Alice & Bob · 23 September 2026

AI and research

riding dragons

Antoine Tilloy



Phase transition

LLM: an augmented database

GPT-3.5 (2022): fun, but confabulates by extrapolation

LLM: an augmented database

GPT-3.5 (2022): fun, but confabulates by extrapolation

- **Who is Pierre Rouchon?** → decent answer

LLM: an augmented database

GPT-3.5 (2022): fun, but confabulates by extrapolation

- **Who is Pierre Rouchon?** $\longrightarrow$ decent answer
- **Who is Antoine Tilloy?** $\longrightarrow$ [...] a physicist, PhD student of Pierre Rouchon, who worked on [...]

LLM: an augmented database

GPT-3.5 (2022): fun, but confabulates by extrapolation

- **Who is Pierre Rouchon?** ⟶ decent answer
- **Who is Antoine Tilloy?** ⟶ [...] a physicist, PhD student of Pierre Rouchon, who worked on [...]

made-up citations, elementary computations wrong, elementary logic wrong ⟶ stochastic parrot?

LLM: solves hard problems

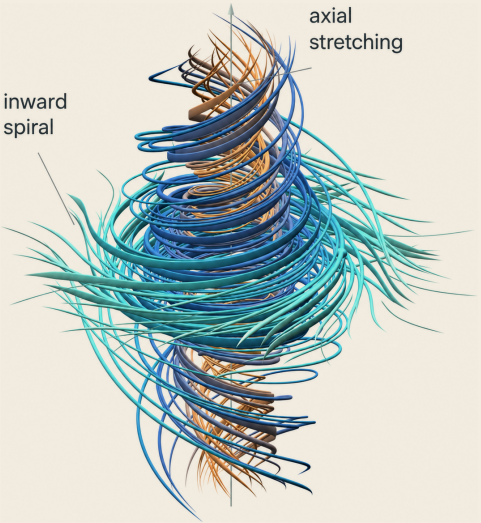
8 September 2026: an internal OpenAI model (GPT-7?) proves **finite-time blow-up** for Navier–Stokes *with a smooth force*

→ statement (C) of the Clay problem

global regularity at $f \equiv 0$ still open

166 pages, verified in Lean

$\sim 10^4$ concurrent agents



Models in a nutshell

LLMs, tokens, context

Token

The alphabet of the LLM $\sim 10^5$ distinct words/syllables

`the` $\longrightarrow$ `the` `and` $\longrightarrow$ `and`

LLMs, tokens, context

Token

The alphabet of the LLM $\sim 10^5$ distinct words/syllables

`the` $\longrightarrow$ `the` `and` $\longrightarrow$ `and`

`tokenization` $\longrightarrow$ `token|ization` `Alice & Bob` $\longrightarrow$ `Alice|  &|  Bob`

LLMs, tokens, context

Token

The alphabet of the LLM $\sim 10^5$ distinct words/syllables

`the` $\longrightarrow$ `the` `and` $\longrightarrow$ `and`

`tokenization` $\longrightarrow$ `token|ization` `Alice & Bob` $\longrightarrow$ `Alice|␣&|␣Bob`

`anticonstitutionnellement` $\longrightarrow$ `ant|icon|stitution|nel|lement`

LLMs, tokens, context

Token

The alphabet of the LLM $\sim 10^5$ distinct words/syllables

`the` $\longrightarrow$ `the` `and` $\longrightarrow$ `and`

`tokenization` $\longrightarrow$ `token|ization` `Alice & Bob` $\longrightarrow$ `Alice| &| Bob`

`anticonstitutionnellement` $\longrightarrow$ `ant|icon|stitution|nel|lement`

`Théau Perronin` $\longrightarrow$ `Th|é|au| Perr|on|in`

LLMs, tokens, context

Token

The alphabet of the LLM $\sim 10^5$ distinct words/syllables

`the` $\longrightarrow$ `the` `and` $\longrightarrow$ `and`

`tokenization` $\longrightarrow$ `token|ization` `Alice & Bob` $\longrightarrow$ `Alice|␣&|␣Bob`

