OpenAI's claimed refutation of ETH: what it says, and how we checked it

READING LEVEL EVERYONE · EXPERTS

Active · Preliminary

proved in writing 2 · Lean-checked 1

Overview

OpenAI says one of its AI models has refuted the Exponential Time Hypothesis, called ETH, a conjecture that a yes-or-no puzzle named 3-SAT needs some fixed exponential amount of time to solve. Hundreds of papers use ETH to explain why a problem probably cannot be solved much faster, so those explanations lose their support if the claim holds. This book does not decide whether the claim is right. It is an audit of OpenAI's published Lean statement and proof. The statement matches the standard reading of ETH, checked in writing for 12 of the 14 rows of the statement table. All 3,091 modules of the proof build under Lean 4.34.1 with Mathlib, and the final theorem depends only on the three standard axioms. That was checked by machine. A kernel replay of each module has finished: all 3,091 were accepted and none rejected. The manuscript's mathematics, 67 pages, has not been checked line by line, and five items are open, among them a second-kernel run with the unpatched checker. Others have run the comparator tool, and it accepts. The review group's examination is ongoing, so the right description is a claimed refutation. The check guide gives the commands for repeating the checks, which need about 16 GB of free memory and many hours.

CHECK IT YOURSELF

The check guide for this book · How to check a result yourself

---

Cite: Sonny (2026). OpenAI's claimed refutation of ETH: what it says, and how we checked it (preliminary). Go check yourself. https://gocheckyourself.org/read/eth-audit/everyone/

1. Start here
2. OpenAI's claimed refutation of ETH, in plain words
3. Try it yourself

Start here

What this book shows

- What I checked. OpenAI's claimed refutation of ETH, the Exponential Time Hypothesis: whether its published Lean statement says what ETH says, and how far Lean accepts the proof. (Lean is a language for writing mathematics so exactly that a small program, its kernel, can check every step.)
- The statement matches the standard reading of ETH. With k = 3, a correct proof of it would contradict ETH. Checked (re-derived in writing): 12 of the 14 rows of the statement table.
- The proof builds, on the standard axioms only. All 3,091 modules build under Lean 4.34.1 with Mathlib (the Lean community's library of formal mathematics), and the final theorem depends only on the standard three axioms (starting assumptions Lean accepts without proof). Checked by machine. A kernel replay of each module (every stored proof step of the module is sent through the kernel a second time) has finished: all 3,091 accepted, none rejected.

- Why it matters. Hundreds of papers use ETH to explain why a problem probably cannot be solved much faster; if the claim holds, those explanations lose their support. The manuscript's mathematics has not been checked line by line, and five items are open (chapter 4).
- What our tools showed. The same list of checks (statement, build, axioms, replay) can be used on other "checked in Lean" claims. The statement and our two small facts about it are in the theorem library.

The question

OpenAI says one of its AI models has refuted ETH. I am not an author of that work. I read the Lean theorem, rebuilt the proof with public tools, and wrote down what I could and could not check.

Why this book exists. The subject of this book is ETH. When OpenAI released its proof, we checked it carefully, in the spirit of shared knowledge. OpenAI also did something we want to see more of: it released its proof in a form anyone can check. We applaud

that. We think releases like this would be stronger still with agreed checks before publication, like the responsible release threshold we propose, and we are encouraged by the open discussion that has followed.

What does this actually mean?

- Not Ethereum. "ETH" here is the Exponential Time Hypothesis, a conjecture in computer science about how long certain puzzles take to solve. It has nothing to do with the cryptocurrency.
- The speed-up is asymptotic. The claimed bound describes how running time grows as puzzles get larger, and it comes without published constants, so no practical speed-up has been shown at any size. OpenAI's own post says turning such advances into practical gains "will require significant further research".

