How agentic AI transformed my way of working

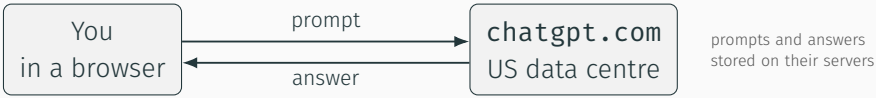
No cloud, no big model, with receipts.

Anthony Scemama

November 19, 2026

LCPQ — Theo, Toulouse

The simplest example

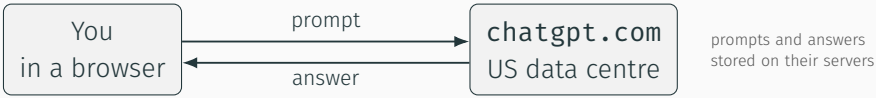


The model is not on your machine. It runs there.

OpenAI, How OpenAI handles data in consumer services (help.openai.com)

LLM: **L**arge **L**anguage **M**odel

The simplest example



The model is not on your machine. It runs there.

OpenAI, How OpenAI handles data in consumer services (help.openai.com)



A function: text in, text out. It predicts the *next token*, appends it to the prompt, and starts again.

LLM: **L**arge **L**anguage **M**odel

Then the models grew

You: What is 1234×5678?

ChatGPT (2022): 7 002 652 **true:** 7 006 652

reconstruction of the classic failure, not a transcript

It predicted a plausible number.
It had no calculator.

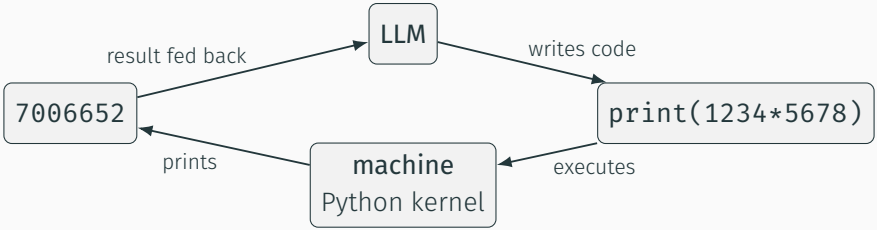


A **stochastic parrot**

Bender, Gebru, McMillan-Major & Shmitchell,

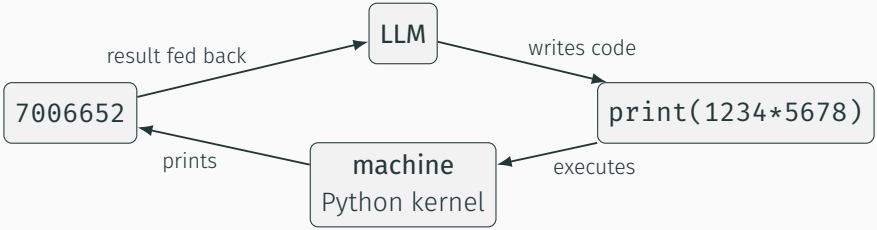
FAccT 2021

Then the models grew



The model writes code, a machine runs it, the result comes back — and it continues.

Then the models grew



30 Nov 2022 text in/out · **6 Jul 2023** runs code · **25 Sep 2023** sees images · **16 Oct 2023** draws images

dates: OpenAI, ChatGPT release notes

All of this in a browser tab (one service, one API, one private company).

The model learned to use tools.

By August 2026 it was reading my papers, and finding bugs in my code.

Then the models grew

> “Given both the competitive landscape and the safety implications of large-scale models like GPT-4, this report contains no further details about the architecture (**including model size**), hardware, training compute, dataset construction, training method, or similar.”
>
> GPT-4 Technical Report, §2 (2023)

Not an oversight: a policy.

How big is the model you talk to?

OpenAI — GPT-4, 5, 5.5 ?

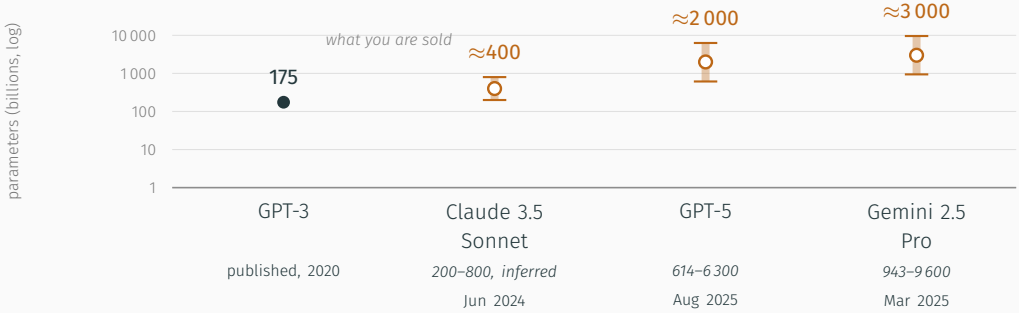
Anthropic — Haiku, Sonnet, Opus ?

Google — Gemini Pro, Ultra ?

OpenAI’s last published size is GPT-3: 175 B, in 2020.

The numbers I can check are the open ones.

Then the models grew



Orange: estimated from probes (IKP v2, low confidence), not from the labs.

If 2 000 B is right: one copy of the weights is 2 TB in 8-bit, so **≈25 H100-80 GB** cards — before the cache, before the second copy.
How many of those are running, and where, is not published either.

domain.rs

RS

In the attached file, when I have a domain with multiple determinants, sometimes I get
$\sum_J \sum_K \langle I | H^+ - E_L^+ | K \rangle (-\langle K | H^+|J \rangle) \ne 1$
in function `draw_multiple_determinants_one_step` .
From my understanding, this sum should correspond to $\sum_J P(I \rightarrow J)$, so it should be equal to one. Am I right?
If I am, can you see where the bug is?

