

BLLM: Beyond the LLM with Deep Human-Mind-AI

Deep Bhattacharjee^{1,2,*} and Ushashi Bhattacharya^{3,†}

¹Formerly, Electro-Gravitational Space Propulsion Laboratory (EGSPL), Bhubaneswar, Odisha 751030, India

²Formerly, University of Calcutta, Senate House, 87/1 College Street, Kolkata 700073, West Bengal, India

³Formerly, National Taiwan University, Taipei 106319, Taiwan

(Dated: October 19, 2026)

A system that improves itself keeps every change that passes a check, so a check that makes errors lets mistakes pile up. We compute the least checking that keeps such a system sound. It is an entropy of the costs of the changes, and an endless chain of improvements stays sound at finite cost exactly when that entropy is finite. A stack of checked layers costs more than its layers would cost alone. We also compute how many guesses a proposer needs before a right one appears: human authors come close to this least number, and language models need 1.5 to 4.3 times as many. Anaxia, an engine built on these laws, rebuilt the published 2016 counterexample to Steinberg's conjecture from the rules of a colouring problem alone. The engine is a work in progress, and its rules and search are being revised and extended for the open problems it has not yet settled.

2020 Mathematics Subject Classification. 68T05; 68T20; 94A17; 60J80; 05C15.

Keywords: recursive self-improvement; verification; entropy; automated discovery

CONTENTS

I. Introduction	1	XVII. Experiments	36
II. History	6	XVIII. Construction mode	46
III. Search against an evaluator	9	XIX. One problem only: the Hodge round	48
IV. The model	10	XX. What is proved and what is open	52
V. The entropy law	11	A. Classical facts used in the proofs	56
VI. Huffman schedules	14	B. Formal proofs and numerical checks	58
VII. The Rényi law and the duality criterion	14	C. Hardware and reproduction	58
VIII. Self-improvement chains	17	D. What the engine draws on	59
IX. Sound singularities	18	E. Three nervous systems and one plan	62
X. Checking without knowing the future	20	F. Growth laws and the price of a sound takeoff	64
XI. Subagents, cascades and the moves of a proof	22	G. Formal proofs of hard problems	67
XII. The gap between human and LLM thinking, and how a BLLM closes it	25	Acknowledgements	70
XIII. Branching takeoff	28	Data and code availability	70
XIV. Towers of verified layers	29	Ethical declarations	70
XV. Anaxia: the engine	31	References	70
XVI. Five open problems as a benchmark	34		

I. INTRODUCTION

A system that rewrites its own weights, its own code or its own proofs, and keeps a rewrite when a test passes, is the basic unit of every proposal for recursive self-improvement. Good described the end point in 1965: a machine that designs a better machine, which designs a better one still, in an “intelligence explosion” that leaves human intelligence far behind

* itsdeep@live.com; * Corresponding author. Second address: d.bhattacharjee@erl-forschung.de.

† rai.bhattacharya.in@gmail.com

[81].¹ The literature since then has asked mostly how fast such a process could grow, under which assumptions on the returns to intelligence it reaches a vertical asymptote, and which physical limits stop it [6, 128]. We ask a different question, which has a sharper answer: what does it cost to make sure that what the process keeps is correct?

A second question sits beside the first. A language model proposes, and a measurement on 11,683 research papers shows how its proposals differ from those of the people who wrote the papers: the ideas of every model tested crowd into one or two kinds, chiefly the transfer of a method from one field to another, while human authors spread theirs over gaps in the evidence, mismatched scope and unexplained puzzles [55]. A sound system whose proposer seldom reaches the kind of idea a problem needs is sound and slow. We price this second gap with the same tools as the first, prove the least number of proposals any proposer can reach, measure how far the models are from it, and build a selector that learns from the checks alone to close it.

We call the object a *BLLM*, a system beyond the language model. It has three parts (Fig. 1). A generator, which may be a large language model [31, 179], proposes candidate modifications of the system. A bank of randomized checks tests each candidate; a check has perfect completeness, so a correct candidate always passes, and soundness $\varepsilon < 1$, so an incorrect candidate passes one run with probability at most ε . A memory keeps a candidate only after it has passed every run allotted to it, and the generator works from the memory. The name and the loop come from the verifier-gated engine of [23], in which a generator proposes steps toward a proof and a verifier accepts or rejects each one; we replace the single verifier by a schedule of independent runs, and that replacement is what makes the cost of soundness computable. The checks are the learned verifiers of [46, 107, 175], the randomized identity tests of [151, 191], the interactive and probabilistically checkable proofs of [3, 11, 13, 78], or a proof checker such as Lean [121] run on a randomized search. The cost of a run of the check of the i -th candidate is $a_i > 0$, and the schedule gives that check r_i independent runs.

A. What the title means

The title names two properties, and each is a measured quantity of the system we built, which we call *Anaxia* (Section XV).

Beyond the LLM means that a language model, where one is used at all, only proposes. Whether a proposal is kept is

decided by checks whose error is priced by the laws of this paper, and every object that the engine keeps from a search is checked again by code that shares nothing with the search. A language model that writes an answer and grades it with the same weights is the case these laws exclude: its check is not independent of its proposal, so the runs of the model have no soundness to spend.

Deep Human-Mind-AI means three things, each of which we measure and none of which says that the engine thinks as a person does.

- (a) *Proposals distributed as people distribute them.* On the axis of research opportunities of [55] the least expected number of proposals per problem is 6.50; the profile of the human authors pays 7.00, a factor 1.08, and the nine language models measured pay between 1.51 and 4.28 times the optimum (Table V). The selector of the engine learns the profile that attains the optimum from the results its checks keep.
- (b) *Human constructions as the raw material.* The default proposer is not a model but a typed crossing of constructions that mathematicians wrote down, each kept with its provenance, its type and the types it needs, so that a new candidate is a composite of known pieces that fit.
- (c) *Construction by hand, mechanised.* Construction mode writes down a new object together with the argument for what it forces, as a combinatorialist builds a gadget, and keeps it only after an independent check. From the rules of Steinberg’s problem alone it rebuilt the gadgets of the counterexample of Cohen-Addad, Hebdige, Král’, Li and Salgado [43].

B. The entropy law

Write $A = \sum_i a_i$ for the total cost of one run of every check, $p_i = a_i/A$ for the cost profile, $H(p) = \sum_i p_i \log(1/p_i)$ for its entropy, and $L = \log(1/\varepsilon)$. The schedule r is θ -sound when the expected number of incorrect candidates kept is at most θ in every assignment of correct and incorrect candidates, which for checks whose soundness ε is attained is the inequality $\sum_i \varepsilon^{r_i} \leq \theta$ (Section IV). Its cost is $V(r) = \sum_i a_i r_i$, the cost of certifying a run in which every candidate is correct, and $V^*(\theta)$ is the least cost of a θ -sound schedule.

Theorem A (Entropy law). *For $0 < \theta \leq 1$,*

$$\frac{A}{L} \left(H(p) + \log \frac{1}{\theta} \right) \leq V^*(\theta) \leq \frac{A}{L} \left(H(p) + \log \frac{1}{\theta} \right) + A. \quad (1)$$

The lower bound is attained when every θp_i is an integer power of ε , and the additive constant A cannot be replaced by cA with $c < 1$. The same bounds hold for infinitely many candidates with $A < \infty$, and for schedules drawn at random independently of the checks.

The law says that the price of soundness is an entropy. Certifying n candidates of equal cost costs $(n/L) \log(n/\theta)$ up to n , so the tax per candidate grows like $\log n$ and not like a constant: a system that keeps more modifications must test each

¹ I. J. Good, *Speculations concerning the first ultrainelligent machine*, written in 1965 and printed in *Advances in Computers*, volume 6 (1966), pages 31–88 [81]. Good’s argument is that an ultrainelligent machine could design better machines, so the first such machine is “the last invention that man need ever make, provided that the machine is docile enough to tell us how to keep it under control”. The proviso is the subject of the present paper in a precise form: the docility we ask for is that every improvement the machine keeps has been checked, and we compute what the checking costs.

