available for rental or purchase on DeepDyve.

2. Locally Expanding Hypergraphs and the Unique Games Conjecture, a masters thesis in computer science accepted in 2008 at the University of Chicago. I had it up on my website (ericpurdy.com) forever, presumably it's on the internet archive somewhere.

2 Introduction

Many have tried to solve the problem of P vs. NP. All have failed. The definition of insanity, according to Einstein, is doing the same thing over and over and expecting different results. Let us stop being insane, then, and try something slightly different.

We propose a new class of attacks on the P vs. NP problem, that use a combination of theoretical mathematics and concrete experimentation. We will no longer try to anticipate the resolution to the question before beginning the attack, since knowing the answer is above any one person's capabilities. We will operate solely within constructivist mathematics, since we would like to get our shiny new algorithm if it turns out that P = NP.

We propose a research program in which theoretical, mathematical argumentation is combined with pragmatic, programmatic experimentation. If P = NP, and the runtime of the most efficient algorithm is small enough to run within the lifetime of the human race for problems of interest, and the algorithm is small enough to fit on the combined computational memory of the Earth's computers, then we believe that the algorithm can be found, eventually. If P != NP, then this research program will never be capable of

proving this fact. And yet, if P != NP, the program's continued failure over the years can be taken as strong Bayesian evidence that P != NP, or at least that P = NP is not true in any way that is relevant to humans as a species.

Given the enormous computational resources that seriously pursuing this research program would entail, we believe that it is necessary to provide something of value in return for all of the computation; the world does not need another Bitcoin network. We therefore propose that a computational system be built that provides an API for solving NP-hard problems of interest to the human race. These problems will be solved on a best-effort basis by a collection of open-source algorithms, running on distributed hardware, in the style of the SETI@home project, or the Folding@home project. I do not know what the most appropriate way would be to gate access to the system; we don't want people running certain computations on this network, such as protein-folding computations for viral gain-of-function research. Perhaps a very large and very deep collaboration between governments and academics could be set up, in order to carefully screen what queries are run on the system, and by whom.

This is the philosophical underpinning of the class of attacks. The rest of this paper is organized as follows. In section 3, we describe our main result, the Probably Unprovable Theorem. In section 4, we give some examples of previous attacks on the P vs. NP question that can be combined into a flaky polynomial time algorithm for 3-SAT. (Flaky in this context means that the algorithm will fail on some instances.) In section 5, we give a fictional story that helps to ground the necessary intuitions. In section 6, we discuss the problem of public channel coordination. In section 7, we lay out some necessary definitions. In section 8, we give a second fictional story that helps to ground more necessary intuitions. In section 9, we discuss potential limitations of deep neural networks, and what it would take to prove them correct. In section 10, we discuss our conjecture that P vs. NP is unprovable within constructive mathematics, and give a heuristic argument that supports the weak form of the Probably Unprovable Theorem. In section 11, we lay out a series of proposed laws of computational science. In section 12, we lay out a heuristic argument for the strong form of the Probably Unprovable Theorem. In section 13, we discuss empirical evidence from deep reinforcement learning that supports the contention that the polynomial time hierarchy does not collapse. In section 14, we discuss the concept of within-lifetime reinforcement learning. In section 15, we discuss the Millennium Prize for solving the P vs. NP problem. In section 16, we posit the Maybe Unprovably Hard Conjecture. In section 17, we discuss the question of whether public key cryptography will stand up to powerful learning algorithms. In

section 23, we will discuss related work; this section is as yet unwritten.

3 The Probably Unprovable Theorem

Let $E(c)$ be the set of all polynomial time algorithms for 3-SAT with exponent at most c. Let $A(c)$ be the set of all valid proofs that $E(c)$ is empty.

We note the following trilemma:

1. Either $E(c)$ is non-empty, and P = NP,
2. Or $A(c)$ is non-empty, and P != NP,
3. Or both are empty, and P != NP, but no verifiable proof that P != NP exists.

We note that, if either of $E(c)$ and $A(c)$ is non-empty, then this fact may require a vast and impractical computation to verify the validity of any particular algorithm for 3-SAT, or the validity of any particular proof that P != NP.

We further note that all existing computational devices are imperfect, if only due to the existence of cosmic rays. Thus, no computation can be 100% trustworthy.