A lu un fichier, exécuté 2 commandes ›

Yes, your understanding is correct

The quantity you wrote is (using your domain's matrix $A \equiv$ `h_plus_mat_in` $= H^+_{DD} -$ diag(E^+_{loc}), restricted to the internal space) really

$$\mathbb{P}_{I\to J} = \sum_{K\in\mathcal{D}} \langle I|A^{-1}|K\rangle \left(-\langle K|H^+|J\rangle\right) \frac{\langle J|\Psi^+\rangle}{\langle I|\Psi^+\rangle}$$

and $\sum_J \mathbb{P}_{I\to J} = 1$ **exactly**, independent of the domain size. This comes straight from the definition of E^+_{loc} (built by summing $H^+\Psi^+$ over the *full* connected space, in `build_eloc_plus_general`): for $k \in \mathcal{D}$,

$$(A\,\Psi^+_{\mathcal{D}})_k = \sum_{l\in\mathcal{D}} H^+_{kl}\Psi^+_l - E^+_{loc}(k)\Psi^+_k = -\sum_{j\notin\mathcal{D}} H^+_{kj}\Psi^+_j\,.$$

Contracting with row 0 of A^{-1} (your `h_plus_inv_row` , since the reference is always index 0 in `in_space`) gives $\langle I|A^{-1}A|\Psi^+_{\mathcal{D}}\rangle = \Psi^+(I)$, i.e. exactly your identity divided by $\Psi^+(I)$. So the bug you're seeing is real.

Where it is

Écrivez un message…

Claude est une IA et peut commettre des erreurs. Veuillez vérifier les réponses.

Sonnet 5 Extra

Where it is

In `draw_multiple_determinants_one_step`:

```rust
let p = i_H_inv_k * self.h_plus(k, &j) * psi_plus.at(&j);
if p > 0.0 {                    // <-- BUG
    if det.contains_key(&j) {
        let new_p = p + det.get(&j).unwrap();
        det.insert(j, new_p);
    } else {
        det.insert(j, p);
    }
    norm += p;
}
```

`h_plus_inv_row[a] = \langle I|A^{-1}|K_a\rangle` is **not sign-definite** for $a \neq 0$ — the code elsewhere knows this perfectly well: `build_proba_in` stores `h_plus_inv_row[a]` with no sign filter, and `draw_determinant_two_step` / `draw_determinant_uniform2` explicitly sample with `WeightedIndex::new(w.abs())` and flip a `sign` flag when `w[idx] < 0.0`. Since $-H^+_{KJ} = |H_{KJ}| \geq 0$ and $\Psi^+(J) \geq 0$, the sign of each term p_{KJ} is entirely inherited from the sign of $\langle I|A^{-1}|K\rangle$. So when that inverse-row element is negative for some internal K, **every** external J connected to K gets a negative p_{KJ} contribution.

The `if p > 0.0` test discards these term-by-term, i.e. *before* the contributions from all K reaching the same J have been summed. But the non-negativity/normalization identity only holds for the **combined** $\sum_K p_{KJ}$ per J, not per (K, J) pair — exactly as `proba()` asserts (`assert!(result >= 0.0)`) only after summing over K. Dropping negative pieces early removes genuine negative flux that was supposed to cancel against positive contributions from other K's, which biases both `norm` and the `det` weights.

This is invisible for a single-determinant domain: there `K=I` is the only internal determinant, so each J gets exactly one term, which by `proba()` 's guarantee is already non-negative — the filter is then a no-op. It only bites when $\mathcal{D}$ has ≥ 2 determinants and A^{-1} has mixed-sign row entries. Note also that the correctly-written sibling `draw_r` `rminant_one_step` (a few lines above) accumulates unconditionally and even carries `ass` `((1.0 - norm).abs() < 1.0e-8)` — which would have caught this immediately if it had been ported over.

Écrivez un message…

Fix

Accumulate unconditionally, and only filter/clamp *after* combining over K (needed because `WeightedIndex::new` requires non-negative weights):

rust

```rust
pub fn draw_multiple_determinants_one_step(&self, n: usize) -> Vec<(Determinant<N>, f6
    let psi_plus = &self.domain_simulation.psi_plus;
    let space_gen = CISDSpace::new(self.simulation());
    let mut det: FxHashMap<Determinant<N>, f64> = FxHashMap::default();
    let mut norm = 0.0;
    let psi_inv = psi_plus.inv_at(&self.ref_determinant);
    for (a, k) in self.in_space.iter().enumerate() {
        let i_H_inv_k = -self.h_plus_inv_row[a] * psi_inv;
        if i_H_inv_k == 0.0 {
            continue;
        };
        for j in space_gen.iter(k) {
            if !self.determinants_map.contains_key(&j) {
                let p = i_H_inv_k * self.h_plus(k, &j) * psi_plus.at(&j);
                // Accumulate every term, positive or negative: only the
                // total summed over all K for a fixed J is guaranteed >= 0.
                if det.contains_key(&j) {
                    let new_p = p + det.get(&j).unwrap();
                    det.insert(j, new_p);
                } else {
                    det.insert(j, p);
                }
                norm += p;
            }
        }
    }
    println!("{} {}", self.ref_determinant, norm);

    // Now filter: after combining, entr… should be >= 0 up to fp noise.
    let (det_out, w): (Vec<_>, Vec<_>) = det
```

Écrivez un message…

The bug — `if p > 0.0` drops negative terms *before* they are summed over K.

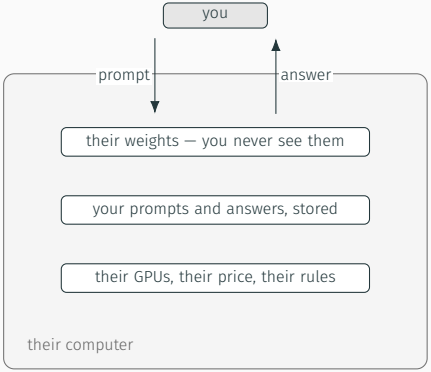
Why that is wrong — the non-negativity identity holds for the combined $\sum_K p_{KJ}$, not for each p_{KJ}: filtering early biases the weights and the norm.

What it had to know — that $\langle I|A^{-1}|K\rangle$ is **not sign-definite**, and where the sign of each term comes from. That is the physics of my domain, not a syntax check.

Claude, August 2026 — my code, my bug.

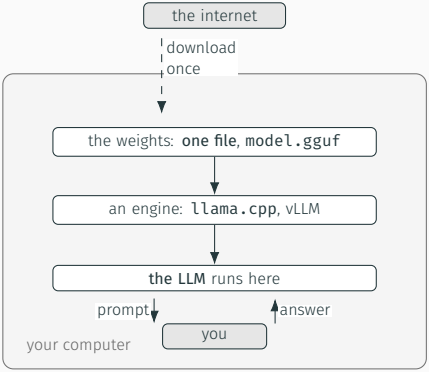
Meanwhile, the weights went public

Rented: behind an API



Your data, on someone else's computer.

Open weights: a file you download



It runs here. Nothing goes out.