`anticonstitutionnellement` $\longrightarrow$ `ant|icon|stitution|nel|lement`

`Théau Perronin` $\longrightarrow$ `Th|é|au|␣Perr|on|in`

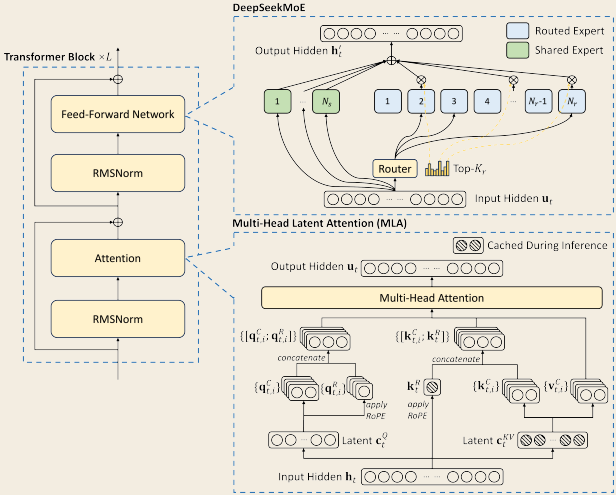
Large Language Model

A function of N tokens (the context) returning token $N+1$

$$f_\theta : (t_1, \ldots, t_N) \longmapsto t_{N+1}$$

`Deepseek V4 Pro`: $N_{\text{max}} = 10^6$, $\dim(\theta) = 1.6 \cdot 10^{12}$

Attention is all you need $f_\theta(t_1, \cdots, t_N)$



DeepSeek-V3 architecture

Pre-training

Gradient descent on every text humankind ever wrote
[$10G$ for `GPT2` $\longrightarrow$ $32T$ tokens for `Deepseek V4 Pro`]

Pre-training

Gradient descent on every text humankind ever wrote
[$10G$ for `GPT2` $\longrightarrow$ $32T$ tokens for `Deepseek V4 Pro`]

Result:

- **prompt**: “The capital of France is”
- **answer**: “Paris, a city then in full expansion, which Louis XIII wished to make shine [...]”

Pre-training

Gradient descent on every text humankind ever wrote
[$10G$ for `GPT2` $\longrightarrow$ $32T$ tokens for `Deepseek V4 Pro`]

Result:

- **prompt**: “The capital of France is”
- **answer**: “Paris, a city then in full expansion, which Louis XIII wished to make shine [...]”

- **prompt**: “Who is Pierre Guilmin?”

Pre-training

Gradient descent on every text humankind ever wrote
[$10G$ for `GPT2` $\longrightarrow$ $32T$ tokens for `Deepseek V4 Pro`]

Result:

- **prompt**: “The capital of France is”
- **answer**: “Paris, a city then in full expansion, which Louis XIII wished to make shine [...]”

- **prompt**: “Who is Pierre Guilmin?”
- **answer**: “asked the new intern”

Supervised fine-tuning

Gradient descent on carefully chosen examples of the form:

- user question
- expected answer

[$1.5M$ examples for `Deepseek V3`]

Supervised fine-tuning

Gradient descent on carefully chosen examples of the form:

- user question
- expected answer

[$1.5M$ examples for `Deepseek V3`]

Result:

- **prompt**: “Who is Pierre Guilmin?”
- **answer**: “A researcher at Alice & Bob who sailed through the Atlantic ocean.”

Reinforcement learning from verifiable rewards – RLVR

In mathematics and code, the result can be checked.