Theorem 1 (Probably Unprovable Theorem, Weak Form)**.** *If $A(c)$ is non-empty, but the problem of verifying membership in $A(c)$ is hard for some complexity class $\mathcal{C}$, then either it is impossible to verify membership in $A(c)$ in polynomial time, or $\mathcal{C} \subseteq P$. In particular, if it is NP-hard to verify proofs that P != NP, then P = NP.*

The weak form of the probably unprovable theorem is quite straightforward to prove, and we prove it later.

We also give a strong form:

Law 1 (Probably Unprovable Theorem, Strong Form)**.** *If $A(c)$ is non-empty, and has an element x of length n, then no polynomially (in n) bounded computational process running on imperfect hardware can establish the validity of x in a way that is truly mathematically trustworthy.*

The strong form of the probably unprovable theorem seems to itself be probably unprovable, and thus we have labeled it a "law".

We give a heuristic argument for the strong form later.

4 Proof of the Sometimes Solvable Theorem

The proposed class of new attacks starts with some very old attacks. We give a specific example of a particular attack below, but this should be considered only an example for the purposes of illustration.

Consider an arbitrary NP-complete problem P. Consider a specific instance x of this problem. We perform a series of classic reductions on the instance x. First we apply the Cook-Levin Theorem to transform x into an instance of 3-SAT. Then we apply the PCP Theorem to transform this into an instance of Gap 3-SAT. Then we apply Hastad's Theorem to transform this into an instance of Gap E3-LIN. Then we apply the method of Purdy (2008) to transform this into an instance of Unique Games (Khot). We then apply various known algorithms for solving instances of Unique Games.

Unfortunately, the soundness of Purdy's method and the known polynomial-time algorithms for Unique Games rely on the constraint graph of the instance being an expander graph. (In the case of known algorithms for unique games, this is a standard definition of expander graphs. In the case of Purdy's method, there is a slightly weirder definition called a "locally expanding hypergraph".) Random instances of Gap E3-LIN will usually be locally expanding hypergraphs, but only at densities far greater than are typically considered relevant.

We thus wind up, via a reasonably efficient polynomial time algorithm, which can be implemented in an actual computer program, with either a verifiable solution to x, or we wind up with an instance y of Unique Games that we still cannot solve, or we wind up with an instance y of Unique Games that we can solve, but we discover that Purdy's method has failed us and thus the Gap E3-LIN graph was not a locally expanding hypergraph.

In the first case, we are happy, because we have solved an NP-hard problem of interest to the human race. In the second case, we are still happy, because we have gathered a very, very small piece of Bayesian evidence that P != NP; not only this, but also we have discovered a tiny piece of Bayesian evidence about which particular instances of the original problem are difficult, which is actually much more useful and relevant information than the single bit of the P vs. NP resolution. In the third case, we are yet again happy, because we have discovered an important piece of information about the original query, namely that its image under the various reductions applied is not a locally expanding hypergraph at the Gap E3-LIN stage. I have not verified this yet, but it seems like we might even be able to compute some sort of graph cut in the Gap E3-LIN instance just by observing the actual pattern of failures that occurred, which it might be possible to lift

all the way back to the original query; whether this lifting process will prove useful is an unknown, but the possibility illustrates what we believe is a general pattern, which is that failed queries may still provide useful information to complexity theorists, and also domain practitioners in the domain of the original query.

5 A Fictional Tale

We begin with a fictional tale to build the necessary intuitions. Following Babai, we set our tale in the Arthurian canon.

Arthur and Mordred are locked in total war. Many valiant knights on both sides have perished. As the war continues, Arthur and Mordred are forced to turn more and more to their most powerful magicians: Merlin (for Arthur) and Morgan le Fay (for Mordred).

One day, Morgan le Fay casts a most terrible piece of magic, and forges the ring of Gyges. Mordred can put on the ring of Gyges, becoming invisible. By turning the collet of the ring, Mordred can, instead of becoming invisible, look exactly as he chooses; in particular, Mordred can look identical to Arthur. No one will ever be able to tell the two apart ever again.

Being a deeply wicked and powerful magician, Morgan le Fay then begins a mass production process, to deliver rings of Gyges to every soldier in the ranks of Mordred's army. Now each member of Mordred's army can become invisible, or look exactly as they choose.

Arthur's forces are devestated by this. The forces of good lose some of their best champions, including Lancelot, to the ensuing battlefield defeats. Of the great champions of old, only Galahad remains to fight.

Arthur consults Merlin, who easily deduces what has happened from the pattern of defeats, but is completely unable to devise a solution. Arthur comes up with the idea to consult Nimue, the Lady of the Lake, who gave him Excalibur.