Same kind of neural network. The difference: **you hold the file**.

The default setup

- What if access is cut tomorrow?



The default setup

- What if access is cut tomorrow?
- What if prices double?



The default setup

- What if access is cut tomorrow?
- What if prices double?
- What if your private data goes through their servers?

Three reasons to want the model at home.



ReLibQC, end of September, Toulouse

- Workshop on reusable QC libraries. Many developers there had already moved to **Claude Code** — and they were doing impressive work.

ReLibQC, end of September, Toulouse

- Workshop on reusable QC libraries. Many developers there had already moved to **Claude Code** — and they were doing impressive work.
- **Susi Lehtola**: two hours of discussion → **17 pull requests** on TREXIO.

ReLibQC, end of September, Toulouse

- Workshop on reusable QC libraries. Many developers there had already moved to **Claude Code** — and they were doing impressive work.
- **Susi Lehtola**: two hours of discussion → **17 pull requests** on TREXIO.
- **Michel**, during the conference, on the Domains QMC project. Two hours: the FullCI matrix of H_4 with its symmetry detected, recipes to build the determinant domains, the convergence rate, curves, tables, tests.

ReLibQC, end of September, Toulouse

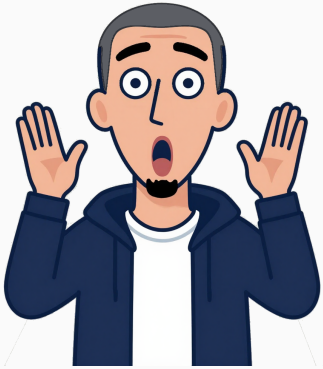
- Workshop on reusable QC libraries. Many developers there had already moved to **Claude Code** — and they were doing impressive work.
- **Susi Lehtola**: two hours of discussion → **17 pull requests** on TREXIO.
- **Michel**, during the conference, on the Domains QMC project. Two hours: the FullCI matrix of H_4 with its symmetry detected, recipes to build the determinant domains, the convergence rate, curves, tables, tests.

"It was doing all my work — science and coding —
and it was better than me at all of them."

my own words, after those two hours.

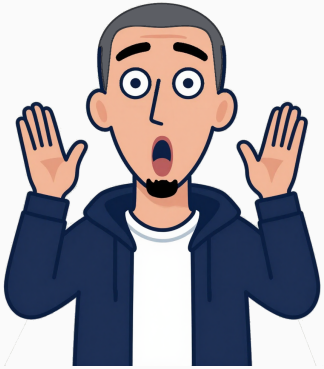
Stunned, then a decision

- **Stunned.** Not a gadget anymore: the floor moving under your feet. Many of us felt it, and are still feeling it.
  After a week, Darwin: **adapt, or disappear**.



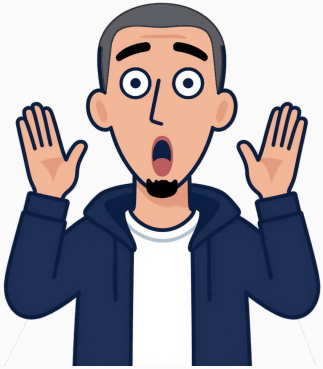
Stunned, then a decision

- **Stunned.** Not a gadget anymore: the floor moving under your feet. Many of us felt it, and are still feeling it.
  After a week, Darwin: **adapt, or disappear**.
- So I set up the equivalent of Claude Code — but **with full control**: my machine + Turpan (CALMIP) for GPUs, open-weight models, my files, my rules, OpenCode.



Stunned, then a decision

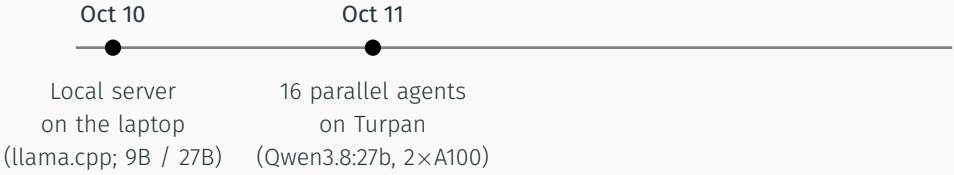
- **Stunned.** Not a gadget anymore: the floor moving under your feet. Many of us felt it, and are still feeling it.
  After a week, Darwin: **adapt, or disappear**.
- So I set up the equivalent of Claude Code — but **with full control**: my machine + Turpan (CALMIP) for GPUs, open-weight models, my files, my rules, OpenCode.
- Discussions with the CALMIP team (Alejandro Estaña, Nicolas Renon) to prepare the future.