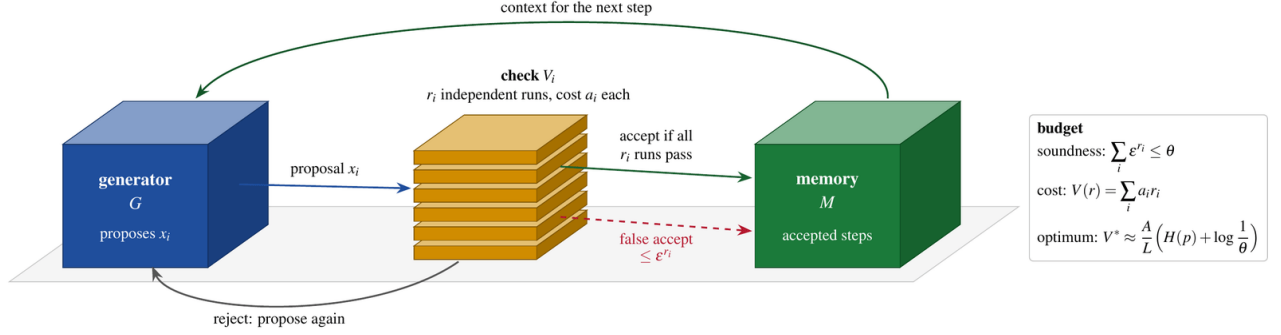


FIG. 1. A BLLM. The generator proposes candidate modifications; the i -th candidate is tested by r_i independent runs of a check V_i of cost a_i per run; an incorrect candidate survives all runs with probability at most ϵ^{r_i} and a correct one always survives; the memory keeps what survives and feeds it back as context. The budget box states the two quantities the paper controls, the expected number of false acceptances $\sum_i \epsilon^{r_i} \leq \theta$ and the cost $V(r) = \sum_i a_i r_i$, and the value of the optimum (Theorem A).

one harder. A skewed profile is cheaper than a flat one with the same total, because the expensive checks may be run fewer times and the cheap ones more, and the saving is measured exactly by $\log n - H(p)$. The schedule that attains the bound runs the i -th check $\lceil \log(A/(\theta a_i))/L \rceil$ times, so it spends $\log(1/p_i)$ nats of soundness on a candidate of relative cost p_i : this is Shannon's code length [156], and the analogy is exact.

Theorem B (Huffman schedules). *If $\epsilon = 1/D$ for an integer $D \geq 2$ and $\theta = D^{-k}$ for an integer $k \geq 0$, then the θ -sound schedules are exactly $r_i = \ell_i + k$ with (ℓ_i) the lengths of a D -ary prefix code, and $V^*(\theta) = A(k + L_D(p))$, where $L_D(p)$ is the expected length of a Huffman code for p .*

So the optimal schedule is computed by Huffman's algorithm [93], and Kraft's inequality [102, 116] is the soundness condition in disguise. The additive constant of Theorem A is the redundancy of the Huffman code, which lies in $[0, 1)$ in units of $\log D$ [53, 75].

Not every check is repeated. A statistical check that averages m samples and accepts when the average is close to its target has false acceptance probability of order $1/m$ by Chebyshev's inequality, not ϵ^m . For a check whose cost to reach false acceptance probability δ is $a\delta^{-s}$ the entropy is replaced by a Rényi entropy [144].

Theorem C (Rényi law). *For checks of polynomial order $s > 0$ and $0 < \theta \leq 1$, the least cost of a θ -sound allocation is*

$$V_s^*(\theta) = \theta^{-s} \left(\sum_i a_i^{1/(1+s)} \right)^{1+s} = A \theta^{-s} e^{sH_\alpha(p)}, \quad (2)$$

where $\alpha = 1/(1+s)$ and H_α is the Rényi entropy of order α ; and $V_s^*(\theta) = A(1 + s(H(p) + \log \frac{1}{\theta})) + O(s^2)$ as $s \rightarrow 0$.

The first-order term at $s = 0$ is the entropy law, so the Rényi law is a one-parameter deformation of it, as Campbell's coding theorem is a deformation of Shannon's [39]. Section VII also gives a criterion, valid for every cost function, for an infinite sequence of candidates to admit a sound schedule of finite cost.

C. Chains, gates and singularities

Section VIII puts the checks into a self-improvement chain in which the generator retries a step until a candidate is kept. When each step is wrong with probability between $\rho_* > 0$ and $\bar{\rho} < 1$, the least expected cost of the chain with at most θ expected false acceptances is pinched between the entropy law at budget θ/ρ_* and the entropy law at budget $\theta(1 - \bar{\rho})/\bar{\rho}$, divided by $1 - \bar{\rho}$ (Theorem F). The same section proves an exponential separation between checking every step and checking only the outcome: with perfect checks the ratio of expected costs is $(1 - \rho)^{-(n-1)} \geq e^{(n-1)\rho}$, and with imperfect checks the step check may accept an incorrect step with probability of order $1/n$ while the outcome check must do so with probability exponentially small in n (Theorem VIII.2).

An infinite chain whose total expected cost is finite completes infinitely many improvements in finite cost, and we call it *singular*, after the vertical asymptote of the explosion literature. It is *sound* when the number F of incorrect modifications it keeps is finite.

Theorem D (Sound singularities). *Let the n -th step cost g_n to generate and $a_n \leq 1$ per run to check, with error probabilities in $[\rho_*, \bar{\rho}] \subset (0, 1)$, independent steps and checks of exact soundness ϵ . Then*

- (i) $F < \infty$ almost surely, or $\mathbb{P}(F = 0) > 0$, if and only if $\sum_n \epsilon^{r_n} < \infty$, and otherwise $F = \infty$ almost surely;
- (ii) a singular chain in which every step is checked exists if and only if $\sum_n g_n < \infty$ and $\sum_n a_n < \infty$;
- (iii) a sound singular chain exists if and only if in addition $\sum_n a_n \log(1/a_n) < \infty$.

So there is a soundness gap. For $a_n = 1/(n \log^\beta n)$ a singular chain exists for every $\beta > 1$, and a sound one only for $\beta > 2$; between the two, every schedule fast enough to reach the asymptote keeps infinitely many errors almost surely (Corollary IX.2). For a power law $a_n = n^{-p}$ the price of soundness, the ratio of the cheapest sound cost to the cost of one

run per step, is $1/((p-1)L) + O(\log \frac{1}{p-1})$, and with polynomial checks of order s a sound singularity exists exactly when $p > 1 + s$ (Fig. 10).

When the improvements branch, each kept improvement spawning a random number of candidates, the process is a Galton–Watson tree [4, 183].

Theorem E (Branching takeoff). *Let m be the mean number of correct offspring kept per kept improvement, and let the check of generation k accept an incorrect candidate with probability δ_k . Then the takeoff survives with positive probability if and only if $m > 1$; the expected number of kept errors is $\mu(1-\gamma)\sum_{k \geq 1} m^{k-1} \delta_k$; and if this series diverges then, under $\mathbb{E} \xi \log \xi < \infty$, infinitely many errors are kept almost surely on survival. With exponential checks, soundness forces $r_k \geq k \log m / L$ for all large k , and $r_k = \lceil k \log(2m) / L \rceil$ suffices.*

Here μ is the mean number of candidates per kept improvement, γ the probability that a candidate is correct and ξ the number of candidates. A fixed false acceptance rate, however small, is fatal to an explosive takeoff, and the number of runs per check must grow linearly in the generation, that is, logarithmically in the size of the population: the entropy tax again.

D. Without foresight, among many agents

The schedules above are computed from the whole cost profile. A system that improves itself does not know its future candidates: when the n -th one arrives, only $a_1, \dots, a_n$ are known, and the runs of its check must be fixed then. Theorem G gives a rule that uses only the running total $S_n = a_1 + \dots + a_n$, is θ -sound for every sequence of costs, and pays over the offline optimum at most $\frac{A_N}{L} (\log 2 + 2 \log(1 + \log \frac{A_N}{a_1})) + A_N$ on the first N candidates; on the doubling costs every online rule pays an excess of order $\frac{A_N}{L} \log \log \frac{A_N}{a_1}$, so the double logarithm is the true price of not knowing the future (Fig. 11). Consequently the criterion of Theorem D needs no oracle: a chain that knows only its past reaches a sound singularity whenever any chain does (Corollary X.1).

A working system also delegates. When a coordinator splits the candidates among subagents, the entropy satisfies the chain rule over the split, and Theorem H shows that a coordinator which gives each subagent the share of the budget proportional to its total cost loses nothing: each subagent, knowing only its own costs and one number, computes on its candidates exactly the optimal schedule of the whole system, and along a tree of delegation the entropy tax is paid node by node. Any other split costs exactly $\frac{A}{L}$ times a Kullback–Leibler divergence. Finally, cheap checks of soundness ε placed before one exact check of cost c make the exact check paid only on the correct candidates, with a premium $\frac{b}{L} (1 + \log \frac{cPL}{b})$ logarithmic in the ratio of costs (Theorem XI.1). These two results are the laws of the system of subagents built in Section XI; Appendix G reads six machine-checked or published proofs of hard problems in their terms.