Nimue makes a compact with Arthur and Merlin. If Merlin will agree to be bound within a magical tree for all eternity after the conclusion of the war, Nimue will grant him one small favor: when Merlin has true need, and only when he has true need, he may prick his finger to release blood; when he does this, he will know with absolute certainty whether or not the person standing before him is Arthur, and also he will know with absolute certainty whether there are any invisible people within 1000 yards of his current location.

Merlin readily accepts the compact, for being bound in a tree for all

eternity is a very small price to pay to defeat the wicked Mordred and his evil forces.

How do Arthur and Merlin defeat Mordred, or at least drive the war to a stalemate, so that an honorable peace may be concluded?

Answer: Mordred really made a tactical error by passing out rings of Gyges to his wicked forces like candy. This will introduce almost arbitrary amounts of coordination problems. So really, the Knights of the Round Table just need to keep Arthur and Merlin alive while Mordred's army tears itself apart in order to win the war. This can be done easily if none of the Knights ever put on a ring of Gyges: Galahad can act as Arthur's bodyguard, and the rest of the Knights can maintain an exclusion zone around Arthur, Merlin, and Galahad as Merlin announces the location of any invisible enemy soldiers approaching. Since all of the knights are stout and true, and none of the knights ever puts on a ring of Gyges, Merlin can trust that everyone else within the exclusion zone is who they seem to be. Since Merlin is announcing the location of invisible enemy soldiers, the knights can easily dispatch them as long as they don't come all at once. So really the outcome of the war just depends on Mordred not knowing where Arthur is for long enough for the forces of evil to tear themselves apart. Ultimately, Mordred will be assassinated and replaced a truly awesome number of times by his invisible bodyguards, making it essentially impossible to maintain a unified command structure long enough to matter.

6 Public Channel Coordination

We examine the problem of *public channel coordination*. Specifically, we wish to design protocols that will allow agents who do not trust each other to communicate over potentially compromised public networks, using potentially compromised cryptosystems. Our ultimate goal is to allow the solution of incredibly difficult coordination problems between mortal enemies.

We posit that some computations are dangerous to perform, and that some are safe to perform. The safety or danger of a computation has no relation whatsoever to the actual computational problem that has been posed; it is, however, roughly deducible if one knows the entire history of why the question was asked in the first place.

For the sake of a concrete example, we consider the problem of implementing the system described in the introduction, without allowing any user of the system to perform protein folding experiments for viral gain of function research. The necessity of forbidding such experiments is hopefully

clear to anyone who is still alive after the COVID pandemic.

So, we would like to be very careful about what queries are allowed to pass into the system from within a virology laboratory. The general problem is much harder to solve than just that simple rule, but hopefully this example gives a feeling for the sort of safeguards we imagine.

7 Theoretical Definitions

We consider two- and three-player games of imperfect information. We are most interested for the moment in zero-sum games, but in the future may consider other classes of games. In the two-player context, we will call the players Arthur and Mordred. In the three-player context, we will call the players Aragorn, Elrond, and Sauron. In the two-player case, our designs are meant to ensure that Mordred can be defeated. In the three-player case, our designs are meant to ensure that Sauron can be defeated without changing the balance of power between Aragorn and Elrond. In a real-life two- or three-player game, everyone will of course have their own opinions about which side is Mordred or Sauron, so the better solution to the three-player case is instead to ensure a sort of three-way stalemate that preserves the balance of power between all three sides without needless bloodshed.

We consider multiplayer games of imperfect information that take place on the nodes of a graph. The nodes of the graph are meant to represent physical locations on Earth or elsewhere. Each player controls some number of pieces, which are tokens meant to represent humans and other computational agents. Each piece resides at a particular node at each timestep; only one piece can occupy a given node. The nodes of the graph are connected via edges representing the potential for one piece to communicate with another piece over a communication channel. Channels may be public or private. Public channels are visible to all players and all pieces. Private channels are visible to some subset of players and pieces at every point in time. Private channels may, over time, become visible to larger or smaller subsets of players and pieces. Since the players and pieces can remember anything they have observed in a private channel they once had access to, it is simpler to assume that the access to a private channel is monotonically increasing. That is, private channels can be treated without loss of generality as if players and pieces never lose access to them once they have gained it.

In addition to 2- and 3-player games, we consider massively multipolar games, in which each piece has free will and is controlled by a separate

player. This seems like the only framework capable of modeling the real world.