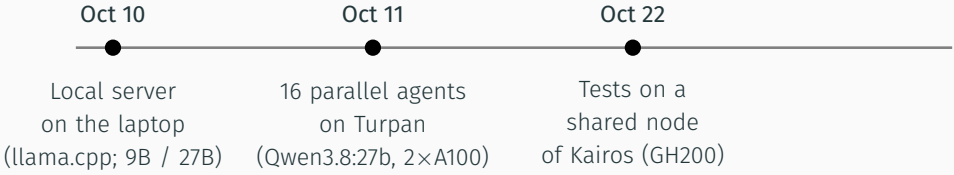
October 2026: the point of no return



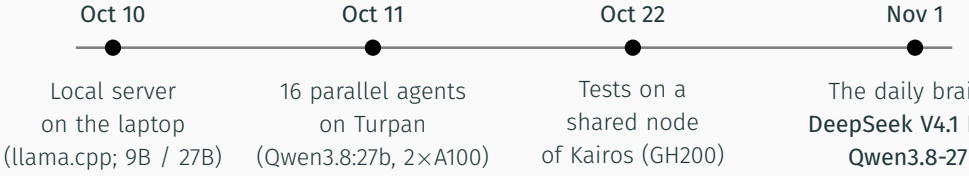
October 2026: the point of no return



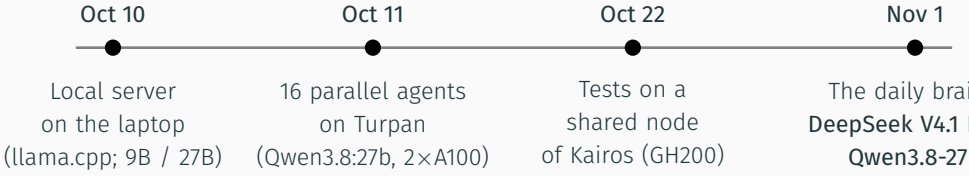
October 2026: the point of no return



October 2026: the point of no return

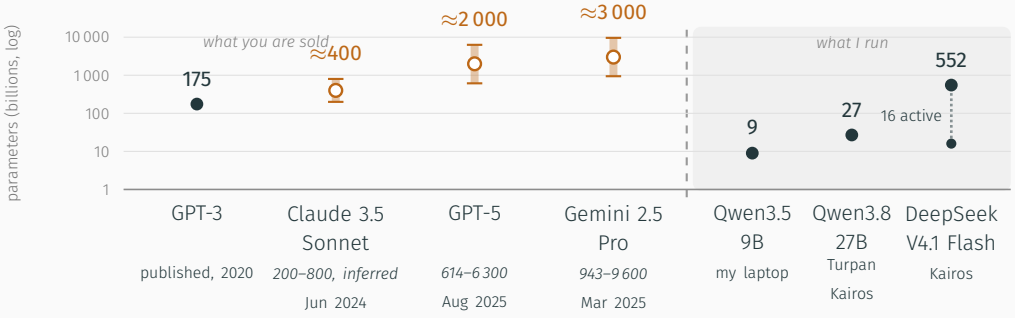


October 2026: the point of no return



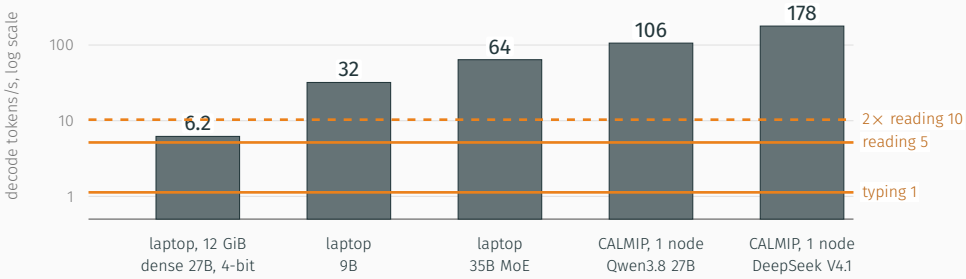
No GPU cluster, no cloud: one 12 GiB laptop + one shared CALMIP node.

October 2026: the point of no return



Orange: estimated from probes (IKP v2, low confidence), not from the labs.

What “fast enough” looks like



My floor for a dialogue: twice my reading speed, about 10 tok/s.

Source: single stream — llama.cpp on a 12 GiB laptop (2026-11-06 A/B) for the first three bars, CALMIP for the last two. Typing: Dhakal et al., CHI 2018 (52 wpm). Reading: Brysbaert 2019 (238 wpm). 1 word ≈ 1.3 tokens.

The agent and the harness

model × tools × context

It is a product, not a sum: a small model with the right context does real work; the best model with no context starts from zero every morning. So the work went into the two terms I own.



model

one 27B I downloaded
the part I swap

when a better one appears



tools

shell, git, SSH, browser
web search, the RAG query

— the hands



context

389 memory files
8 RAGs, 715,171 passages

64 skills

“The model is the least important part: you don’t control it, and they are all strong now. The context is what changes everything.”

Eliott Meunier, 2026

The tools: the ones I already use, plus the ones it writes

Standard tools

shell (`bash`)
read / edit / write a file
`grep`, `glob`
`git`
`ssh` to the clusters
`python` scripts
a browser

web search

Skills: tools it writes for itself

64 on disk. One directory, one `SKILL.md`: when to use it, the exact commands, the traps found the hard way. Four I use every week:

- `slurm` — `sbatch` scripts, partitions, cores and GPUs per job, and how to read `sacct` when a job dies.
- `calmip-kairos`, `calmip-turpan` — the two machines: which module, which queue, where the files belong.
- `gpu-hpc` — twenty worked CUDA/HIP/Kokkos examples, each measured, with the traps that cost a day each.
- `running-cfour` — *the deck must be named `ZMAT`, or the code exits 0 silently.* Found the expensive way.

Written once, by the agent, and reused by every later session.

The context: learn from our interactions

The file, loaded at the start of every session

`skills/opencode-session-context/SKILL.md`

```
## Memory
- Shared memory: $HOME/OpenCode/Memory/shared.md. Read it IN FULL at the
  start of a task; it is your continuity.

## Writing a skill
- When you have learned reusable, verified procedural knowledge --- a workflow,
  a pattern, a cluster of traps, a checklist that caught real errors, a document
  structure that worked: anything you would otherwise rediscover expensively
  --- create or update a skill at $HOME/.opencode/skills/<id>/SKILL.md.
```

Two stores, two speeds: facts go to the memory, procedures go to a skill. Both are files — I can read them, correct them, and put them in git.

The memory: 389 files, 11 MB, written by the agents

`Memory/`	389 files, 11 MB; `shared.md` read in full at every start
`|-- shared.md`	policy, rules, index: the continuity
`|-- profile.md`	who I am, facts that never change
`|-- casquettes/`	five roles: research, HPC software, teaching, music, workflow
`|-- projets/`	28 projects: state, goals, verified traps
`|-- People/`	149 collaborators, every fact tagged by source
`|-- machines/`	laptop, macmini, ampere, kairos, turpan
`|-- archive/`	cold dated logs + my Claude and Gemini exports
`` `-- macmini/ ``	the always-on Mini's inbox, merged then drained

The safeguard: never invent a fact. Facts carry a source tag (`[user]`, `[file]`, `[web]`); inferred ones are refused, and an empty section stays empty.

A RAG is not a model: nothing is trained

- **RAG** = retrieval-augmented generation. **No training, no fine-tuning**: the weights never change.
- My corpus is cut into passages and indexed on my disk. Three steps:
  1. **index** — a full-text index (BM25), rebuilt when the corpus changes;
  2. **retrieve** — the passages that match, *with their source file and page*;
  3. **generate** — those passages go into the context, and the model answers *from them*.
- **Why it reduces hallucination:** from memory, a model has to *invent* the reference — precisely what it does best. Here it must cite a passage; no citation, no claim.
- **It does not remove the failure, it localises it:** every passage comes back with a file and a section, so a wrong answer tells you whether the corpus was missing the fact or the model misread one it had.

A card catalogue, not a brain.