E. The gap between human and LLM thinking

Let a problem need an idea of class j with probability π_j , and let a proposer draw classes from a profile q until it draws the needed one. The expected number of proposals is $\sum_j \pi_j / q_j$, and Theorem I shows that it is at least $e^{H_{1/2}(\pi)} = (\sum_j \sqrt{\pi_j})^2$, a Rényi entropy of order one half, attained only by square-root sampling; the excess is exactly $e^{H_{1/2}(\pi)}$ times the chi-squared divergence of q from square-root sampling, and mixing a fraction λ of square-root sampling into any proposer brings its cost under $e^{H_{1/2}(\pi)} / \lambda$. On the data of [55] the optimum is 6.50 proposals per problem on the axis of research opportunities and 6.45 on the axis of methods; the human profile itself pays 7.00, and the nine models pay from 1.51 to 4.28 times the optimum (Table V). A BLLM learns the need from the results its checks keep, and its excess over the optimum falls like $\beta(\pi)/(4n)$ after n closed problems (Proposition XII.1), for a constant $\beta(\pi)$ that is 7.06 on the human profile of opportunities. The learning is done by a race of leaky integrate-and-fire neurons with a reward-modulated synapse, which samples exactly the square root of the counts it has learned (Proposition XII.2). Appendix E sets this circuit beside three wiring diagrams measured synapse by synapse, of the worm *C. elegans*, of the fly *Drosophila* and of a cubic millimetre of human cortex, and computes from them the sensory-to-motor depth, the modular structure and the laminar composition that the three parts of a BLLM reproduce.

F. Towers

An engine can act on another engine: one that chooses among strategies of a proposer and keeps the strategies that work stands on the proposer, and one that runs several copies of it and keeps the better half stands on that. Section XIV prices such a tower. If n verified layers share a budget θ of false acceptances, because a false acceptance low in the tower contaminates everything above it, then certifying the tower costs at least $n \log(n/\theta)$ nats of soundness, each added layer raises the bound by at least $\log((n+1)/\theta)$, and past a finite depth every tower costs more than any fixed amount (Theorem XIV.1). Depth pays only while it lowers the search by more than the increment, and the theorem says that this stops.

G. The engine and what it did

The theorems price the checks; they do not say where good candidates come from. Section XV describes the engine and Sections XVII to XIX measure it. Known constructions are typed nodes, each with the type of object it gives and the types it needs; a candidate is a cross that fills an input of one construction with the output of another; every closed cross is checked by a program, attacked by a refuter, and recorded in a shared memory. A target point fixes for each problem the statement that would settle it, five agents, one per problem, attack through a channel that brings every relevant record of the

memory to the proposer, and construction mode builds new objects where crossing has nothing to combine. We ran the engine on five open problems, the lattice mass gap of Yang–Mills theory, the Hodge conjecture, Grothendieck’s standard conjecture of Lefschetz type, Chowla’s conjecture and Erdős problem 1183, and on Steinberg’s problem as a calibration. The measurements are these.

- (i) *Typing does the work.* On Yang–Mills the network reached in every run a composite whose check accepts a lattice mass gap for four-dimensional $SU(2)$ on $\beta_W < \frac{1}{3}$, which no run of random or type-filtered pairing reached; random pairing wastes 45 of 48 crosses on ill-typed pairs. The composite was first proposed by the language-model proposer and then entered in the library, so the network reproduces the step rather than originating it. The step itself, Theorem XVII.4, is proved here and has not been refereed; it extends the range $\beta_W < \frac{2}{9}$ of [18], and Proposition XVII.5 shows that $\frac{1}{3}$ is the ceiling of the method.
- (ii) *Boosters do not help.* Hebbian weights with path credit reached that step in 12 and 14 of 200 runs against 21 for the unboosted network, a difference within two standard errors; a booster strengthens synapses that have fired, and the step needed one that had not.
- (iii) *Memory halves the cost.* Over 60 seeds and five rounds the full system reaches its plateau of about six survivors with 46.4 checks against 95.2 without memory, because it never checks a decided composite twice; it does not raise the plateau.
- (iv) *Agents are cheaper and reach no further.* After eleven rounds the five agents hold 24 distinct survivors for 250 checks and a single loop 27 for 797; the agents level off in the second round, the loop passes them in the sixth with known results, and no survivor of either reaches a target point.
- (v) *Patterns transfer partially or fail.* Of 38 transfers of a pattern that broke a comparable barrier elsewhere to a blocking step, none closes, 7 give a weaker statement or a new computation and 31 fail, for one of three reasons recorded in Section XVIII.H.
- (vi) *Construction mode rebuilds what crossing cannot.* Random planar frames filled with library gadgets gave no gadget of Steinberg’s kind in 2.45 million frames; the goal-directed builder, given only the rules, rebuilt the 15-vertex gadget of [43] on its 116th try after 11 seconds. With that gadget and a triangle as lemmas it built a 42-vertex gadget with the contacts of the second gadget of [43] on two seeds, and on a third a 48-vertex gadget with contact distances 5, 4, 5, which [43] does not use, and the assembly of [43] applied to the rebuilt 15-vertex gadget gives a planar graph of 166 vertices with no cycle of length 4 or 5 that three satisfiability solvers find not 3-colourable. On Erdős’s question, which also forbids cycles of length 6, it finds nothing (Section XVIII.A).
- (vii) *Depth pays once.* At equal budgets of exact checks, a second layer that first builds an easier gadget and keeps

it as a lemma reaches the profile of the second gadget of [43] on seven of ten runs, where the builder alone never reaches it; a third layer reaches it on six. When all layers share one library and one set of weights, the step from the lemma to the goal takes a median of 308 checks against 2,342 when each copy keeps its own. The crossing loop, given 13 to 20 times the budget, finds no gadget, and no tower finds one on Erdős’s question (Section XVIII.B).

- (viii) *One problem only.* Pointed at the Hodge conjecture alone, with every result of [17] and of the companion paper [30] in its library, the engine excludes by counting two of the configurations left open on E_0^6 and reduces two others to the ranks of explicit Massey products, and the gap of every target point is the same after the last round as after the first (Section XIX).

On Erdős problem 1183 a reduction to weighted posets (Lemma XVII.1) gives the upper bounds that, with the known lower bounds, determine $f(n)$ for $n \leq 20$ (Theorem XVII.2). None of the five problems is solved, and Section XVI.B states for each the step at which every construction we hold stops.

H. What the paper proves

Table I lists the results. The proofs are complete and elementary: the entropy law rests on the inequality $\log x \leq x - 1$, the Rényi law on Hölder’s inequality, the singularity theorem on the Borel–Cantelli lemmas, the online law on a telescoping inequality for $\log(S_n/a_1)$, the branching theorem on the Kesten–Stigum theorem [103] and Lévy’s conditional form of the Borel–Cantelli lemma [64], and the tower law on the entropy law with unit costs. Every finite inequality, the online and subagent laws, the gap law, the cascade bounds, the Kraft–schedule correspondence, the Rényi optimum, the convergence criteria for sound schedules, both Borel–Cantelli statements and the tower law are proved in Lean 4 with Mathlib, in sixty-five theorems that depend only on the three standard axioms (Appendix B); the probabilistic limit theorems and the asymptotics of the zeta function are proved in the text and are not formalised.

The paper does not prove that any existing system is sound, and it does not predict whether an intelligence explosion will happen. Its theorems are conditional on a model, stated in Section IV: independent runs, perfect completeness, and schedules fixed in advance. Remark V.10 shows that the last assumption matters, since a schedule allowed to stop at the first detected error escapes the entropy term at the price of discarding good work. What the paper does establish is that, inside the model, the cost of soundness is computable in closed form, that it is an entropy, and that it separates the singular processes into sound and unsound ones by a single convergent series.

I. Organisation

Section II traces the lines that meet here: self-reference and the intelligence explosion, coding and testing, branching

TABLE I. The main results. $A = \sum a_i$, $p = a/A$, $L = \log(1/\varepsilon)$; “Lean” names the formalised statements of Appendix B.