8 Another Fictional Tale

Frodo is bequeathed a ring by his uncle Bilbo. Gandalf, being both wise and deeply suspicious of Bilbo's gentle aging process, suspects that the ring is a ring of Gyges. He thrusts it into the fire in Bilbo's home, and is proved right; the runes appear in the dark speech of Mordor, labeling it clearly as such, in rhyming couplets full of puns.

Gandalf realizes that the ring must be destroyed, and the fellowship travels to Rivendell with extreme caution, just barely dodging a contingent of Sauron's ringwraiths.

Gandalf and Elrond consult with the fellowship of the ring. The ring must be destroyed, and yet it can only be destroyed in the fires at the heart of Mount Doom where it was forged. To take the ring to Mount Doom would be insane, as it would risk delivering the ring to Sauron directly, and the war would then be lost.

Legolas suggests that perhaps a second ring could be forged; then Frodo could wear one ring, and he could wear the other. The rings would swiftly corrupt them if worn, but perhaps a decisive strategic advantage could be obtained over Sauron with two rings instead of one, and if one was gifted to the Elves, and carried by Legolas, and the other to the race of Men, and carried by Frodo, then perhaps their personal virtues would be enough to survive the corruption for long enough to win the war against Sauron. Gandalf establishes a two-way psychic link with Galadriel, but neither of them can figure out how to forge a second ring. It seems that, in truth, a second ring could only be forged in the fires at the heart of Mount Doom, and its forging would require the presence of the first ring. To take the first ring into the heart of Mordor would be madness; one does not simply walk into Mordor. And yet a second ring is the only way to maintain a balance of power between the races of Men and Elves. And after the war is concluded, even a third ring would have to be forged, and gifted to what remains of the Orcs, in order to set up a stable tripolar situation that will last until the next age of Middle Earth.

How can Gandalf and Galadriel forge a second and third ring, given that they cannot take the first ring into Mordor, and that the only place the new rings may be forged is in the fires of Mount Doom at the heart of Mordor? Aragorn is quite certain that no frontal assault against Mordor

can be successful. Boromir is quite certain that no subterfuge will be clever enough to get them to Mount Doom without detection, and delivering the ring to Sauron.

What must the Fellowship do? How can they do it?

Answer: The only answer I can think of is that the Fellowship must parley with Sauron. They can do this easily, because Pippin has a palantir that Sauron is watching. Gandalf can use the palantir to establish a two-way psychic link with Saruman and iron out the details of the peace treaty. Aragorn initially refuses to parley with Sauron, but the Fellowship cannot see any alternative that will preserve Middle Earth. Ultimately, a peace treaty is concluded between Sauron and the free peoples of Middle Earth, on terms that both parties are willing to accept. With the war over, Gandalf, Galadriel, and Saruman are free to collaborate on the problem of forging the three necessary rings. Since only one ring may rule them all, a fourth, more powerful ring is forged and given to Eru Iluvatar.

Eru Ilúvatar, also known as the One, is the single omniscient, omnipotent, and omnibenevolent creator. He has been existing eternally in the Timeless Halls and possesses the Flame Imperishable in his spirit which kindles existence from nothingness.

Since Eru Iluvatar cannot be reached on short notice, cheap plastic knockoffs are made of the fourth ring, and distributed amongst the populace at large. The fourth ring is forged of mithril, so that it has the same weight and is indistinguishable from the plastic versions. The plastic rings are placed in an enormous tumbler, along with the mithril ring. The tumbler is mixed with great care as Gandalf, Galadriel, and Saruman keep their eyes closed. Aragorn, Elrond, and Sauron are on hand to make sure that none of the wizards peek.

Ultimately, the new one ring winds up on the hand of an unknown person in an unknown place. It rests in wait, against the day that it will be needed again.

9 Conjecture: Limitations of Deep Neural Networks

We conjecture that a fixed deep neural network is unable to solve an instance of Gap-E3LIN, at least given the most natural encoding strategy. A proof sketch is provided below, but it has so many unknowns in it that it doesn't make sense to call it anything other than a conjecture. The purpose of including it in this document is to demonstrate the extraordinary lengths

that one would be required to go to in order to prove that P != NP.