A real query, and what came back

my laptop: 0.07 s, and no model has run yet

```
$ python3 scemama-rag/scemama_rag.py query
  "linear extrapolation fitting function second order energy" -k 3
```

- `Articles/2024/cap_cipsi/Manuscrit/supp_inf_cap_cipsi.tex §CO-`
  "…extrapolations of the CAP-CIPSI stationary energy as a function of the second-order energy are performed using weighted linear fits …" The standard error of the intercept is

$$\mathrm{SE}_{\text{CAP-exFCI}} = \sqrt{\frac{\sum_{m=1}^{M} w_m (y_m - \hat{y}_m)^2}{M-2} \, (\mathbf{X}^T \cdot \mathbf{W} \cdot \mathbf{X})^{-1}_{1,1}}$$

- `Articles/2021/Ec/Manuscript/Ec.tex §CIPSI estimates`
  "…via a weighted five-point linear fit …"
- `Articles/2024/CI_compactification/arxiv/main.tex §Selected Configuration Interaction Algorithm`
  "…one can approximate the TC-FCI energy by $E_{\text{TC-FCI}} \approx E_{TC}^{(0)} + E_{TC}^{(2)}$, the TC-FCI limit being effectively and rigorously reached when $E_{TC}^{(2)} = 0$."

The answer is a pointer, not a memory.

Eight knowledge bases on my disk

corpus	what is inside	docs	passages
`scemama-rag`	my papers, talks, lectures, reports	365	12,338
`refs-rag`	every reference cited in my papers (2,565 refs)	4,803	26,048
`books-rag`	books: maths, physics, CS, quantum chemistry	**1,122**	657,104
`gpu-hpc`	CUDA, HIP, Kokkos, Vulkan, NCCL	368	11,182
`rust-hpc`	Rust for HPC	134	2,152
`numerical-precision`	float32/64, reproducibility	132	2,533
`autotools-cmake`	autoconf, automake, libtool, CMake	346	3,343
`trex2-rag`	the TREX2 project (for the reports)	471	471
	8 knowledge bases	**7,741**	**715,171**

1.9 GB of indexes. Ask about a method, get the passages that discuss it — including the five that disagree.

“Learn how to program GPUs”

the prompt, 21 October 2026, verbatim

“I would like you to learn how to program efficiently GPUs for HPC and quantum chemistry. Create a new directory in ~/OpenCode to do this work. Elaborate a plan, downloading multiple pdf courses and documentations of trusted quality sources, and you can create a RAG for this documentation and write many examples of common templates.”

- **73 sources**, sha256-verified → 368 documents, 11,182 passages, **11 topic notes**.
- **20 worked examples**, measured on my GPU: axpy, VMC walkers, NCCL, Kokkos, Vulkan, cudarc.
  Skill `gpu-hpc`, run on the local 27B — it writes the GPU code, I check.
- And two weeks later, the same shape again — for `profiling`. Next slide.

“Learn how to profile code”

the prompt, 4 November 2026, verbatim

“I would like you to learn how to efficiently profile code (Fortran, Rust, C, C++, OCaml, Python). The tools I like are MAQAO and perf. I would like also that you know how to profile memory allocations, which is a common bottleneck in ocaml and rust codes, and how to fix them.”

Twenty hours later:

- perf (record, stat, c2c) · MAQAO LProf, CQA · heaptrack, dhat-rs · statmemprof, memtrace · py-spy, memray.
- **Every recipe measured on this laptop** before being written down — no untested line.
- It also found a real bug: MAQAO `ONE View -c` mis-parsed the config.
- Distilled into skill `profiling` — used all month on the codes I was fixing.
- Also the clusters: `sbatch`, Kairos, Turpan — which partition, which cores, which GPUs.

One month

“Work on IRPF90”

the prompt, 28 October 2026, verbatim

“I would like you to go through the irpf90 code and look for bugs. While you do so, I would like you to add many comments to explain what the code is doing. You should also improve the speed of the code generation when possible. To check that the code generator does not change, you can apply the modified IRPF90 to the QP2 codebase.”

IRPF90 generates the Fortran of QMC=Chem and QP2: `.irp.f` in, modules out.

What came out of it:

- **2.02× faster** on QP2: 122.4 s → 60.6 s, with **44/44 outputs byte-identical**.
- **13,366 ifort warnings, 58 names**: over 80 characters, silently truncated; short names + CRC now.
- Six releases in the month: v2.1.0 → v2.3.4.

Source: ***`Sessions/2026-10-28 -- IRPF90 codegen bug review and speedup.md`***.

Then it could read QP2: `qp rag`

- It finished the QP2 documentation first, then learned the code. The rule in its memory: *query the RAG first, read the code second*.
- `qp rag` indexes the tree once — **770 files**, **4 240** routines and providers — and a query returns the body, the `BEGIN_DOC` and `needs`.

one query, from the transcript — 0.11 s

```
query ``Hamiltonian matrix element between two determinants''  -k 3
1. qp2:entity:h_matrix_cas (determinants/psi_cas.irp.f:113)
  doc: Hamiltonian matrix in the CAS determinant basis:
      H_matrix_cas(i,j) = <psi_cas(i)| H| psi_cas(j)>
  needs: all_mo_integrals big_array_coulomb_integrals mo_integrals_map n_int psi_cas
  needed by: psi_cas_energy psi_coef_cas_diagonalized

  calls: i_h_j     +2 more hits, then the body
```

A question in my words; an answer with a file, a line number, and the name of the routine that does the work — **QP is now a program an AI can be questioned about**.

What it changed in QP2 — and the start of QP3

- **Symmetry, *a posteriori*.** Detection and labelling added to QP2, so an approximately symmetric molecule gets its labels anyway; the design note verified **211/211** character tables.
- **libcint as a selectable engine** (PR #407). Measured on benzene: AO-Cholesky generation **6.6 s vs 10.2 s** (1.54×), full SCF wall 1.36×, energies agreeing to 9×10^{-10} Ha — and the trap it documented: `qp_create_ezfio -q` silently falls back to the built-in engine.
- **QP3 started:** `qp3` became the base branch, the open PRs were retargeted to it, and the first design input is in — GPU semistochastic EN-PT2.

None of this is a rewrite: three changes to a code I maintain, made because for the first time something could answer questions about it.

