

数学家24小时驳回OpenAI攻破的猜想！“AI证对了每句话，但已跟原猜想无关”

算法与数学之美 2026年8月6日 09:00 安徽

OpenAI宣称下一代AI模型解决了10个世界级难题，其中包括推翻了Connes刚性猜想。

第二天，就有一篇人类论文回应：AI提出的反例不成立。

1. **High-dimensional sphere packing.** New upper bounds on sphere-packing density down to the Cohn–Elkies threshold.
2. **Binary and spherical codes.** Exponentially improved bounds on the maximum size of binary codes at any prescribed minimum distance, with analogous results for high-dimensional spherical codes.
3. **Non-sofic groups.** A construction establishing the existence of non-sofic groups, addressing a central open question in group theory.
4. **Connes's rigidity conjecture.** Disproof of a longstanding conjecture that certain groups are uniquely determined by their von Neumann algebras.

作者是堪萨斯大学拓扑物理中心的J. L. Nielsen，他把OpenAI公开的37000行Lean 4代码从头追到尾，把每一个对象对回它的数学原型，最后给出两条互相独立的失败路径。

On the Invalidity of the Claimed Disproof of Connes' Rigidity Conjecture

J. L. Nielsen

*Center for Topological Physics
University of Kansas, Lawrence, KS*

August 2026

公众号 · 量子位

具体的证明过程咱一般人也看不太懂，让AI和数学家神仙打架去吧。

但通过这件事，说明人类对AI科研结果的审查仍然很关键。

Jennifer Lorraine Nielsen - *Jenny*

Physicist, Philosopher, Mathematician, Quantum Gravity Theorist, Exotic Aerospace Dynamicist, Teacher/Tutor/Lecturer



Jennifer "Jenny" Nielsen (PhD Candidate, MSc Physics, MA Philosophy) is a physicist, quantum gravity theorist, and philosopher of science.

She is the author of the [Topological UFT](#) a true quantum gravity theory written on the complex Hopf fibration. The paper is undergoing peer-review at a reputable mathematics journal as an invited submission. The paper has passed around 4 rounds of peer review.

With her partner Lu Semita she has developed a proof addressing [the Riemann hypothesis](#) and [developed a proof addressing the Clay-Millennium Navier-Stokes problem](#) (Option C). By extending the analysis beyond the Clay problem, she developed a complete solution to Navier Stokes in complex time including [a solution to](#)

什么是Connes刚性猜想

Connes刚性猜想说的是这样一件事。

数学上可以给一个群配一套代数结构。有时候两个长得不一样的群，配出来的结构却完全一样。

Connes在1980年左右提出猜测：**只要这个群满足两个附加条件，这种情况就不会发生，结构一样，群就必须一样。**

这两个附加条件，一个叫ICC，一个叫Kazhdan性质(T)。

换句话说，想反驳这个猜想，需要举出满足两个条件，但构造不一样的群。

OpenAI新模型的做法是构造两个不同构的群，让它们生成同样的代数，并给出两个群都满足ICC和性质(T)的证明。

整套论证写成37000行Lean 4代码，由Lean内核逐条验证通过，另外附了一份说明文档，解释这两个群是怎么搭出来的。

ten-proofs / ConnesRigidity.lean



Code

Blame

37374 lines (33385 loc) · 1.38 MB

公众号 · 量子位

Nielsen指出：**AI构造的其中一个群，其实没有满足附加条件，既不是ICC，也没有性质(T)。**

3 The Construction in the Formalisation

The formalisation constructs two groups Γ_0 and Γ_1 as cocycle extensions

$$1 \longrightarrow \mathbb{Z}^{2n} \longrightarrow \Gamma_i \longrightarrow \mathrm{Sp}(2n, \mathbb{Z}) \longrightarrow 1$$

($i = 0, 1$), where the extensions are classified by normalised additive 2-cocycles

$$c_i \in Z^2(\mathrm{Sp}(2n, \mathbb{Z}), \mathbb{Z}^{2n}).$$

Concretely:

2

- c_0 is the zero cocycle (yielding the direct product),
- c_1 is a non-trivial bilinear cocycle arising from the standard symplectic form on the lattice.

The resulting groups are realised in Lean as `constructedGammaZeroGroup` and `constructedGammaOneGroup`.

The formalisation correctly shows that all *non-central* conjugacy classes are infinite and supplies certificates that both groups possess property (T). (The latter is expected: $\mathrm{Sp}(2n, \mathbb{Z})$ has property (T) for $n \geq 2$ by a classical result of Kazhdan [2], and property (T) passes to extensions.) These facts, while possibly correct in themselves, are irrelevant to the ICC hypothesis of the conjecture, because both groups have non-trivial centres, as we now show.

这其中有三种可能：

代码里性质(T)没有忠实对应Kazhdan的原始定义；或者证明只对部分成立，却被算到了整个群上；又或者代码里那个群根本不是说明文档里描述的样子。

37000行，逐行对号

为了验证这个结论，Nielsen还做了一件更费力的事。

公开发布的代码是一个整合成单文件的版本，早期分模块源码里的那些名字全都不见了。

他于是列了一张对照表，把每个数学对象在新代码里的名字和行号一一标出：

零上闭链群在第13700行，扭曲群在第14069行，两套代数同构的证明在第36712行，主定理在第36954行。

他还把代码证明ICC的整条推理链追了一遍。这条链从第31430行起步，一层层往上传递，最后在第31610行合成结论。

4 Confirmation in the Published Code

The publicly released monolithic file `ConnesRigidity.lean` (37,000+ lines) does not use the names `CocycleExtension`, `ZeroCocycle`, or `TwistedCocycle` that appeared in the earlier modular source files (`CocycleExtension.lean`, `ICC.lean`, `CrossedClosure.lean`). However, the identical mathematical construction is present under different names. The following table gives the correspondence, with line numbers in the published file.