Conjecture 1. *A fixed deep neural network is unable to solve an instance of Gap-E3LIN, given the most natural encoding strategy.*

Proof. Perform mechanistic interpretability on the neural network to discover how it thinks. Find a flaw in its reasoning; if P != NP, this will exist, if only because of floating point limitations. Build an instance of Gap-E3LIN that exploits the flaw. Verify that the neural network classifies the Gap-E3LIN instance correctly. Then flip a small number of constraints to move the Gap-E3LIN instance across the gap, without creating a new solution that crosses back across the gap. By construction, this will not flip the prediction of the neural network, and thus the neural network has misclassified the example. □

Unfortunately, the above proof sketch is deeply non-constructive. For a polynomially bounded computational agent to complete the proof for any fixed neural network would be a massive undertaking. We provide this mainly as an example of how extremely difficult it would be to prove that P != NP given finite computational resources.

We further note that proving the conjecture wouldn't even come close to proving that P != NP. One could simply allow the neural network to learn over the course of time, making the above conjecture inapplicable. The phenomenon of grokking in deep neural networks suggests that networks can (sometimes) converge to essentially perfect performance on (some) algorithmic tasks when trained for an unreasonably long amount of time.

10 Conjecture: P vs. NP is undecidable within constructive mathematics

Conjecture 2. *The P vs. NP question is undecidable within constructive mathematics. More specifically, any computationally-generated proof that P != NP will contain at least one flaw.*

The significance of this conjecture should be obvious. The larger a proof is that P != NP, the more places there are for a flaw to hide.

If the conjecture is true, then there are some rather bizarre consequences. Let E be the set of polynomial-time algorithms for 3-SAT. Let A be the set of valid proofs that P != NP. We then note the following trilemma:

1. If P = NP, then E is non-empty and A is empty.

2. If P != NP is provable within constructive mathematics, then E is empty and A is non-empty.

3. If P != NP is not provable within constructive mathematics, then both E and A are empty.

Empirically, it seems to be difficult to devise and prove the validity of polynomial-time algorithms for 3-SAT. Also empirically, it seems to be difficult to devise and prove the validity of polynomially-bounded proofs that P != NP. However, every purported polynomial-time algorithm for 3-SAT has been shown to contain a flaw, and every purported polynomially-bounded proof that P != NP has either been shown to contain a flaw, or the proof is sufficiently large and its provenance sufficently untrusted that potential verifiers have simply refused to engage with it.

Consider any finite number of polynomially-bounded computational agents cooperating to try to establish the resolution to P vs. NP. Consider any protocol between them that they will agree in advance to participate in, that they also agree will establish that P vs. NP is true or false. We can then ask the question, what is the expected runtime of such a protocol? How many rounds would it take to execute, and what computational resources would be required to complete it? The experience of complexity theorists since the beginning of the field suggests that the difficulty and required computational resources would be non-trivial, to say the least.

In particular, consider the problem of 2-party verification of proofs that P != NP. Suppose that a protocol is agreed upon, with a finite number of rounds, that will be sufficient to demonstrate that P != NP. This protocol's validity, if that validity could be proven, would constitute proof that the problem of checking proofs that P != NP resides at a particular finite level of the polynomial time hierarchy.

Suppose that it could additionally be proved that the problem of verifying proofs that P != NP is hard for some complexity class $\mathcal{C}$. Then any problem in $\mathcal{C}$ could be reduced to the problem of verifying proofs that P != NP. This would in turn show that $\mathcal{C}$ is a subset of the finite level of the polynomial time hierarchy that we agreed that the problem of checking proofs that P != NP resides at.

We can summarize this state of affairs as follows: P vs. NP empirically seems to be difficult to prove or disprove. Any proof that was accepted to demonstrate that that difficulty resides at any particular level of the polynomial time hierarchy, together with any proof that was accepted to resolve the question itself, would collapse the polynomial time hierarchy,

which seems unlikely. Thus, any proof that demonstrated the things we believe to be true would have deeply unintuitive consequences that we believe to be false.

We believe that this rather informal series of arguments establishes that the undecidability conjecture has high likelihood of being true. Specifically, all results that have ever been proved in complexity theory are consistent with the conjecture. If the conjecture were false, then P vs. NP would either be easy to solve, or the polynomial time hierarchy would collapse. Since P vs. NP does not seem to be easy to solve, it therefore seems almost certain that, if the conjecture were false, the polynomial time hierarchy would collapse.