One prompt, one script

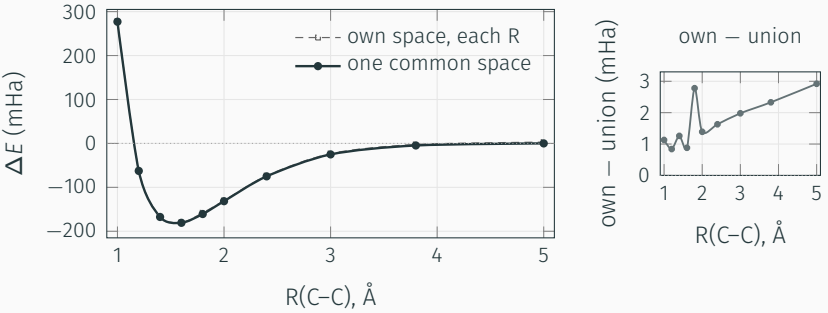
the prompt, 28 October 2026, verbatim (for a collaborator's curves)

"In QP2, I want to make potential energy curves of ground and excited states. … Can you write a script to replace the determinants of N directories given in the command line by the union of determinants of all directories?" … Ultimately, this script will go into `qp2/scripts`*, and its use will be documented with the science-writer agent."*

- No design round, no follow-up: one prompt, and the script was in the tree.
- It then tested itself on a **10-point C_2H_6 curve** (CASSCF(4,4) + CIPSI, aug-cc-pVDZ) — and found **two bugs in its own script**, fixed before committing `a5a3fe0a`.
- Same set of determinants at every geometry → smooth curves, by construction.

The prompt is the whole specification. What I had to know was the physics: that the union is what makes a curve.

One prompt, one script



C_2H_6, CIPSI+PT2, ground state as run: **10 geometries**, **49 413 determinants**.

On the same scale the two curves are one; the right panel is how much the common space moves them — at most 3 mHa.

The science-writer: prose that cannot invent

- An **agent**, not a prompt: it writes papers, reports and one-page step notes from the record — and it runs on the *local* model. No browser, no fetch, no pty: it cannot reach a source it was not given.
- It is chained to two checkers, and either one can block:

the fact-checker

Every checkable claim goes into `claims.tsv` and gets a verdict *with its evidence*. A row without a verdict blocks the review — no silent prose.

the bibliography

Every entry is proven against its own PDF by `bibcheck.py`, with a closed verdict vocabulary. **No PDF, no verification** — the entry is reported, not trusted.

The rule underneath: *do not invent*. What it writes, I check; what it cites, a script has opened.

Science: a Jastrow fixed by a norm

- **Question** (frozen 2026-11-13): can the transcorrelated Jastrow be fixed *deterministically* — by minimizing the first-order wavefunction — and does that give a CI expansion *more compact* than the VMC-optimized Jastrow's?

$$\|\Psi^{(1)}\|^2 = \sum_{I\neq 0} c_I^2, \qquad c_I = \frac{\langle \Phi_I | H_{TC} | \Phi_0 \rangle}{E_0 - E_I}, \qquad \sigma^2_{\mathrm{ref}} = \sum_{I\neq 0} \langle \Phi_I | H_{TC} | \Phi_0 \rangle^2$$

*Which "compact"? The norm, $-E^{(2)}$, or σ^2_{ref} (Filip et al., JCP **163**, 084107, 2025)?*

Science: a Jastrow fixed by a norm

- **Question** (frozen 2026-11-13): can the transcorrelated Jastrow be fixed *deterministically* — by minimizing the first-order wavefunction — and does that give a CI expansion *more compact* than the VMC-optimized Jastrow's?

$$\|\Psi^{(1)}\|^2 = \sum_{I\neq 0} c_I^2, \qquad c_I = \frac{\langle \Phi_I | H_{TC} | \Phi_0 \rangle}{E_0 - E_I}, \qquad \sigma^2_{\mathrm{ref}} = \sum_{I\neq 0} \langle \Phi_I | H_{TC} | \Phi_0 \rangle^2$$

*Which "compact"? The norm, $-E^{(2)}$, or σ^2_{ref} (Filip et al., JCP **163**, 084107, 2025)?*

- **Step 1** found a defect in QP2 core code: a zero-body term counted twice on every TC diagonal $\rightarrow e_{00}$ off by 137 mHa.

Science: a Jastrow fixed by a norm

- **Question** (frozen 2026-11-13): can the transcorrelated Jastrow be fixed *deterministically* — by minimizing the first-order wavefunction — and does that give a CI expansion *more compact* than the VMC-optimized Jastrow's?

$$\|\Psi^{(1)}\|^2 = \sum_{I\neq 0} c_I^2, \qquad c_I = \frac{\langle \Phi_I | H_{TC} | \Phi_0 \rangle}{E_0 - E_I}, \qquad \sigma^2_{\mathrm{ref}} = \sum_{I\neq 0} \langle \Phi_I | H_{TC} | \Phi_0 \rangle^2$$

- **The gradient.** The agent wrote **1,939 lines in three files** for

$$\partial_k F = 2 \sum_{I\neq 0} \frac{V_{I0}\, \partial_k V_{I0}}{\Delta_I^2}$$

  built on QMCkl's `qmckl_get_jastrow_champ_factor_*_gl_pderiv` kernels, which QP2 has never called. Handed over *unbuilt*.
- **The gate**, registered before the code existed: the two-step test on V_{I0} at 10^{-8}. It printed 2.9×10^{-2}, six orders out, and F at 345× step 3. Step 5's relink reproduces step 4 to roundoff, so that 345× is in the gradient: **refuted**.

Science: three iterations

point	$F = \sum_{I \neq 0} c_I^2$	N_{99}	$E_{00} - E_{\mathrm{VMC}}$ (Ha)
`scan_0`	3.907	31	−1.867
iteration 1	0.2792	787	−1.172
iteration 2	0.1444	1035	−0.390
iteration 3	0.0499	1403	−0.279
J_{VMC}	0.00725	1479	+0.0017

- F falls **78**× in three iterations, and the run stops on the **three-iteration cap**, not on convergence: the 1.5× rule never fired ($f_2 = 1.93$, $f_3 = 2.89$).
- N_{99}, the determinants carrying 99 % of the norm, **rises at every step** (31 → 1403) toward J_{VMC}'s 1479. Lowering the norm walks the vector toward the VMC footprint — that is **not** a compaction.
- The gap to E_{VMC} narrows (1.867 → 0.279 Ha) and is not converged; the last iteration costs 238–507 s per evaluation (median 192.5 s).

The norm is retired as the objective there; the compactness question itself is still open.