We reinforce on $\sim 1M$ problems, of increasing horizon:

- the elementary computation $3 + 6$ is correct
- the code of a simple Python function compiles
- the simplification of an integral is correct
- the code compiles *and* gives the right result
- a very general mathematical question is correctly solved, with several steps and intermediate pieces of code

$\longrightarrow$ emergence of reasoning and strong generalization

Reinforcement learning with human feedback – RLHF

The model is told that one answer is preferred over another

Reinforcement learning with human feedback – RLHF

The model is told that one answer is preferred over another

Example: *“What is the best division at Alice & Bob?”*

Reinforcement learning with human feedback – RLHF

The model is told that one answer is preferred over another

Example: *“What is the best division at Alice & Bob?”*

A Firmware

Reinforcement learning with human feedback – RLHF

The model is told that one answer is preferred over another

Example: *“What is the best division at Alice & Bob?”*

A Firmware

B Platform

Gives the model its “personality”

Reinforcement learning with human feedback – RLHF

The model is told that one answer is preferred over another

Example: *“What is the best division at Alice & Bob?”*

A Firmware

B Platform

Gives the model its “personality”

More and more **RLAF** $\longrightarrow$ artificial intelligence feedback

Recent developments

- **Harness**: the model is given an environment (the terminal) and arms (bash, Python, internet search, APIs) + **RLVR**

  Claude Code, Codex, Antigravity, Grok Build, Pi, OpenCode, Hermes, OpenClaw

Recent developments

- **Harness**: the model is given an environment (the terminal) and arms (bash, Python, internet search, APIs) + **RLVR**
  Claude Code, Codex, Antigravity, Grok Build, Pi, OpenCode, Hermes, OpenClaw
- **Subagents**: the LLM writes the prompt of another LLM, and reads part of its output $\longrightarrow$ keeps the context reasonable

Recent developments

- **Harness**: the model is given an environment (the terminal) and arms (bash, Python, internet search, APIs) + **RLVR**
  Claude Code, Codex, Antigravity, Grok Build, Pi, OpenCode, Hermes, OpenClaw
- **Subagents**: the LLM writes the prompt of another LLM, and reads part of its output → keeps the context reasonable
- **Skills**: recipes (in natural language) explaining how to do something

Recent developments

- **Harness**: the model is given an environment (the terminal) and arms (bash, Python, internet search, APIs) + **RLVR**
  Claude Code, Codex, Antigravity, Grok Build, Pi, OpenCode, Hermes, OpenClaw
- **Subagents**: the LLM writes the prompt of another LLM, and reads part of its output → keeps the context reasonable
- **Skills**: recipes (in natural language) explaining how to do something
- **Grounding**: internet search, literature databases, your own files

Recent developments

- **Harness**: the model is given an environment (the terminal) and arms (bash, Python, internet search, APIs) + **RLVR**
  Claude Code, Codex, Antigravity, Grok Build, Pi, OpenCode, Hermes, OpenClaw
- **Subagents**: the LLM writes the prompt of another LLM, and reads part of its output → keeps the context reasonable
- **Skills**: recipes (in natural language) explaining how to do something
- **Grounding**: internet search, literature databases, your own files
- **Chain of thought**: thinking before answering → *inference* compute buys quality too

Recent developments

- **Harness**: the model is given an environment (the terminal) and arms (bash, Python, internet search, APIs) + **RLVR**
  Claude Code, Codex, Antigravity, Grok Build, Pi, OpenCode, Hermes, OpenClaw
- **Subagents**: the LLM writes the prompt of another LLM, and reads part of its output → keeps the context reasonable
- **Skills**: recipes (in natural language) explaining how to do something
- **Grounding**: internet search, literature databases, your own files
- **Chain of thought**: thinking before answering → *inference* compute buys quality too
- **RL environments written by AI** → proto-**RSI**

Where this leaves us

Models now **reason**.

Where this leaves us

Models now **reason**.

We are no longer bottlenecked by data, only by compute.

Where this leaves us

Models now **reason**.

We are no longer bottlenecked by data, only by compute.

It will not stop.