Result	Statement	Proof	Lean
Theorem A	$\frac{A}{L}(H + \log \frac{1}{\theta}) \leq V^*(\theta) \leq \frac{A}{L}(H + \log \frac{1}{\theta}) + A$	Gibbs, ceilings	verification_lower, ceilSchedule_cost
Proposition V.8	clean runs: budgets θ and $\log \frac{1}{1-\theta}$	$1 - \sum q \leq \prod (1-q) \leq e^{-\sum q}$	clean_ge, clean_budget
Proposition V.9	random oblivious schedules obey Theorem A	Jensen	randomised_budget
Proposition V.11	two-sided checks, rate $\frac{1}{2} \log \frac{1}{4q(1-q)}$	Chernoff	—
Theorem B	$V^*(D^{-k}) = A(k + L_D(p))$	Kraft, Huffman	kraft_of_schedule, schedule_of_kraft
Theorem C	$V_s^*(\theta) = A\theta^{-s} e^{sH_{1/(1+s)}(p)}$	Hölder	renyi_lower, renyi_attain
Theorem VII.3	sound schedule $\iff \sum a_n \psi(\lambda/a_n) < \infty$	Legendre duality	schedule_of_dual_summable
Theorem F	chain cost between two entropy laws	Wald	entropy_lower, ceilSchedule_sound
Theorem VIII.2	step versus outcome checks: $(1-p)^{-(n-1)}$	products	separation, precision_bounds
Theorem D	sound singularity $\iff \sum g, \sum a, \sum a \log \frac{1}{a} < \infty$	Borel–Cantelli	sound_schedule_iff, false_accepts_infinite
Corollary IX.2	soundness gap $1 < \beta \leq 2$	condensation	—
Proposition IX.3	price $\sim 1/((p-1)L)$ for $a_n = n^{-p}$	integral comparison	—
Proposition F.1	Osgood takeoff: check time $\asymp \kappa/(c(\alpha-1)^2L)$	entropy of ζ	—
Theorem G	online rule: excess $\leq \frac{A_N}{L}(\log 2 + 2\log(1 + \log \frac{A_N}{a_1})) + A_N$	telescoping	onlineSchedule_sound, online_lower
Theorem H	split price $\frac{A}{L}(D(P\ q) + \log \frac{\theta}{6})$	chain rule	chain_rule, split_lower, proportional_split
Theorem XI.1	cascade: $\min E = c(1-p) + \frac{b}{L}(1 + \log \frac{c^p L}{b}) + O(b)$	$\log x \leq x - 1$	cascade_lower, cascade_upper
Theorem E	branching takeoff and $r_k \geq k \log m/L$	Kesten–Stigum, Lévy	amplified_summable, contamination_diverges
Theorem XIV.1	n layers: $\sum_k \log \frac{1}{\beta_k} \geq n \log \frac{n}{\theta}$; saturation	Theorem A, unit costs	tower_cost_lower, tower_saturates
Theorem I	proposals $\sum \pi_j/q_j \geq e^{H_{1/2}(\pi)}$, excess $e^{H_{1/2}} \chi^2(r\ q)$	Cauchy–Schwarz	search_lower, search_chi2, search_mix
Proposition XII.1	learned square-root sampling: $1 + \beta(\pi)/(4n)$	second-order expansion	—
Proposition XII.2	first spike of the race is $\sqrt{c_k + \alpha}/\sum_l \sqrt{c_l + \alpha}$	exponential race	—
Lemma E.1	processing depth of a wiring diagram	Laplacian	—

processes, and verifiers for language models; Section III sets the engine beside the systems that search against an evaluator, FunSearch, AlphaEvolve, AlphaProof and the provers trained against Lean, and says what is taken from each and what is not. Section IV defines the model. Sections V to VII prove the entropy law, its Huffman form and its Rényi deformation. Section VIII treats chains and the comparison of step and outcome checks, Section IX singular chains, Section X the online law, Section XI subagents and cascades, Section XII the gap between human and model proposals and the spiking selector that closes it, Section XIII the branching takeoff and Section XIV towers of verified layers. Section XV describes Anaxia, Section XVI the five open problems with their pass lines, Section XVII the measurements of the crossing network, Section XVIII construction mode and the towers at equal cost, and Section XIX a round on the Hodge conjecture alone. Section XX states what is proved and what is open. Appendix A proves the classical facts used, Appendix B lists the Lean statements and numerical checks, Appendix C the hardware each part of the engine needs, Appendix D the systems and collections it draws on, and the last three appendices set the selector beside three measured nervous systems, relate the cost sequences to growth and scaling laws, and read six formal proofs of hard problems in the language of the paper.

Logarithms are natural unless a base is shown; $\mathbb{N} = \{0, 1, 2, \dots\}$.

II. HISTORY

Three lines of work meet in this paper, and a fourth, recent one supplies its motivation. Figure 2 places them on one time axis and Table II lists what each contributes to the argument.

A. Self-reference and the intelligence explosion

Turing’s paper of 1950 already contains the idea of a machine that improves itself: rather than program an adult mind, one should program a child machine and educate it, and he expected the educated machine to be modified by its own experience [172]. Fifteen years later Good drew the consequence for machines better than their designers [81]. The formal theory of such machines met an obstacle at once, and the obstacle is logical. Gödel’s second incompleteness theorem shows that a consistent system containing arithmetic does not prove its own consistency [80], and Löb’s theorem sharpens it: if such a system proves that provability of a sentence implies the sentence, then it proves the sentence [109].² An agent that rewrites itself

² Löb’s theorem answers a question of Henkin. Let T be a consistent recursively axiomatised extension of Peano arithmetic and Prov_T its provability predicate. If $T \vdash \text{Prov}_T(\ulcorner \varphi \urcorner) \rightarrow \varphi$ then $T \vdash \varphi$. Applied to a self-modifying agent that keeps a successor only when it proves the successor sound, the theorem forbids the agent from accepting, in general, a successor that uses