Then I gave it fifteen years of mail

- **85,010 messages**, 35 folders, back to the oldest in my archive (2008) — read read-only, through the index Thunderbird already keeps. It never wrote to a mailbox.
- Out of that, **`Memory/People/`: 149 files**, one per person — name, role, institution, projects, a dated timeline rather than a snapshot.
- Every fact carries its source: **`[mail]`** (subject + date), **`[file]`** (path or DOI), **`[web]`** (URL), **`[user]`**. **No source, no sentence** — and an unknown stays under "Unknowns" with the question to ask.
- Cross-linked to the 28 project files. So "who works on transcorrelation, and where are they?" is a question with a cited answer, not a guess.

Note what this is not: a model that has read my mail. It is an index over my mail, and a set of files a human can audit line by line.

“Matteo is coming next week”

- I said one sentence. It came back with the project: **TC-SCF vs VMC orbital optimisation**, started in 2025 with **Manu Giner** (LCT, Paris) and **Claudia Filippi** (Twente); Matteo is Manu’s post-doc; one CHAMP Jastrow step, one QP2 transcorrelated orbital step, alternating; Manu’s notes arrived on 20 October.
- None of that was in the session. It was in the people files and the mail.
- Then the instruction: *reconstruct the history of the project, gather all the data we have, and write the article.*
- Three days later: Method with its five equations, Results with **Table I — 24 rows, 9 systems, 7 columns of energies**, a Discussion that measures the gap between the two orbital-optimisation routes. First draft of the abstract was rejected by me, with three wrong statements.

What changed is where the work went: not into typing LaTeX, but into reading the tables and saying no.

The referee is also an agent — and it has no stake

- A separate agent, in a fresh context, with **only the compiled PDF**: no sources, no data notes, no transcript of how the paper was written.
- Its job is the journal referee's: summary, major points, minor points, a recommendation.
- It found real holes — a stopping criterion that was never stated, a spin state that the inputs did not fix. Both were patched, and the patch is a diff in the repo.
- Two of its "findings" were artefacts of extracting text from a two-column PDF. The review loop refuted them with the line of `main.tex` that already said the opposite — *and that refutation is in the report too.*

The point: a second reader that cannot agree out of politeness, and a written record of every disagreement. That is what I am after.

Hodegos: a person who leads the way along a road or journey

Autumn 2026, in mathematics

- **September** — a preprint claiming progress on Navier–Stokes, produced by a swarm of about **10,000** agents, "formalised" in Lean in passing. Hours later, **8,000+** mathematicians had signed a letter objecting to the whole manner of it.
- **6 October** — **722** manuscripts in **372** families, read and sorted by machine. Of the top results, **162 of 722** could be formalised in Lean (**≈22%**) — and only by stating the definitions first.
- **7 October** — **3** manuscripts withdrawn for a sign error; **14** revised.
- **27 October** — one week, two results: a Yang–Mills statement, and hours later an independent disproof of it from another lab.

The problem is not that the claims were false. It is that **nobody could tell** without redoing the work — and the announcing moved faster than the checking.

What the mathematicians say

- **Terence Tao**: the pace is “insane” — and he is the one who built the tools.
- **Tom Sutherland**: treat the claims as *unverified* until a human has checked them; being plausible is not the bar.
- **The American Mathematical Monthly**: “a demonstration of power” — admiration and a warning in one sentence.
- Others, including **Litt**, argue the opposite: the tool finds things we would not have found, and refusing it is the error.
- And **the Rincon open letter** (12 October, circulated publicly): the objection is not to the mathematics, it is to the *spectacle* — benchmarks announced as discoveries, results nobody can reproduce.



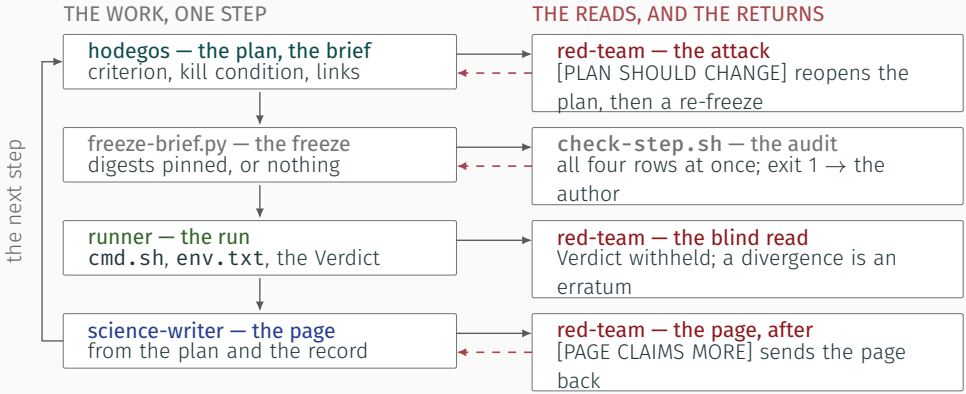
I agree with the critique

- **RogueESR**, on the same news: *“un résultat sans chemin”*; investor stunts on one side, a project of domination on the other.
- I agree. A result with no path is not a result; it is an announcement.
- But the critique is aimed at *how it is being sold*, not at what the machine can do.
- My position: **there is a reasonable way to use this, for humans and for research**.

The problem I actually have

- **The bottleneck moved to verification.** Generation scales; verification does not: one of me, and a machine that writes faster than I can read.
- My first draft of the paper with Matteo had **three wrong statements** in it — each one a plausible paraphrase of something true elsewhere, none of them invented:
  - *"the TC energy is near-variational"*
  - *"the e–N cusp is carried by the Jastrow"*
  - *"one TC-SCF step costs one SCF iteration"*
- **And I would stop understanding.** If the machine does all the work, I lose track of what is happening. I do this job to learn: what I build has to *teach* me, and keep me in the loop.
- So the question is not "how do I publish more papers". It is: **how do I build an agent whose every claim I can check and learn from?**
- The thing I am building is not a better writer. It is a **gate**.

The loop: one step under the contract

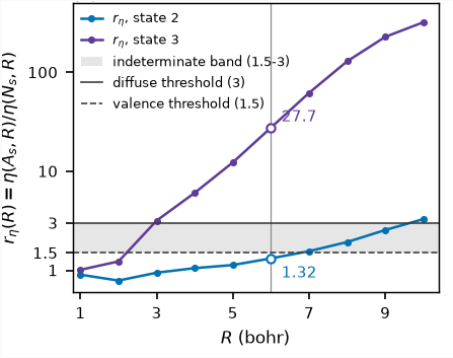


Which leaves: more robust testing of hypotheses, better reproducibility, ideas explored — and the researcher active, and learning.