- The claimed running-time bound. ETH implies that the polynomial-time question has a negative answer (not every problem whose answers can be checked quickly can be solved quickly), but refuting ETH says nothing about the polynomial-time question itself. The claimed algorithm is still not polynomial time (running time that grows like a fixed power of the input size). Problems that convert to 3-SAT (a yes-or-no puzzle: can a list of conditions all hold at once?) with only a linear increase in size would inherit a speed-up if the claim holds.
- Encryption. As far as we know, mainstream encryption (AES, RSA, elliptic curves) does not rest on ETH, so this indicates no weakening of it. Chapter 0 says what we base this on.
- Research. If the claimed refutation holds, results that assumed ETH lose that support. Results resting on other hypotheses are not touched.

- Status. It is a claimed refutation. The Lean proof builds with only the standard axioms on our machine, and the review group's examination is ongoing.

The question this book answers: what does the Lean statement OpenAI published say, does it match what ETH says, and how far have we been able to check that the proof is accepted by Lean?

Why it matters

A computer-science puzzle called k-SAT asks: given a list of conditions ("clauses"), each an "or" of at most k yes-or-no variables, such as "x1 or not x2 or x5", can every variable be set to true or false so that all the conditions hold? (A formula written this way is a k-CNF.) With n variables there are 2^n settings, and trying them all (brute force) doubles in cost with each added variable. For decades the best algorithms for 3-SAT (k = 3) still took time that grows exponentially in n.

The Exponential Time Hypothesis (ETH), proposed by Impagliazzo, Paturi and Zane around 1999 to 2001, says this is unavoidable: no algorithm solves 3-SAT in time

2^{o(n)}, meaning time growing more slowly than 2^{c·n} for every fixed c above zero. Put plainly, 3-SAT needs some fixed exponential amount of time, however clever the algorithm. A stronger form, SETH, says that for every ε above zero there is some clause width k for which k-SAT cannot be solved in time 2^{(1−ε)n}.

ETH matters because hundreds of papers, in graph theory, computational biology, databases and computational geometry, use it as a working assumption to explain why a problem probably cannot be solved much faster than it is now. If ETH fails, those explanations lose their support.

What answering it would help with

- Readers worried by headlines may use the plain answers in chapter 0 to see what changes and what does not.
- Researchers who relied on ETH may use the statement table in chapter 1 to see exactly which version of the claim has been formalised.
- Anyone who meets a "checked in Lean" claim may use the same list of checks (statement, build, axioms, replay) on other results.

What our research shows

What this is

A BOOK ON OPENAI'S CLAIMED REFUTATION OF ETH (26 OCTOBER 2026): WHAT ITS PUBLISHED LEAN STATEMENT SAYS, LINE BY LINE, AND HOW FAR WE HAVE CHECKED IT WITH PUBLIC TOOLS (SONNY, 2026).

Checked by

THE PROOF WAS BUILT WITH LEAN 4.34.1 AND THE PINNED MATHLIB VERSION (ONE EXACT VERSION, LISTED IN THE CHECK GUIDE); ITS AXIOMS WERE PRINTED; A KERNEL REPLAY OF ITS 3,091 MODULES HAS FINISHED, WITH ALL 3,091 ACCEPTED; A SECOND READING WITHIN THIS PROJECT RE-DERIVED THE STATEMENT-MATCHING POINTS OR CITED THEIR SOURCES. THE MANUSCRIPT'S MATHEMATICS HAS NOT BEEN CHECKED LINE BY LINE.

Labels used throughout: checked means re-derived in writing; checked by machine means a program confirmed it; cited means we opened a source; open means not settled. (A module is one Lean file of the proof.)