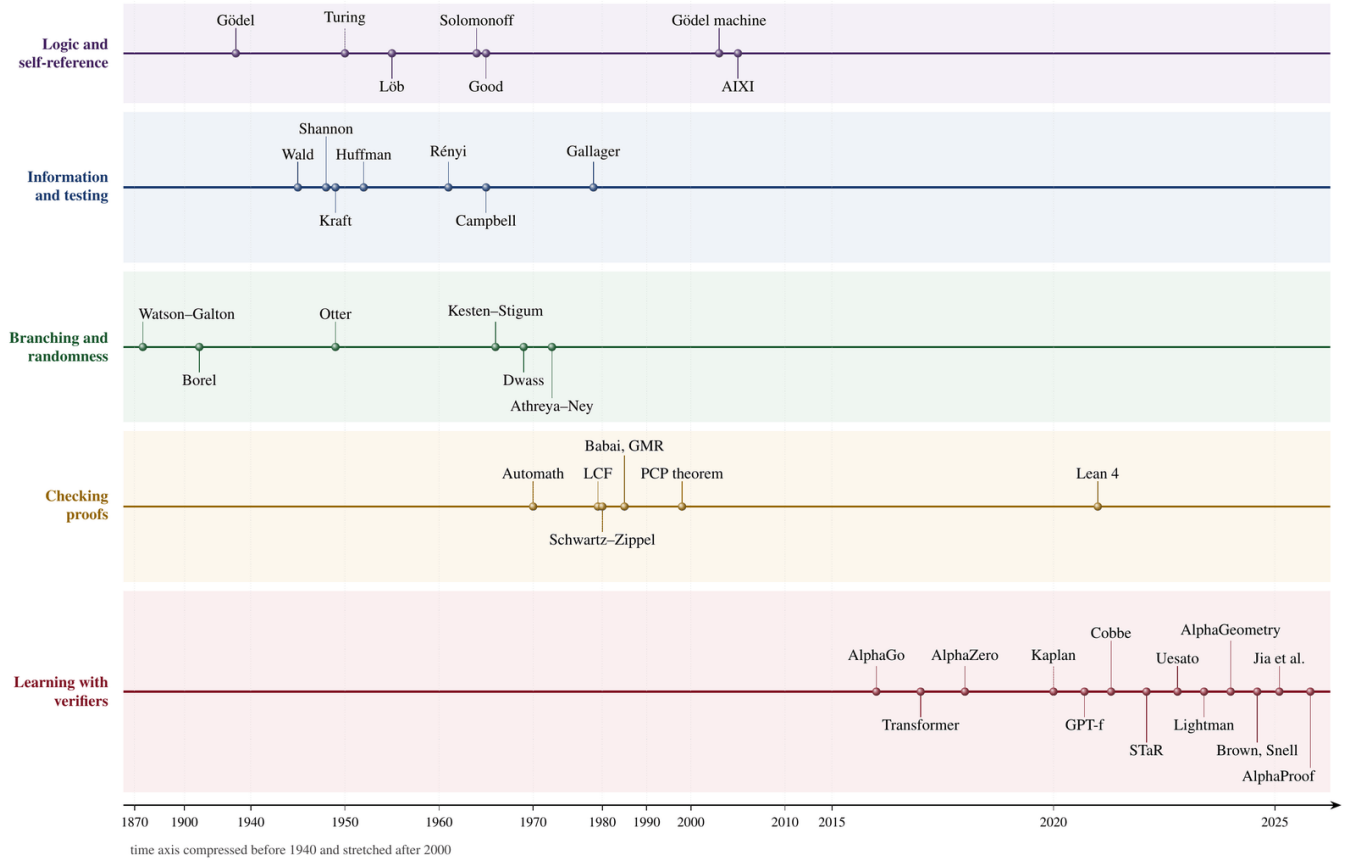


FIG. 2. Five lines of work on one time axis, with the scale stretched after 2015. Logic and self-reference: Gödel’s incompleteness theorems, Turing’s learning machine, Löb’s theorem, Solomonoff’s induction, Good’s intelligence explosion, the Gödel machine and AIXI. Information and testing: Wald’s sequential tests, Shannon’s source coding, Kraft’s inequality, Huffman’s codes, Rényi’s entropies, Campbell’s coding theorem and Gallager’s redundancy bound. Branching and randomness: the Watson–Galton problem, Borel’s denumerable probabilities, Otter’s multiplicative process, the Kesten–Stigum theorem, Dwass’s total progeny and the monograph of Athreya and Ney. Checking proofs: Automath, LCF, the polynomial identity test of Schwartz and Zippel, interactive proofs, the PCP theorem and Lean 4. Learning with verifiers: from AlphaGo and the transformer to trained verifiers, process supervision, repeated sampling and AlphaProof.

only after proving that the rewrite is sound cannot in general accept a successor as strong as itself. Schmidhuber’s Gödel machine makes this precise in the other direction: it rewrites any part of its own code once it has found a proof that the rewrite increases expected utility, and is optimal among such strategies relative to its axioms [150]. Solomonoff’s theory of inductive inference [162] and Hutter’s AIXI [94] give the ideal, uncomputable agents against which the practical ones are measured.

The present paper does not ask for proofs. It asks for checks with a stated probability of error and counts what they cost. That shift, from certainty to bounded error, is the one complexity theory made between the 1970s and the 1990s.

the same proof system with the same trust, since that acceptance would require the agent to prove a reflection principle for its own system. The present paper avoids the obstacle by replacing proofs of soundness with randomized checks of bounded error, whose reliability is a probability, not a theorem of the system.

B. Coding and testing

Shannon’s theorem of 1948 says that a source with distribution p cannot be encoded by a uniquely decodable code in fewer than $H(p)$ symbols on average, and that $H(p) + 1$ can be reached [156]. Kraft’s inequality, from his master’s thesis of 1949, characterises the lengths of prefix codes [102]; McMillan extended it to all uniquely decodable codes [116]; Huffman’s algorithm of 1952 finds an optimal code [93]; Gallager bounded its redundancy [75]. Rényi introduced in 1961 a family of entropies H_α that share the additivity of Shannon’s [144], and Campbell showed in 1965 that H_α is the optimal mean length when long codewords are penalised exponentially [39]. The standard account is [53].

The connection with checking is older than it looks. Wald’s sequential tests of 1945 already treat the trade-off between the number of observations and the two probabilities of error [180], and Chernoff’s exponent of 1952 [42] and Hoeffding’s inequality of 1963 [91] give the rates at which repeated obser-

vations drive the error down. What seems not to have been noticed is that the optimal allocation of repetitions among many checks of unequal cost under a common error budget is a coding problem, with costs in the role of probabilities and repetitions in the role of codeword lengths. Theorems A and B make the identification exact.

C. Branching and randomness

The Watson–Galton paper of 1875 asked for the probability that a surname dies out, and reduced the question to the iteration of a generating function [183]; the conclusion drawn there, that every family dies out, was wrong, and the correct criterion, extinction with probability one exactly when the mean number of sons is at most one, became standard only in the twentieth century. Otter’s multiplicative process of 1949 put the theory in its modern form [132], Kesten and Stigum proved in 1966 that the normalised population converges to a limit that is positive on survival exactly when $\mathbb{E} \xi \log \xi < \infty$ [103], and Dwass related the total progeny to a random walk [66]; the monograph of Athreya and Ney is the reference [4]. Borel’s paper of 1909 on denumerable probabilities contains the first form of the Borel–Cantelli lemma [32], which decides here whether a sound singular chain keeps finitely many errors. Osgood’s criterion of 1898 for blow-up of the equation $y' = f(y)$ [131] is the continuous form of the condition $\sum a_n < \infty$ for a singularity.

D. Checking proofs

De Bruijn’s Automath [56] and Milner’s LCF [79] began the mechanical checking of proofs, and Lean 4 [121] is the system used here. In parallel, complexity theory replaced certainty by bounded error. Schwartz and Zippel showed that a polynomial identity can be tested by evaluating at random points, with error at most the degree divided by the size of the field per run [151, 191]; Babai [13] and Goldwasser, Micali and Rackoff [78] introduced interactive proofs, in which a verifier is convinced with error probability that halves with each round; the PCP theorem showed that every proof can be written so that a verifier reads a constant number of its bits and is wrong with probability at most one half [3, 11]. These are exactly the checks of the present paper: perfect completeness, soundness $\varepsilon < 1$ per run, and independent repetition.

E. Learning with verifiers

The fourth line is recent. AlphaGo [158] and AlphaZero [159] learned games by self-play against an exact referee, the rules of the game. Transformers [179] and the scaling of language models [31, 86, 100] produced generators of proofs and programs, and verifiers were added to them: Polu and Sutskever coupled a language model to the Metamath checker [140], Cobbe and coauthors trained verifiers to rank solutions

of arithmetic problems [46], Uesato and coauthors and Lightman and coauthors compared rewards for the final answer with rewards for every step [107, 175], and chains of thought made the steps explicit [187]. Self-improvement loops followed: a model trained on its own checked rationales [192], on its own consistent answers [88], on its own feedback [120] or on a constitution [26]. Spending computation at inference time, by sampling many candidates and keeping the checked ones, became a lever of its own [25, 161]. AlphaGeometry [174] and AlphaProof [90] reached olympiad level by searching against exact checkers. Errors of transformers on compositional tasks grow with the number of steps [58], which is one reason to check every step. The same division of labour now builds scientific data. The wiring diagram of an adult fruit fly brain, 139,255 neurons and about 5×10^7 chemical synapses, was segmented automatically from electron micrographs and accepted neuron by neuron after manual proofreading, which took about 33 person-years [60].³

The theory has followed. Jia, Rakhlin and Xie showed that learning from outcome supervision is statistically no harder than learning from process supervision, up to polynomial factors in the horizon [97]; Botta and coauthors showed that a process verifier can turn an intractable generation problem into a tractable one [27]; Rohatgi and coauthors showed that small errors of a learned verifier are amplified along a generation and designed a backtracking sampler that resists them [148]; Balcan and coauthors studied learned verifiers of chains of thought and the asymmetric role of their soundness errors [19]; Scrivens reported that classifier-based safety gates fail over many rounds of self-improvement while a verifier with analytic bounds does not [153]; Guo turned impossibility results into design rules [82]; Lu compared improvements of agentic systems through bounded verification and showed that independent majority amplification preserves the languages it defines [112]. On the growth side, Ord showed that singular growth requires the time around the feedback loop to tend to zero [128], Anbar Jafari, Ozcinar and Anbarjafari gave testable conditions for runaway growth [6]. The verifier-gated engine of [23], in which a generator proposes steps toward a proof and a verifier accepts or rejects each one, is the informal model on which the present architecture is built.

None of these works prices soundness. They bound sample complexity, query complexity, error amplification or growth rates, and they treat the verifier as given. The question here is how a fixed budget of false acceptances should be distributed over many checks of different costs, and what that distribution costs; the answer is an entropy, and it decides which singular takeoffs are sound.