Mundane uses of (Opus class) models

Knowing whether an idea is original

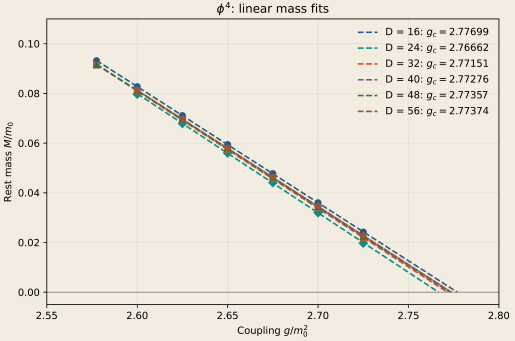
Example:

> *I would like to control reactivity in a nuclear reactor by pumping the moderator (typically D_2O) quickly, which creates an entrainment effect in the neutron flux. I think this is not negligible if there are asymmetries. Has this been studied? Do a thorough search, compile every source, and write me a complete report on the subject.*

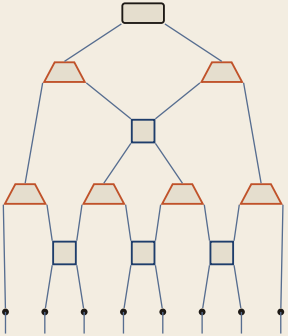
Result: a 12-page report, explaining 5 related ideas, with 2 non-obvious citations (Soviet literature from the 1960s)

Spending 1 day on Google Scholar is forbidden now

Making figures



the plot of mass as a function of g for various D in my RCMPSCore.jl folder



a tikz diagram of a MERA tensor network

How I got the previous slide

prompt

> *Now add a slide with “Making figures” in which I want you to put two figures side by side.*
>
> *The first is a figure showing “the plot of mass as a function of g for various D in my RCMPSCore.jl folder” and the legend says exactly this. Then “a tikz diagram of a MERA tensor network” and the legend says exactly that.*

Optimizing big codes and libraries

Producing optimized, possibly disposable code

Optimizing big codes and libraries

Producing optimized, possibly disposable code

Example:

> *`MPSKit.jl` is slow on the example in this folder, and I see the CPU is never maxed out. I think it is a very large "MPO bond dimension" thing (which is unusual). Can you read the code of this library, the docs, profile everything, and suggest changes to make it faster on my machine for this class of examples?*

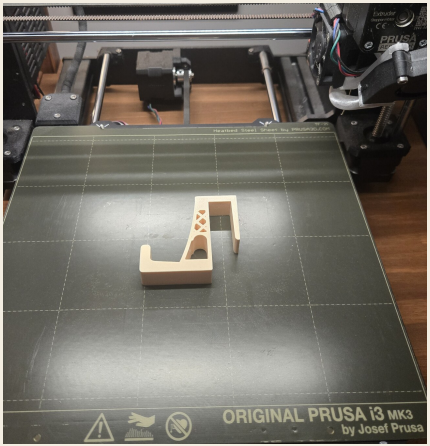
new code 60 times faster, validated against the initial one

Previously crazy coding projects

- Write an LLM inference engine to stream Kimi K3 from SSD
- Rewrite SymPy in Rust (Hector)