We can summarize this situation in the following table:

	No PTH collapse	Yes PTH collapse
P = NP	Contradiction	Consistent but seems unlikely
P != NP, easy to verify	Consistent but seems unlikely	Consistent but seems unlikely
P != NP, hard to verify	Contradiction (by this paper)	Consistent but seems unlikely
P != NP, impossible to verify	Consistent, maybe likely?	Consistent but seems unlikely

11 The Laws of Computation

We posit a series of laws for computer science. In science, a law is a falsifiable prediction that has never been falsified, despite extensive theoretical and experimental attempts to do so. Such laws form the very bedrock of a scientific field.

We propose the following laws:

1. The polynomial time hierarchy does not collapse to any finite level.
2. The first law is unprovable within constructive mathematics.
3. The second law is unprovable within constructive mathematics.
4. Etc.

We argue in this paper that each law seems to imply the next law, as long as one assumes that it is computationally difficult to write down and verify a proof of this fact. Thus, if one believes that the polynomial time hierarchy does not collapse to any finite level, and that complexity theory is a non-trivial endeavor, this suffices to establish belief in the entire series of laws by induction. The base case (polynomial time hierarchy does not collapse) is not established by this argument, making the induction suspect. However, no human being has ever successfully proved that the polynomial

time hierarchy does or does not collapse, and most complexity theorists find it more likely that it does not collapse than that it does.

12 A heuristic argument for the strong form of the Probably Unprovable Theorem

Suppose that some polynomially bounded computation takes place to prove that P != NP. Suppose that there is some single flaw in the proof; imperfect computational hardware could in theory mask this flaw. Likewise, suppose that there is no flaw in the proof; imperfect computational hardware could in theory report a flaw that does not exist. Therefore, no polynomially bounded proof that P != NP, evaluated on imperfect hardware, can be trusted.

If we are willing to accept a small probability that the proof is misclassified, then simply building extremely good chips and using the PCP Theorem would of course suffice to establish truly staggering levels of certainty in a purported proof, if such a proof existed, which we are arguing maybe it cannot.

This can be formalized into the Maybe Unprovably Hard Conjecture, which we think is probably unprovable, and thus is headed towards law status rather than theorem status.

13 Empirical Evidence that the Polynomial Time Hierarchy Does Not Collapse

Consider an arbitrary multiplayer game that is tractable via deep learning methods, such as chess, go, Diplomacy, etc. Then the following observations are empirically well-established:

1. MCTS, together with heuristics derived from deep reinforcement learning algorithms, is an incredibly powerful algorithm

2. MCTS is more powerful when you do more and deeper rollouts

If the polynomial time hierarchy collapsed, then it seems likely that MCTS would max out at some finite rollout depth, since the polynomial time hierarchy corresponds to the problem of winning a zero-sum game of perfect information and fixed duration. But empirically, it is observed that even relatively simple games like chess and go require arbitrarily deep search trees in order to be competitive.

14 Within-Lifetime Reinforcement Learning

We consider the class of algorithms that change over time by modifying certain parameters that are stored in their memory. Neural networks that are still learning from experience are perhaps the most salient examples of this class.

We consider the problem of within-lifetime reinforcement learning. (Give some definitions.)

Specifically, we consider the Complexity Theory Game, the game in which agents are rewarded for establishing new complexity theory results to the satisfaction of other agents.

We believe it is fair to say that agents receive -1 point for claiming that they have proved that P != NP, and that an agent would receive 1,000,000 points for actually establishing a social consensus that they have proved that P = NP or that P != NP.

What is the optimal strategy for this game? It seems that there will be two dominant strategies: the humble strategy (don't claim to have solved P vs. NP) and the courageous strategy (try to create valid proofs that P = NP or that P != NP and then announce them). If either E or A is non-empty, and the smallest member of one of them can be found using the computational resources available to any one agent, then the courageous strategy is potentially dominant, since $1000000 > 0$. Otherwise, the humble strategy dominates the courageous strategy, since $0 > -1$.

If both E and A are empty, then nobody can ever collect the 1,000,000 points for establishing the social consensus that they have proved that P = NP or that P != NP. Suppose that there is some unknown number of points $n > 0$ that will be awarded for establishing a social consensus that E and A may indeed both be empty, and that it can be assumed for practical purposes that the smallest element of $E \cup A$ is too large to be easily and efficiently located. There is then a third potential strategy, which we will call the wise strategy, which is to provide theoretical and experimental justification for the proposition that the smallest element of $E \cup A$ cannot be easily and efficiently located, and then announce that. If $E \cup A = \emptyset$, then the wise strategy is clearly dominant, since it achieves a reward of $n > 0$, thus beating both the humble and courageous strategies.