Why we audit published results: About this research

<!--BOX-->

What's in this book

Status	Count
Checked by machine: the whole proof builds in Lean, and its axioms are the standard three and no others	1 (the final theorem)
Checked by machine: kernel replay of each module	3,091 of 3,091 accepted, none rejected
Checked (re-derived in writing): rows of the statement table in chapter 1	12 of the 14 rows
Cited: sources we opened for the definitions of ETH and for the known proof-system lower bounds	see chapter 3 and the credits
Open	6 (listed in chapter 4)

Item	Count
Chapters	5 (numbered 0 to 4)
Statement files	1
Written check recipes	1 (the check guide)

<!--/BOX-->

What OpenAI claims

OpenAI reports an algorithm that counts the satisfying settings of any k-CNF formula (k fixed, 2 or more) with n variables exactly, deterministically and in polynomial memory, in time $2^{O(n (\log \log n)^2 / \sqrt{(\log n)})}$ times a polynomial, which is $2^{o(n)}$, with a proof formalised in Lean. The manuscript, "The Exponential Time Hypothesis is False" (17 October 2026, revised 25 October), claims sharper exponents, which are not in the Lean statement. A review group led by Russell Impagliazzo, Ramamohan Paturi and Ryan Williams is verifying and explaining the result, and OpenAI has made the proof public at github.com/openai/math (we used the pinned commit listed in the check guide).

What we checked and found

- The statement matches the standard reading of ETH. For each k, the Lean theorem gives one fixed machine (a Turing machine, the textbook model of a computer) that, on every k-SAT formula, halts with the exact count and the yes-or-no answer, within the stated time and using only polynomial space ("polynomial" means growing like a fixed power of the input size). The machine is chosen before the formula, it is deterministic (no coin flips), and the count is exact. With k = 3, a correct proof of it would contradict ETH. Checked; chapter 1.

- The whole proof builds. All 3,091 modules in the proof build under Lean 4.34.1, with Mathlib, the Lean community's library of formal mathematics. Checked by machine.
- The axioms are the standard three and no others. An axiom is a starting assumption Lean accepts without proof. The final theorem depends on the three standard axioms that Mathlib normally uses (named in the check guide), and on no placeholder axiom. Checked by machine.

- Kernel replay. A replay with the tool leanchecker sends every stored proof step of a module back through the kernel to catch anything that slipped past the build. The replay of all 3,091 modules has finished: every one was accepted, none rejected. It replays each module's own steps and does not by itself cover Mathlib or core Lean. Checked by machine.
- The second reader. A second reading within this project re-derived the statement-matching points in writing (checked) and cited the sources for the definitions of ETH and for what the comparator tool guarantees (cited). See chapter 2.

What it means for the polynomial-time question

The polynomial-time question asks whether every problem whose answer can be checked quickly can also be solved quickly. ETH implies a negative answer to the polynomial-time question, but refuting ETH does not imply a positive answer to the polynomial-time question. The claimed algorithm is not polynomial time, and the manuscript derives nothing about the polynomial-time question. The related question NP vs coNP, whether "no solution exists" can always be backed by a short proof, is not settled by it either. If the claim holds, problems that convert to 3-SAT with only a linear increase in size, such as 3-Colouring, Independent Set and Vertex Cover, would also get $2^{o(N)}$ algorithms. See chapter 3.

What is still open

Five items, listed in chapter 4: a second, independently written kernel with the unpatched checker (one outside run with a patched build accepted; another with the unpatched checker did not finish); the manuscript's 67 pages of mathematics, not checked line by line; the manuscript's other claims (its sharper exponents, corollaries and appendix on local maps), not re-derived; whether the algorithm's correctness gives short proofs in a strong proof system (a fixed set of rules for writing proofs), a conjecture, untested; and any indirect consequence for the polynomial-time question, which nobody can rule out. The comparator tool, which compares the published challenge statement with the proved theorem, has been run by others and accepts; we have not run it.

Check it yourself

Everything needed is public: OpenAI's repository at the pinned commit, Lean, Mathlib at the pinned version, and Lean's own tools. The check guide (VERIFY.md) gives the steps: build the module, print the axioms of the final theorem (expected: the standard three), and replay modules with leanchecker. You need about 16 GB of free memory and many hours. In outline:

```
lake build OAI.Algorithms.SubexponentialSAT.Main
lake env lean Axioms.lean
lake env leanchecker OAI.Algorithms.SubexponentialSAT.Algorithm
```

The last command replays one module, as an example. To replay all 3,091, run the same replay command once for every module that has a compiled file in the project's build folder (VERIFY.md gives the exact command and folder).