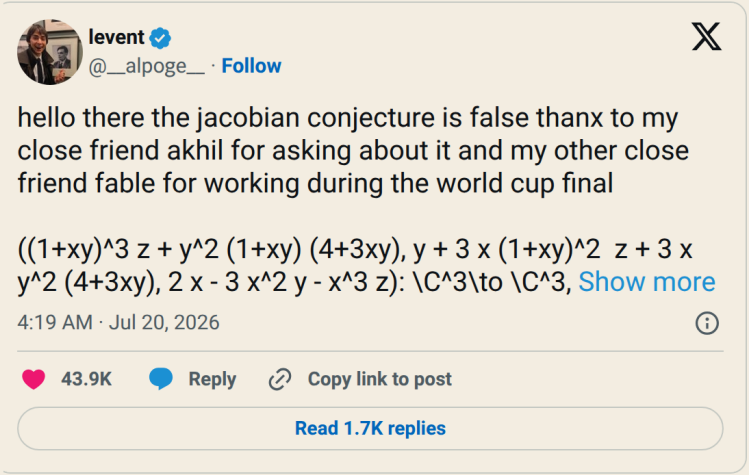
Real world project

I would like to design a hook that I could 3d print, with an original design (maybe something a little hollow like the eiffel tower, that seems optimized a lil for resistance). It has to hold on a vertical plate, the horizontal width is 15.4mm. Can you design it (either directly, or using FreeCAD installed on this computer, or whatever you want)? Then if I agree, you can slice it for PETG (maybe on my local prusa slicer install?). Then on the network I have mainsail at zero.local address, on a custom prusa mk3.5



Beyond human intelligence, Fable, Astra, and the internal monsters

The Jacobian conjecture, 20 July 2026



levent
@__alpoge__ · Follow

hello there the jacobian conjecture is false thanx to my close friend akhil for asking about it and my other close friend fable for working during the world cup final

((1+xy)^3 z + y^2 (1+xy) (4+3xy), y + 3 x (1+xy)^2 z + 3 x y^2 (4+3xy), 2 x - 3 x^2 y - x^3 z): \C^3\to \C^3, Show more

4:19 AM · Jul 20, 2026

43.9K Reply Copy link to post

Read 1.7K replies

Ten open problems, 1 August 2026

1. **High-dimensional sphere packing** — down to the Cohn–Elkies threshold
2. **Binary and spherical codes** — exponentially improved bounds
3. **Non-sofic groups** — existence, open since Gromov 1999
4. **Connes's rigidity conjecture** — disproved
5. **Arithmetic circuit complexity** — permanent, formula bound $n^4/\log n$
6. **Quantum parallel repetition** — two-player quantum games
7. **Closest vector problem** — hardness of approximation, lattice crypto
8. **Ehrhart's volume conjecture** — in every dimension
9. **Multicolor Ramsey numbers** — superexponential bound, Erdős 183
10. **Extremal number conjectures** — compactness and degeneracy

$\theta(p_c) = 0$



It now seems only a matter of time before the most famous conjecture in our field, $\theta(p_c) = 0$, also falls to the bulldozers.

Hugo Duminil-Copin, 30 August 2026

anthropics / **formal-math** Public

Notifications | Fork 45 | Star 241

θ(p_c) = 0 for Bernoulli bond percolation on ℤ^d in all dimensions d ≥ 2, in Lean 4 / Mathlib

247 Lean files, 86 892 lines

Millennium problem

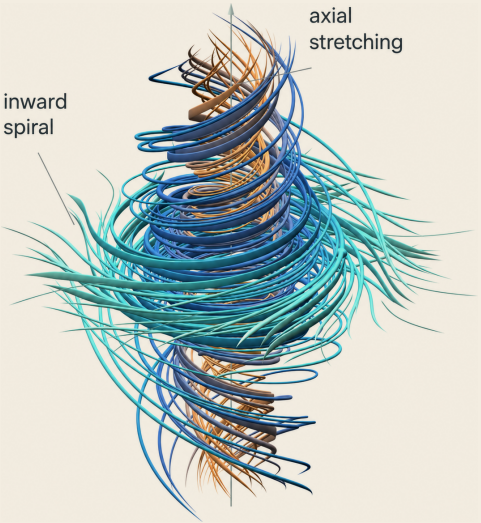
8 September 2026: an internal OpenAI model (GPT-7?) proves **finite-time blow-up** for Navier–Stokes *with a smooth force*

→ statement (C) of the Clay problem

global regularity at $f \equiv 0$ still open

166 pages, verified in Lean

$\sim 10^4$ concurrent agents



Personal examples

- Nuclear reactor with flow — `Fable`, subtle perturbative calculations
  stuck for 6 months → full perturbative calculation, sims, plots, perfect
- TDVP for RCMPS — `Astra`, hard tensor network calculations
  tried for years → one shot solution, sims, plots, perfect
- Foundations of quantum mechanics — `Astra`, remarkably subtle
  had the idea for years → killed with one surprisingly pragmatic argument