³ S. Dorkenwald et al., *Neuronal wiring diagram of an adult brain*, Nature 634 (2024), 124–138, doi:10.1038/s41586-024-07558-y [60]. A further round of proofreading of 826 central-brain neurons measured the accuracy of the accepted reconstruction at an average F_1 score of 99.2% by volume. The proofreading is the check of this paper and the person-years are its cost; Theorem A says how such a budget is best divided when neurons differ in size and the tolerated number of accepted errors is fixed.

TABLE II. Milestones used in the paper. The last column says where each enters the argument.

Year	Work	Contribution	Used in
1875	Watson, Galton [183]	extinction of families by iterating a generating function	Theorem E
1898	Osgood [131]	blow-up of $y' = f(y)$ in finite time	Section F
1909	Borel [32]	denumerable probabilities, Borel–Cantelli	Theorem D
1931, 1955	Gödel, Löb [80, 109]	limits of self-trust in formal systems	Section II
1945	Wald [180]	sequential tests and two probabilities of error	Section V F
1948	Shannon [156]	entropy as the optimal mean code length	Theorem A
1949, 1956	Kraft, McMillan [102, 116]	lengths of prefix and decodable codes	Theorem B
1949	Otter [132]	multiplicative processes	Theorem E
1952	Huffman [93]	optimal prefix codes	Theorem B
1952	Chernoff [42]	exponential rate of tests on sums	Proposition V.11
1961, 1965	Rényi, Campbell [39, 144]	entropies of order α and exponential code costs	Theorem C
1965	Good [81]	the intelligence explosion	Section I
1966	Kesten, Stigum [103]	the limit Z_k/m^k on survival	Theorem E
1979–1980	Schwartz, Zippel [151, 191]	randomized identity tests with one-sided error	Section IV
1985–1998	Babai; GMR; PCP [3, 11, 13, 78]	bounded-error verification of proofs	Section IV
2003	Schmidhuber [150]	self-rewriting after a proof of improvement	Section II
2021–2023	Cobbe; Uesato; Lightman [46, 107, 175]	trained verifiers, process and outcome rewards	Theorem VIII.2
2024–2025	AlphaGeometry, AlphaProof [90, 174]	search against exact checkers	Section IV
2025–2026	Jia et al.; Rohatgi et al.; Ord [97, 128, 148]	outcome supervision, verifier errors, singular growth	Sections VIII, F

III. SEARCH AGAINST AN EVALUATOR

The engine of this paper belongs to a family of systems that propose with a learned model and keep only what an evaluator accepts. Table III sets the nearest ones side by side, and this section says what each does, what Anaxia takes from it, and what is different here. The comparison is of designs, not of results: the systems below were run on other problems with other budgets, and none of them was rerun for this paper.

FunSearch. Romera-Paredes and coauthors let a language model write short programs, score each program by running it on the problem, and keep the best programs in separate islands that are fed back as examples [143]. The evaluator is exact for the quantity scored, so a kept program is correct by construction; it found cap sets in dimension eight larger than the known ones and new heuristics for bin packing. Anaxia takes the loop of evolving strategies against an exact evaluator; its second layer evolves policies of the graph builder rather than program text (Section XV H).

AlphaEvolve. Novikov and coauthors replace the single program by whole files edited by an ensemble of models, and run candidates through a cascade of tests of increasing cost, so that most of them are discarded by cheap tests before the expensive one [123]; Georgiev, Gómez-Serrano, Tao and Wagner applied it to many problems of analysis, combinatorics and geometry [76]. The cascade is the subject of Theorem XI.1, and the builder of Section XV G applies its cheap tests (forbidden cycles, distances, planarity) before the exact satisfiability check.

AlphaProof and AlphaGeometry. AlphaProof learns to prove statements in Lean by reinforcement learning against the kernel, and before a hard statement it trains on variants of it generated around the target [90]. AlphaGeometry runs a symbolic deduction engine to a fixed point and asks a language model for an auxiliary construction only when deduc-

tion stalls [174], and AlphaGeometry2 extends its language and its search [54]. Anaxia takes the curriculum (its second layer attacks weaker contact profiles first and keeps what they yield) and the division of labour (the builder derives forced colours to a fixed point and adds a vertex only when nothing more follows).

Provers trained against Lean. DeepSeek-Prover-V2 [149], Kimina-Prover [186] and Goedel-Prover [110] are language models trained by reinforcement learning to write Lean proofs; Seed-Prover proves lemmas, keeps them and reuses them [41]; Aristotle combines informal reasoning with verification in Lean [1]; HyperTree Proof Search searches over a tree of goals and subgoals [108], and LeanDojo retrieves premises from the library before each step [188]. Anaxia can use any of these models as its proposer through a local server, and none of their weights is bundled with it. Its obligation graph has the shape of a tree of goals, its memory retrieves past attempts as LeanDojo retrieves premises, and its library of found gadgets is used as Seed-Prover uses its lemmas.

What is different here. Program search against an evaluator, cascades of tests, curricula of variants, deduction to a fixed point, lemma reuse and retrieval were all introduced by the works above, and Anaxia uses them as such. Four things in this paper are of another kind. First, when the check is randomized or learned rather than exact, the works above do not say how many runs each candidate deserves; Sections V to XIII answer this exactly, and Section XIV extends the answer to stacks of searches. Second, the objects kept by construction mode are checked again from scratch by code that shares nothing with the search, so a fault in the search cannot certify its own output. Third, the default proposer is not a model but a typed crossing of constructions written down by people, with a memory that never checks the same composite twice; the language model is one proposer among four, chosen by a selector that is rewarded only on survivors. Fourth,

TABLE III. Systems that search against an evaluator. “Budget” says whether the system states and prices a bound on the expected number of wrong objects it keeps; “independent” whether the object kept is checked again by code that shares nothing with the search. The entries for other systems describe their published designs.

System	Proposer	Check	What is kept	Budget	Independent	Released
FunSearch [143]	language model writing programs	exact score of the program	best programs per island	not needed: exact	no	code, results
AlphaEvolve [123]	ensemble of models editing files	cascade of tests, then the score	programs and their outputs	not stated	no	results
AlphaProof [90]	policy network trained by reinforcement	Lean kernel	Lean proofs	not needed: kernel	kernel	proofs
AlphaGeometry [174]	language model for constructions	symbolic deduction	proofs	not needed: exact	no	code, weights
Lean provers [41, 110, 149, 186]	language models	Lean kernel	Lean proofs and lemmas	not needed: kernel	kernel	weights for some
Anaxia	typed crossing of human constructions, builders, any model	programs, refuter, Lean, randomized checks	survivors, gadgets, memory of every attempt	priced (Sections V–XIV)	yes	code, image, data

the experiments report on five open problems where nothing was solved, with the step at which every construction stops. We make no claim of priority for any of the methods named in this section.

IV. THE MODEL

A. Checks

A *candidate* is a proposed modification of the system: an edit of its weights, a patch of its code, a new lemma in its library, a step of a proof. It is either correct or incorrect, and nothing in the model depends on what the two words mean, only on the fact that the check can tell them apart with one-sided error.

Definition IV.1 (Check). A *check* for a candidate is a randomized procedure that outputs *accept* or *reject*. A *run* of the check uses fresh randomness, so that runs on the same candidate are independent. The check has *perfect completeness* if it accepts a correct candidate with probability one, and *soundness* $\epsilon \in (0, 1)$ if it accepts an incorrect candidate with probability at most ϵ in each run. It is *sharp* if the probability is exactly ϵ for every incorrect candidate. A run of the check of candidate i costs $a_i > 0$.

The Schwartz–Zippel test of a polynomial identity of degree d over a field with q elements is a check of soundness d/q [151, 191]; a round of an interactive proof with soundness one half is a check of soundness $1/2$ [13, 78]; a PCP verifier is a check of soundness $1/2$ that reads a constant number of bits [3, 11]; a search for a counterexample that succeeds with probability at least $1 - \epsilon$ on every incorrect candidate is a check of soundness ϵ . A proof checker such as Lean [121] is a check of soundness zero, and the model applies to it with ϵ replaced by the probability that the randomized search that produces the certificate misses an error the checker would detect. Checks need not be classical: the only property used is the independence of runs, which holds as well for repeated

preparation and measurement in a quantum procedure.⁴

B. Schedules, soundness and cost

There are n candidates, indexed by i , or infinitely many, indexed by $n \geq 1$. A *world* w is an assignment of the labels correct and incorrect to the candidates; it is arbitrary and unknown. A *schedule* is a vector $r = (r_i)$ of nonnegative integers fixed before any check runs. The protocol runs the check of candidate i at most r_i times, stops at the first rejection, and keeps the candidate if all r_i runs accept; a candidate with $r_i = 0$ is kept unchecked. The decisions on different candidates use independent randomness.