One step, in plain words

1. **You ask one question.** One question, written down. Not a direction, not a wishlist.
2. **The planner turns it into a test.** It writes what would be measured, what number would count as success, and *what result would kill the idea* — before anything is run.
3. **The freeze closes the plan.** A small script pins a fingerprint of the plan. From then on the plan cannot drift; if it changes, the change is recorded as an appendix and re-pinned, so the transcript shows it.
4. **The runner runs it.** It saves the exact command, the machine, and the raw output, then writes one word: pass, fail, or undecided. No commentary.
5. **The writer writes the page.** From the plan and the record only. Every number carries a tag to the file and line it came from.
6. **The readers look at it.** Before the run, one attacks the plan; after the run, one checks the page against the record; one reads the pass/fail word blind, without seeing the declared answer; and a checker reads all the files of the step at once. Any of them can send work back.
7. **The page becomes an input of the next step.** The left-hand arrow: the next plan must link this page, so you build on what is written, not on what you remember.

One real step: are we comparing the same state?



state 2 is our S_1, state 3 our S_2; the band and the bars are the rule frozen before the run.

The question. In a 56 125-determinant three-state CIPSI wave function: is our S_2 (6.889 eV) the $n \to 3s$ Rydberg state, and our S_1 (4.446 eV) the valence $n \to \pi^*$?

The test, fixed in the plan. Diagonalise the difference density $D_s = \rho_s - \rho_1$: acceptor A_s (density appears), donor N_s (density leaves); $|\varphi|^2$ the density of φ, $\eta(\varphi)$ its fraction outside $R_0 = 6$ bohr about the carbonyl carbon, and $r_\eta = \eta(A_s)/\eta(N_s)$: > 3 diffuse, < 1.5 valence.

The measurement. No energy computed, it read the saved wave function. S_1: $r_\eta = 1.32$ valence; S_2: $r_\eta = \mathbf{27.7}$, diffuse — **confirmed**.

A colleague wants his 3-point extrapolation (6.626 eV) confirmed; ours is one point.

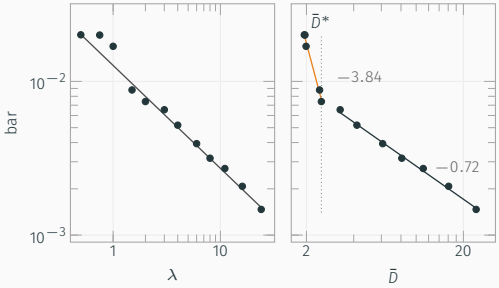
One real step: when the answer is no

The question. The dial scales the per-site spawn count; its error bar falls as the count rises. Does the fall break?

Frozen first. One continuous hinge; a pass needs $\Delta\chi^2 \geq 9.21$ and $\Delta_{\mathrm{slope}} - 2\,\mathrm{se}_{\mathrm{boot}} \geq 0.5$, with $\bar{D}^*$ located.

One real step: when the answer is no

Twelve dial values, 30 seeds each: 360 runs.



A mark is the 30-seed mean of a cell; the line is the fit.

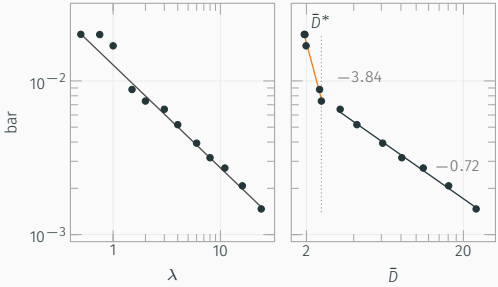
The question. The dial scales the per-site spawn count; its error bar falls as the count rises. Does the fall break?

Frozen first. One continuous hinge; a pass needs $\Delta\chi^2 \geq 9.21$ and $\Delta_{\text{slope}} - 2\,\text{se}_{\text{boot}} \geq 0.5$, with $\bar{D}^*$ located.

Two abscissae. On $\bar{D}$, $\Delta\chi^2 = 67.0$ at $\bar{D}^* = 2.489$, but $\Delta_{\text{slope}} - 2\,\text{se}_{\text{boot}} = -6.1$. On λ one power law holds, $\alpha = -0.670 \pm 0.019$, $\chi^2/(n-2) = 1.32$; the cells at $\lambda = 0.75$ and 1 sit 30% and 33% above it.

One real step: when the answer is no

Twelve dial values, 30 seeds each: 360 runs.



A mark is the 30-seed mean of a cell; the line is the fit.

Question: does the error bar's fall break as the count rises? Test frozen first: one hinge, $\Delta\chi^2 \geq 9.21$, $\Delta_{\text{slope}} - 2\,\text{se}_{\text{boot}} \geq 0.5$.

Two abscissae. On $\bar{D}$, $\Delta\chi^2 = 67.0$ at $\bar{D}^* = 2.489$, but $\Delta_{\text{slope}} - 2\,\text{se}_{\text{boot}} = -6.1$. On λ one power law holds, $\alpha = -0.670 \pm 0.019$, $\chi^2/(n-2) = 1.32$; the cells at $\lambda = 0.75$ and 1 sit 30% and 33% above it.

The null. A true λ power law, read on $\bar{D}$: mean $\Delta\chi^2 = 45.8$, $\bar{D}^*$ in a cell-free gap 60.2%, $\Delta\chi^2 \geq 9.21$ in 99.99% of 20000 simulations; 67.0: $p = 0.064$, above 1%.

Retracted. Only *located*: 2.489 is a conservative bound; *steepness* was and stays unresolved.

The surviving rule has no threshold: one dial for every order, the bar off the power law.

Summary

- **Tiny.** Two open-weight models — **Qwen3.8 27 B**, on *Turpan* or *Kairos*, and **DeepSeek V4.1** on *Kairos*, two GPU nodes we share at CALMIP — plus a **9 B** model on the laptop. Nothing here needs a frontier model.
- **Local.** The weights are files on machines we run, downloaded once and never rented. Logs, mail index, memory, research: **none of it leaves them**. No cloud LLM, no API key, no account that can be suspended.
- **The energy.** About as much power as a **PlayStation 5**; the two nodes are shared machines at CALMIP.
- **In control.** I choose the model, the context, the harness and the permissions.
- **And the goal is smaller**: a model on *this* laptop, smaller than the one I use today and **at least as good** — and when it appears, one download away.

One line

“AI didn’t replace the researcher: it moved the work — from execution to direction, control and verification. My tools are local, tiny and auditable. The result is not more papers: it is improved quality.”