The chapters

- Should I worry? Questions people are asking: calm answers on encryption, passwords, AI, the polynomial-time question, and whether a computer proof can be trusted.
- What the Lean statement says, line by line: the statement against the standard form of ETH.
- How we checked it: the build, the axioms, the replay and the second reader, with counts.
- What it means, and what it does not: the algorithm in plain words, other problems, proof systems, the polynomial-time question.
- What is still open: what remains to check.

The statement, the check guide and the credits page complete the book.

We stand on the shoulders of giants: OpenAI, the review group, the originators of ETH and the Lean community are credited on the credits page.

Where to go next

- Does this change anything for passwords or payments? See Should I worry?.
- If a computer accepted the proof, why is the review still going? See How we checked it and What is still open.

Go check yourself.

Have a question about this? Ask it

OpenAI's claimed refutation of ETH, in plain words

OpenAI's claimed refutation of ETH, in plain words

The question: OpenAI says one of its AI models has refuted ETH, the Exponential Time Hypothesis. What does its published Lean statement (Lean is a language for writing mathematics so exactly that a small program can check every step) say, and what does the claim mean for everyone else? The answer in brief: the statement matches the standard reading of ETH, and the proof builds in Lean. As far as we know, it does not break passwords, payments or encryption, and it does not settle the polynomial-time question (whether every problem whose answer can be checked quickly can also be solved quickly).

What ETH is

A computer-science puzzle called 3-SAT asks whether a list of yes-or-no conditions can all hold at once. With n variables there are 2^n settings, and trying them all doubles in cost with each added variable. ETH, proposed by Impagliazzo, Paturi and Zane around 1999 to 2001, says this is unavoidable: 3-SAT needs some fixed exponential amount of time, however clever the algorithm. Hundreds of papers use ETH to explain why a problem probably cannot be solved much faster. "ETH" here has nothing to do with the cryptocurrency Ethereum.

What I checked

I am not an author of OpenAI's work. I read the Lean theorem, rebuilt the proof with public tools, and wrote down what I could and could not check. The statement matches the standard reading of ETH: with k = 3, a correct proof of it would contradict ETH. Checked (re-derived in writing): 12 of the 14 rows of the statement table. The whole proof builds in Lean. Checked by machine. The manuscript's mathematics has not been checked line by line; the open items are listed in the expert version.

Should I worry?

As far as we know, no. Passwords, payments and mainstream encryption (AES, RSA, elliptic curves) do not rest on ETH. The claimed speed-up describes how running time grows as puzzles get larger, and it comes without published constants, so no practical speed-up has been shown at any size. It does not settle the polynomial-time question: ETH implies a negative answer to the polynomial-time question, but refuting ETH says nothing about the polynomial-time question itself. If the claimed refutation holds, results that assumed ETH lose that support.

Why it is worth checking

OpenAI released its proof in a form anyone can check. We applaud that. It is a claimed refutation, and this book is preliminary, not yet refereed.

Experts: the files and checks are here: check guide.

We hope it helps others find more results and keeps their curiosity going. We'd love to collaborate with anyone.

Try it yourself

The figures of this chapter are the interactive checks on the tools page (they need JavaScript).

Each result on its own page

- Theorem 1: exact counting of k-SAT in subexponential time
- Lemma 2: the encoding length is polynomial in n
- Corollary 3: Theorem 1 contradicts the Exponential Time Hypothesis

---

© 2026 Sonny. The books are under CC BY 4.0 (Creative Commons Attribution 4.0 International): reuse them with credit and a link. Each book's licence file lists the files that keep their own licences.