15 The Millennium Prize

There is currently a $1,000,000 bounty on the problem of P vs. NP. Significantly more than $1,000,000 has been expended on attempts to prove that P = NP or P != NP. The theorems in this paper demonstrate that any attempt to claim the prize is likely doomed:

1. If a short proof of either P = NP or P != NP existed, it would have been found by now.

2. If a long, invalid proof is presented, it will consume significantly more than $1,000,000 to find the flaw.

3. If a long, valid proof is presented, it will consume even more resources to fail to find any flaw, and no one will know if there is a flaw, and the prize will take forever to be awarded, and no one will know if it should be.

4. If a long, valid proof that P != NP is presented, and we can prove that it will be difficult to verify the proof, then that proof would also demonstrate that the polynomial time hierarchy collapses.

Therefore, if the polynomial time hierarchy does not collapse, and there is no short proof, and proofs are difficult to verify, then the prize can never be awarded.

If the prize is to be awarded ever, and the polynomial time hierarchy does not collapse, the prize must therefore be awarded for some accomplishment other than a proof that P != NP. It is unclear what would constitute such an accomplishment in the eyes of the committee that administers the prize; it should presumably be for a theorem that is both relevant to the question, and relatively conclusive on the issue, to the extent allowed by the laws of mathematics and the experimental evidence that we have access to.

We would argue that the theorems proved in this paper are at least potentially a candidate for such an accomplishment.

16 The Maybe Unprovably Hard Conjecture

We would like to be able to detect the existence of possible proofs that P != NP well before they become feasible to discover. We conjecture that there is some combination of theory and experimentation that would demonstrate this. Since we conjecture that no proofs exist, this procedure is perhaps

not a fruitful use of time, but we include the conjecture for the sake of completeness.

Conjecture 3 (Maybe Unprovably Hard Conjecture)**.** *Given $\delta > 0, n > 0$, there is some computational budget $N(\delta, n)$ such that it can be established (with probability of error at most δ) via a series of theoretical and experimental methods whether there exists a valid probabilistically checkable proof that P != NP, where the proof is required to be of length at most n.*

17 Does Public-Key Cryptography Survive?

It seems quite relevant to the alignment problem whether public key cryptography will survive when exposed to powerful learning algorithms. We conjecture that some cryptosystems will survive, since otherwise it seems likely that the polynomial hierarchy would collapse. It seems like the algorithm in section 4 will fail to break some cryptosystems just based on properties of the underlying constraint graph of the system. We have not had a chance to investigate this issue in any depth beyond the thoughts contained in this paragraph.

18 Coordination Problems Arising from Status Games

Many people spend a lot of mental energy trying to estimate the status level of people around them. This may be smarts, money, etc., but it generally seems to be a pretty one-dimensional scale for a lot of people: some people are high, and some people are low.

Status can be estimated via competitive games, via the Elo or Glicko rating systems. It can also be estimated via observing feats of strength in a two-sided Elo or Glicko rating system, as in the chess puzzle ratings on chess.com. This seems like a reasonable approximation to how people estimate such things in their head.

What would happen if a low-rated player claimed to have solved a high-rated puzzle? Would anyone believe them? It would boggle the imagination. It seems prima facie impossible.

Consider again the Complexity Theory Game, where computational agents try to establish results in complexity theory to the satisfaction of other experts in complexity theory. How would you establish that you solved a high-status puzzle if you yourself are low-status? There are two basic strategies: find one or two highly capable, high-status experts to just complete the work

on their own or with their own massive network of collaborators, or find a large number of highly capable people at a similar status to yourself who can collaborate with you and with others.

In particular, consider the social graph, with directed edges between people indicating which of them seems to be higher status. If there is a single axis of status, then there are a large number of graph cuts that lie between a given low status person and any high status person, where each edge that crosses the graph cut goes the same way. This makes it essentially impossible for the result to percolate upwards, as each communication edge has a substantial probability of the higher status person refusing to believe that the lower status person has solved a puzzle that they could not. However, if a path of length one to an extremely capable, extremely high status person can be found, then perhaps the extremely capable, extremely high status person will be mildly curious and actually read the damn thing.

One can also attempt to coordinate with a large number of people at a similar status to oneself. Since these are edges between equals, no particular status games will exist, and the argument can stand on its merits or lack thereof.