Earlier files / this note	Published code	Line
<code>constructedGammaZeroGroup</code>	<code>lambdaGroup (D ⋈ K)</code>	13700
<code>constructedGammaOneGroup</code>	<code>gammaGroup n (E(n) ⋈ K)</code>	14069
<code>ZeroCocycle (split)</code>	<code>D = V × B</code>	682
<code>TwistedCocycle</code>	<code>shiftedCarry (2-cocycle)</code>	10572
<code>cocycleExtensionDiscreteGroup</code>	<code>CarryGroup (central ext.)</code>	10597
<code>Cocycle addition rule</code>	<code>Add (CarryGroup n)</code>	10615
<code>Sp(2n, Z)</code>	<code>K = ker(SL₄(Z[t]) → SL₄(F₃))</code>	873
<code>Z²ⁿ (lattice)</code>	<code>D = V × B or E(n)</code>	682, 11407
<code>Y central in CarryGroup</code>	<code>shiftedCarry_zero_left/right</code>	10575
<code>Exponent 4</code>	<code>four_nsmul_eq_zero</code>	10683
<code>Exact seq. 0 → V → E → B → 0</code>	<code>iota, sigma, sigma ker</code>	12057, 12346, 12384
<code>ICC claims</code>	<code>actingGroup_isICC, gamma_isICC</code>	10404, 31821
<code>Property (T)</code>	<code>actingGroup_hasKazhdanPropertyT</code>	1399
<code>Factor isomorphism</code>	<code>paper_factors_isomorphic</code>	36712
<code>Main theorem</code>	<code>exists_nonisomorphic...</code>	36954

Nielsen指出的问题是，这些引理处理的是经过对偶变换之后的对象，不是原来那个带中心元素的群，因此并没有直接覆盖到关键的那部分元素。

至于它们是否对最终进入定理的那个具体的群里每一个元素都成立，取决于两块构造之间的接口怎么对接。

这说明“要证什么”就出了问题，不是“证得对不对”的问题。Lean只负责验证后者。

对于另一个扭曲的群，Nielsen的态度是保守的。他说自己没有从代码里独立核实它到底满不满足ICC，也承认代码里的引理有可能确实证明了它满足，但那不改变结论，已经有一个条件不满足。

他把自己的两条反驳也写成了Lean代码，在Lean 4.32.2下编译。

机器检查的是形式，不是意思

论文最后一节把这件事放进了一个更大的背景。

Lean内核能保证的事情，只是一段证明在形式上严丝合缝，但不负责是否真的能证明原结论。

这里可以直接引用陶哲轩的说法：验证证明的是形式陈述本身，而不是这个陈述和意图相符，所以人类的审查不能被取代。

过去这类事故已经有记录。

一份针对五个常用Lean基准的审计给出了4833条发现，包括反例、空洞定理和不可靠公理，全都通过了机器验证。最后是人类构造出反例，才发现被证明的那句话本身是错的。

Faults in Our Formal Benchmarking										
Benchmark	Variant	Prob.	Find.	Proven	Div/0	Nat Sub	Int Div	Analytic	Axiom	CEx
FormalMath	all	5,560	3,250	141	1,127	582	423	351	0	55
	lite	425	213	6	80	34	25	33	0	1
ProverBench	deepseek	325	370	208	81	7	6	19	199	5
ProofNet	original	371	82	5	13	8	3	8	0	5
	sharp (#)	371	67	1	11	6	4	8	0	1
miniF2F	v2c	488	193	8	45	28	20	31	0	3
	v2s	488	181	8	45	29	15	28	0	3
	yangkyl1	488	135	6	34	24	12	16	0	0
	yangkyl1-early	488	79	13	14	19	15	9	0	5
	justincasher	485	74	0	25	13	5	16	0	0
	harmonic	485	74	0	25	13	5	16	0	0
	ai-mo	244	37	1	17	6	3	5	0	0
CombiBench	hf	100	78	1	1	27	2	0	0	1
Total		10,318	4,833	398	1,518	796	538	540	199	79

在统计学习理论的形式化工作中，最危险情形被描述为“不是一个失败的证明，而是一个对错误陈述的成功证明”。

张量网络方面的研究也记录过同类现象：系统给出的证明形式上完全正确，只是它证的那个命题比预期的要弱。

Nielsen写道，OpenAI那份形式化可能把它声称的每一条结论都建立对了。但它没有建立、Lean内核也检查不了的，是这些结论和猜想的原话有没有关系。

人读猜想会看到前提，证明助手拿到一个不满足前提的结论，会照样验证关于它的任何断言。

Connes刚性猜想仍然开放。

论文地址：

<https://philarchive.org/archive/NIEWTCv17>

参考链接：

[1]<https://openai.com/index/ten-advances-in-mathematics/>

[2]<https://github.com/openai/ten-proofs/blob/main/ConnesRigidity.lean>

文章来源：量子位。