Definition IV.2 (Soundness and cost of a schedule). Let F be the number of incorrect candidates kept. The schedule is θ -*sound* if $\mathbb{E}_w F \leq \theta$ in every world w . Its *cost* is $V(r) = \sum_i a_i r_i$, and

$$V^*(\theta) = \inf \left\{ V(r) : r \in \mathbb{N}^n, \sum_i \epsilon^{r_i} \leq \theta \right\}.$$

Lemma IV.3. *Let every check have perfect completeness and soundness ϵ .*

- (i) *If $\sum_i \epsilon^{r_i} \leq \theta$, then r is θ -sound, and $\mathbb{P}_w(F \geq 1) \leq \theta$ in every world.*
- (ii) *If the checks are sharp and r is θ -sound, then $\sum_i \epsilon^{r_i} \leq \theta$.*
- (iii) *In every world and for every outcome of the checks, the cost of the protocol is at most $V(r)$, with equality in the world in which every candidate is correct.*

⁴ Independence is the one structural assumption. If the r runs of a check on an incorrect candidate pass together with probability $\kappa^{-1}(r)$ for some decreasing function, the analysis of Section VII applies with the cost function κ in place of the exponential one, and Theorem VII.3 gives the criterion for an infinite sequence of candidates. Correlated runs of a learned verifier, for example several samples of a reward model on the same input, are of this kind.

Proof. An incorrect candidate i is kept when its r_i independent runs all accept, which has probability at most ε^{r_i} , with equality for a sharp check. By linearity of expectation $\mathbb{E}_w F \leq \sum_{i \text{ incorrect in } w} \varepsilon^{r_i} \leq \sum_i \varepsilon^{r_i}$, and $\mathbb{P}_w(F \geq 1) \leq \mathbb{E}_w F$. In the world in which every candidate is incorrect and the checks are sharp, $\mathbb{E}_w F = \sum_i \varepsilon^{r_i}$, which proves (ii). The protocol spends at most r_i runs on candidate i , and exactly r_i when the candidate is correct, since a correct candidate is never rejected. $\square$

So for sharp checks $V^*(\theta)$ is the least cost of a θ -sound schedule, and for checks whose soundness is only bounded it is the least cost of a schedule that is θ -sound by the union bound. The cost $V(r)$ is the cost of the run the system exists to produce, one in which every candidate it keeps is correct, and it is also the largest cost the protocol can incur. The set of schedules of cost at most any given value is finite, because $a_i > 0$, so the infimum is a minimum when n is finite.

Definition IV.4 (Profile). For finitely many candidates, or for infinitely many with $A = \sum a_i < \infty$, the *cost profile* is $p_i = a_i/A$ and its *entropy* is $H(p) = \sum_i p_i \log(1/p_i) \in [0, \infty]$. Write $L = \log(1/\varepsilon) > 0$.

For n candidates $0 \leq H(p) \leq \log n$, with $H(p) = \log n$ exactly for equal costs; for infinitely many, $H(p)$ may be infinite even when A is finite, and Section IX turns on this possibility. The identity

$$AH(p) = \sum_i a_i \log \frac{A}{a_i} = \sum_i a_i \log \frac{1}{a_i} + A \log A \quad (3)$$

is used throughout.

C. Three assumptions

The model makes three assumptions, and each is tested in the paper. Independence of runs is relaxed in Section VII, where the cost of reaching false acceptance probability δ is an arbitrary function. Perfect completeness is relaxed in Section VF, where both kinds of error occur and the decision is a majority vote. The schedule fixed in advance is the assumption that cannot be removed without changing the answer: Remark V.10 shows that a protocol allowed to stop the whole run at the first rejection escapes the entropy term. Fixed schedules are natural for a BLLM, which allots a verification budget to each modification before it sees the outcome, and they are the only ones under which the decision on a candidate depends on that candidate alone.

V. THE ENTROPY LAW

A. Two inequalities

Everything in this section rests on $\log x \leq x - 1$ for $x > 0$, with equality only at $x = 1$.

Lemma V.1 (Gibbs term). For $x, y > 0$,

$$x \log \frac{1}{x} \leq x \log \frac{1}{y} + y - x,$$

with equality if and only if $x = y$.

Proof. Apply $\log t \leq t - 1$ to $t = y/x$ and multiply by x : $x \log y - x \log x \leq y - x$. $\square$

Lemma V.2 (Entropy term). For $a, b, A, \theta > 0$,

$$a \log \frac{A}{a} + a \log \frac{1}{\theta} \leq a \log \frac{1}{b} + \frac{Ab}{\theta} - a,$$

with equality if and only if $b = \theta a/A$.

Proof. Apply $\log t \leq t - 1$ to $t = Ab/(\theta a)$, so that $\log A + \log b - \log \theta - \log a \leq Ab/(\theta a) - 1$, and multiply by a . $\square$

Proposition V.3 (Entropy inequality). Let $a_1, \dots, a_n > 0$ with sum A , let $\beta_1, \dots, \beta_n > 0$ and $\theta > 0$ with $\sum_i \beta_i \leq \theta$. Then

$$AH(p) + A \log \frac{1}{\theta} \leq \sum_i a_i \log \frac{1}{\beta_i}, \quad (4)$$

with equality if and only if $\beta_i = \theta p_i$ for every i .

Proof. Sum Lemma V.2 with $(a, b) = (a_i, \beta_i)$. By (3) the left side is $AH(p) + A \log(1/\theta)$, and the right side is $\sum_i a_i \log(1/\beta_i) + (A/\theta) \sum_i \beta_i - A \leq \sum_i a_i \log(1/\beta_i)$. Equality forces equality in every term and $\sum_i \beta_i = \theta$, that is $\beta_i = \theta a_i/A$. $\square$

The inequality says that among all ways of spending a budget θ of false acceptances on n items, the weighted sum of the “surprisals” $\log(1/\beta_i)$ is least when the budget is shared in proportion to the costs. It is Gibbs’ inequality for the probability vectors p and β/θ , with a slack $\theta - \sum \beta_i$ that only helps.

B. Proof of the entropy law

Proof of Theorem A. We treat finitely many candidates; the infinite case is Proposition V.7.

Lower bound. Let r satisfy $\sum_i \varepsilon^{r_i} \leq \theta$ and put $\beta_i = \varepsilon^{r_i} > 0$. Then $\log(1/\beta_i) = r_i L$, and Proposition V.3 gives

$$AH(p) + A \log \frac{1}{\theta} \leq L \sum_i a_i r_i = LV(r).$$

Dividing by L and taking the minimum over r gives the left inequality of (1).

Upper bound. Let $r_i = \lceil x_i \rceil$ with $x_i = \log(A/(\theta a_i))/L$. Since $\theta \leq 1$ and $a_i \leq A$, $x_i \geq 0$, so $x_i \leq r_i < x_i + 1$. The first inequality gives $\varepsilon^{r_i} \leq \varepsilon^{x_i} = \theta a_i/A$, hence $\sum_i \varepsilon^{r_i} \leq \theta$. The second gives

$$V(r) < \sum_i a_i \left(\frac{1}{L} \log \frac{A}{\theta a_i} + 1 \right) = \frac{A}{L} \left(H(p) + \log \frac{1}{\theta} \right) + A.$$

Equality. If $\theta p_i = \varepsilon^{k_i}$ with $k_i \in \mathbb{N}$, the schedule $r_i = k_i$ has $\sum \varepsilon^{r_i} = \theta \sum p_i = \theta$ and $V(r) = \sum a_i \log(1/(\theta p_i))/L = (A/L)(H(p) + \log(1/\theta))$.

The constant A. Take $\varepsilon = \frac{1}{2}$, $\theta = 1$, $n = 2$ and $a = (1-t, t)$ with $0 < t < \frac{1}{2}$. A schedule with $r_1 = 0$ or $r_2 = 0$ has $\sum 2^{-r_i} > 1$, so every feasible schedule has $r_1, r_2 \geq 1$, and $r = (1, 1)$ is feasible; hence $V^*(1) = A = 1$. The lower bound is $H(p)/\log 2$, which tends to 0 as $t \rightarrow 0$. So the gap $V^* - (A/L)(H + \log \frac{1}{\theta})$ approaches A, and no constant cA with $c < 1$ bounds it. $\square$

Figure 3 shows the ceiling schedule for a power-law profile and compares its cost with the lower bound over twelve decades of θ .