Ultimately, the actual best solution is to attempt to bypass any possible graph cuts at all. Consider directed tournaments. Call a directed tournament turbulent if there is no graph cut that separates the nodes into a higher portion and a lower portion without at least a constant fraction of the edges going the opposite way. (This is a closely related notion to that of expander graphs.) We hypothesize that turbulent tournaments are ideal for intellectual collaboration, because the necessary ideas can flow freely between equals with no status games or credit games. This seems to be how the PCP Theorem was initially proved; several of the participants in that achievement seem to credit the achievement to the existence of email, which seems plausible, since the semirandom and extremely nonlocal nature of who emails who would naturally establish an expander graph structure, which is also necessary to make sure that all the necessary ideas can be combined.

We also note that, in the case of this paper, no puzzle has really been solved. We've just pointed out that it's conceivable that the puzzle has no solution, which feels more like a draw than a win. If the previously estimated Elo/Glicko rating of P vs. NP was close to infinity, and a low-rated player has a draw with P vs. NP, then it just means that P vs. NP was roughly at the same level as the low-rated player, not that the low-rated player also has an infinite Elo/Glicko rating.

We also note that, if P vs. NP is actually not provable within constructive mathematics, then attempts to prove that P != NP are essentially the

same as designs for perpetual motion machines; they are worthless, and the more intricate they are, the more annoying they are.

19 The Safety Quantifier

Consider three-player games between Aragorn ($\forall$), Elrond ($\exists$) and Sauron ($). We call $ the safety quantifier. Aragorn and Elrond are having a collegial discussion in which they try to determine what is true about the universe in which they reside. Sauron is trying to kill both Aragorn and Elrond by any means possible. How do Aragorn and Elrond agree on the truth without allowing Sauron to kill them? Playing such games skillfully using deep reinforcement learning seems eminently possible, and if Aragorn and Elrond between them possess more computation than Sauron does, then their MCTS rollouts will be deeper and more informative, causing them to win.

20 The Cut Method for proving bounds inside a computational paradigm

We consider the problem of counting the number of counterfeit coins in a set of n coins, using only a balance. Previously, we proved an upper bound of $\log(n) * \log(n)$ and a lower bound of $\log(n)$ for this problem. We conjecture that this bound cannot be significantly narrowed except through experimentation, which can only ever work for finite values of n. Thus, providing a proof of a matching lower bound may be another probably unprovable theorem, and providing a proof of correctness for any finite algorithm may be another probably unprovable theorem.

We think that the cut-based method is a promising approach for proving lower bounds inside simple computational paradigms. Unfortunately, the Von Neumann architecture for a computer introduces arbitrary amounts of expander graph structure inside any modern computer, and thus the cut-based method provides literally no information in practice.

21 The Navier-Stokes Equation

We generalize the problem slightly to include partial differential equations in an finite but unlimited number of variables. We then implement a deep reinforcement learning algorithm in those equations (very straightforward,

since learning by gradient descent is a partial differential equation), and use it to train a large language model on the combined data of the human race. We thus wind up with a highly capable general intelligence that we can ask questions of. We could give it Godel-style instructions, like, become turbulent if and only if you will not become turbulent in this situation. The intelligence may then do one of three things:

1. Obey, and become turbulent
2. Disobey, and not become turbulent
3. Ignore us, and do what it chooses to do, becoming either turbulent or non-turbulent as its parameters dictate.

Since we cannot predict which of these four it will do without solving the mechanistic interpretability problem, we cannot predict what the n-dimensional fluid will do.

This argument makes heavy use of the number of parameters being potentially unlimited. The actual resolution of the Navier-Stokes problems in a finite number of dimensions is thus not addressed by this argument. The most straightforward way to finish the proof for a finite number of dimensions is to implement a Turing machine inside the equations, and then invoke the Probably Unprovable Theorem. We have not had time to do this, but it doesn't sound particularly difficult. We leave it as an exercise to the reader; good luck with the prize committee!

22 The Massively Multipolar Alignment Problem: A Fictional Tale

We set our tale in the Harry Potter universe. Voldemort has a ring of Gyges, and Harry Potter has a cloak of invisibility. Since Voldemort has a ring of Gyges, his followers suffer from various coordination problems, while Harry and his friends do not. As long as Harry keeps at least one deathly hallow away from Voldemort, things will be fine. This can be done approximately as described in the original books.

23 Related Work

Haven't started this section yet, will include it in later drafts.