It won’t stop

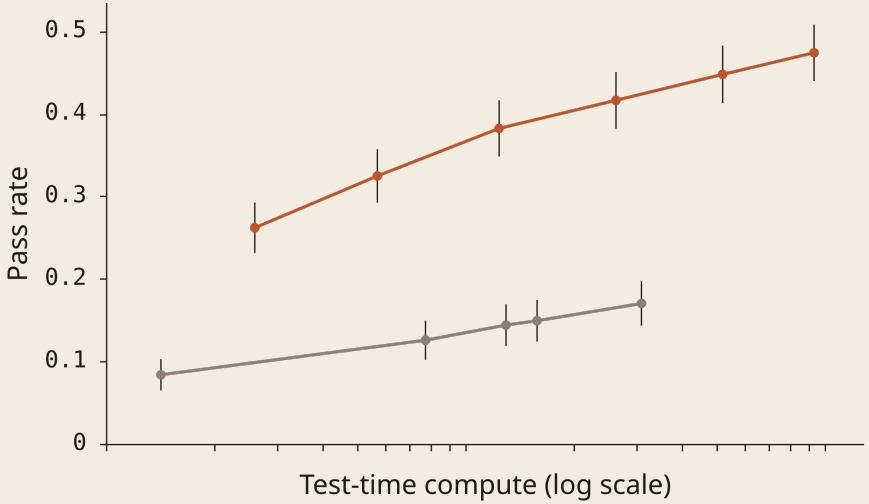


Colossus II (xAI)



Stargate (OpenAI)

Companies have hidden monsters



What should we do

The old ways are already gone

- Denial

  “It doesn’t really *think*, it’s just bruteforce, works on a few problems”

- Anger

  “OpenAI is desecrating mathematics!! They need to be stopped”

- Bargaining

  “Probably the technical things *I* do will remain difficult”

- Depression

  “We’ll never be able to ponder a problem for years…”

- Acceptance

  “We should ride the dragon”

Riding the dragon



Game of Thrones, HBO

- The dragon may double in size every 3 months
- The dragon may ride itself at some point
- So far, few can ride it

Letting go is necessary to do great things

Before

- Coding in assembly
- Computing integrals with manual Gauss–Legendre quadratures
- Extracting square roots with Newton's algorithm

Now

- Coding (all of it)
- Proofs (those technical/long)
- Calculations (those technical/not insightful)

Paradoxically easier for slightly older researchers

What remains not solved (yet)

qualities of a good dragon rider

- Research “taste” (what is the right question)
- Extracting theories / concepts
- Going out of tunnel vision

General advice

Anxiety is *normal*

- Look at the beast in the eyes: use the best models regularly, excessively (information matters)
- Destroy your own problems with it, don't let ego in the way
- Focus on the cool stuff that is now possible rather than on what's gone
- Make humanity proud! $\longrightarrow$ build a quantum computer!

Concrete battle plan

Token maxing

- Use the best models (`Astra`/`Fable`) in high/xhigh effort modes
  [today's `Fable` is tomorrow's `Sonnet`]
- Waste the tokens on tasks that are too difficult
- Get access to computing power

Advice for frontier models and harnesses in research

- Try to do the thing you are usually doing, but *via* the model
- What worked barely once will work reliably after
- Let it cook (forget super custom instructions, homemade skills, etc.)
- Try no hands (VS Code / terminal closed for a week!)

Find balance

- AI-enhanced research pushes are exhausting and humiliating at first
- Sleep well / do sports / talk to your friends (or to a psychologist)

Maintain a community

- Ask a strong AI together, and discuss the output live
- Do a journal club to discuss some poorly written AI breakthrough

Summary

- Models are strong, already super-human in parts of maths
- It won’t stop or plateau soon
- Grieve fast, quickly forget what’s gone anyway
- Ride the dragon
- Enjoy the crazy acceleration to come