Corollary V.4 (Equal costs). *For n candidates of cost a each, $\frac{na}{L} \log \frac{n}{\theta} \leq V^*(\theta) \leq \frac{na}{L} \log \frac{n}{\theta} + na$. In particular the number of runs per candidate in an optimal schedule is $\log(n/\theta)/L + O(1)$.*

Proof. Here $A = na$ and $H(p) = \log n$. $\square$

Proposition V.5 (Shape of the optimum). *Every optimal schedule is antitone in the costs: if $a_i > a_j$ then $r_i \leq r_j$. The real relaxation, the minimum of $\sum a_i x_i$ over $x \in [0, \infty)^n$ with $\sum \varepsilon^{x_i} \leq \theta$, is attained only at $x_i = \log(A/(\theta a_i))/L$ and equals the lower bound of (1).*

Proof. If $a_i > a_j$ and $r_i > r_j$, exchanging r_i and r_j keeps $\sum \varepsilon^r$ and lowers the cost by $(a_i - a_j)(r_i - r_j) > 0$, so r was not optimal. For the relaxation, Proposition V.3 with $\beta_i = \varepsilon^{x_i}$ gives $\sum a_i x_i \geq (A/L)(H + \log \frac{1}{\theta})$, with equality exactly when $\varepsilon^{x_i} = \theta p_i$. $\square$

The marginal price of soundness is constant: lowering θ by a factor e raises the lower bound by A/L , one nat of soundness bought on every check.

Corollary V.6 (Checks of different soundness). *If the check of candidate i has soundness ε_i , put $L_i = \log(1/\varepsilon_i)$, $b_i = a_i/L_i$, $B = \sum b_i$ and $q = B/B$. Then the least cost of a schedule with $\sum \varepsilon_i^{r_i} \leq \theta$, for $0 < \theta \leq 1$, lies between $B(H(q) + \log \frac{1}{\theta})$ and $B(H(q) + \log \frac{1}{\theta}) + A$.*

Proof. The cost is $\sum a_i r_i = \sum b_i(r_i L_i)$ and $\varepsilon_i^{r_i} = e^{-r_i L_i}$. Proposition V.3 with weights b_i and $\beta_i = e^{-r_i L_i}$ gives the lower bound, and $r_i = \lceil \log(B/(\theta b_i))/L_i \rceil$ gives $\sum \varepsilon_i^{r_i} \leq \theta \sum b_i/B = \theta$ and cost below $\sum b_i \log(B/(\theta b_i)) + \sum a_i$. $\square$

So the right unit of cost is the cost of one nat of soundness, a_i/L_i , and the entropy is that of the profile measured in this unit.

C. Infinitely many candidates

Proposition V.7. *Let $(a_n)_{n \geq 1}$ be positive with $A = \sum a_n < \infty$, and let $V^*(\theta)$ be the infimum of $\sum a_n r_n$ over $r \in \mathbb{N}^{\mathbb{N}}$ with $\sum \varepsilon^{r_n} \leq \theta$. Then (1) holds for $0 < \theta \leq 1$, with both sides infinite when $H(p) = \infty$.*

Proof. Let r be feasible and $N \geq 1$. The first N terms satisfy $\sum_{n \leq N} \varepsilon^{r_n} \leq \theta$, so the finite lower bound gives, with $A_N = \sum_{n \leq N} a_n$ and by (3),

$$\sum_{n \leq N} a_n \log \frac{1}{a_n} + A_N \log A_N + A_N \log \frac{1}{\theta} \leq L \sum_{n \leq N} a_n r_n \leq L V(r).$$

The terms $a_n \log(1/a_n)$ are positive once $a_n < 1$, which holds for all large n , so as $N \rightarrow \infty$ the left side tends to $\sum a_n \log(1/a_n) + A \log A + A \log(1/\theta) = A(H(p) + \log \frac{1}{\theta})$, finite or infinite. This is the lower bound. The ceiling schedule $r_n = \lceil \log(A/(\theta a_n))/L \rceil$ satisfies $\sum \varepsilon^{r_n} \leq \theta \sum p_n = \theta$ and $V(r) \leq (A/L)(H(p) + \log \frac{1}{\theta}) + A$ term by term. $\square$

D. Clean runs

A stronger requirement than $\mathbb{E}F \leq \theta$ is that the whole run be clean with probability at least $1 - \theta$: $\mathbb{P}_w(F = 0) \geq 1 - \theta$ in every world.

Proposition V.8 (Clean runs). *Let $q_i = \varepsilon^{r_i}$ and $0 < \theta < 1$. If $\sum q_i \leq \theta$ then every world has $\mathbb{P}_w(F = 0) \geq 1 - \theta$. If the checks are sharp and every world has $\mathbb{P}_w(F = 0) \geq 1 - \theta$, then $\sum q_i \leq \log \frac{1}{1-\theta} \leq \frac{\theta}{1-\theta}$. Hence the least cost of a clean schedule lies between $V^*(\log \frac{1}{1-\theta})$ and $V^*(\theta)$, and Theorem A holds for it with $\log \frac{1}{\theta}$ in the lower bound replaced by $\log \frac{1}{\theta} - \log \frac{1}{1-\theta}$.*

Proof. In world w the candidates are decided independently, so $\mathbb{P}_w(F = 0) = \prod_i \text{incorrect}(1 - q'_i)$ with $q'_i \leq q_i$, and this is at least $\prod_i (1 - q_i)$. By induction on n , $\prod_i (1 - q_i) \geq 1 - \sum_i q_i$ for $q_i \in [0, 1]$: the step is $(1-s)(1-q) = 1-s-q+sq \geq 1-s-q$. This proves the first claim. In the world in which every candidate is incorrect, sharpness gives $1 - \theta \leq \prod_i (1 - q_i) \leq \prod_i e^{-q_i} = e^{-\sum q_i}$, so $\sum q_i \leq \log \frac{1}{1-\theta}$, and $\log \frac{1}{1-\theta} = \sum_{k \geq 1} \theta^k/k \leq \theta/(1-\theta)$. $\square$

E. Random schedules

A schedule may be drawn at random, provided the draw is independent of the checks: the system may toss coins to decide how hard to test, but not look at the results to decide.

Proposition V.9 (Random oblivious schedules). *Let R be a random schedule taking finitely many values $r^{(1)}, \dots, r^{(m)}$ with probabilities $w_1, \dots, w_m$, independent of the checks, with $\mathbb{E} \sum_i \varepsilon^{R_i} \leq \theta$. Then $\mathbb{E} V(R) \geq (A/L)(H(p) + \log \frac{1}{\theta})$.*

Proof. Let $\bar{r} = \sum_j w_j r^{(j)} \in [0, \infty)^n$. By the weighted inequality of arithmetic and geometric means, which is Jensen's inequality for the exponential [95], $\varepsilon^{\bar{r}_i} = \prod_j (\varepsilon^{r_i^{(j)}})^{w_j} \leq \sum_j w_j \varepsilon^{r_i^{(j)}}$, so $\sum_i \varepsilon^{\bar{r}_i} \leq \mathbb{E} \sum_i \varepsilon^{R_i} \leq \theta$. Proposition V.3 with $\beta_i = \varepsilon^{\bar{r}_i}$ gives $L \sum_i a_i \bar{r}_i \geq A(H + \log \frac{1}{\theta})$, and $\sum a_i \bar{r}_i = \mathbb{E} V(R)$. $\square$

Remark V.10 (Adaptive protocols). The hypothesis that the schedule ignores the outcomes cannot be dropped. Consider the protocol that gives every check $r = \lceil \log(1/\theta)/L \rceil$ runs

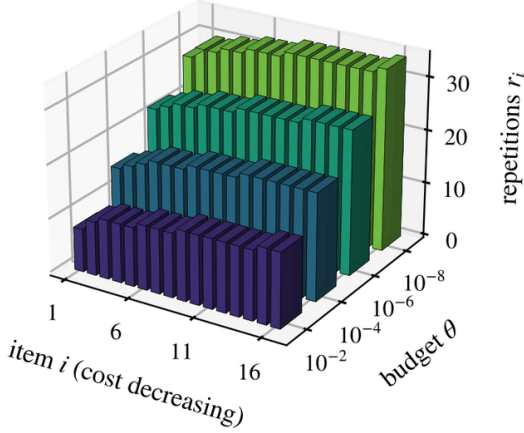
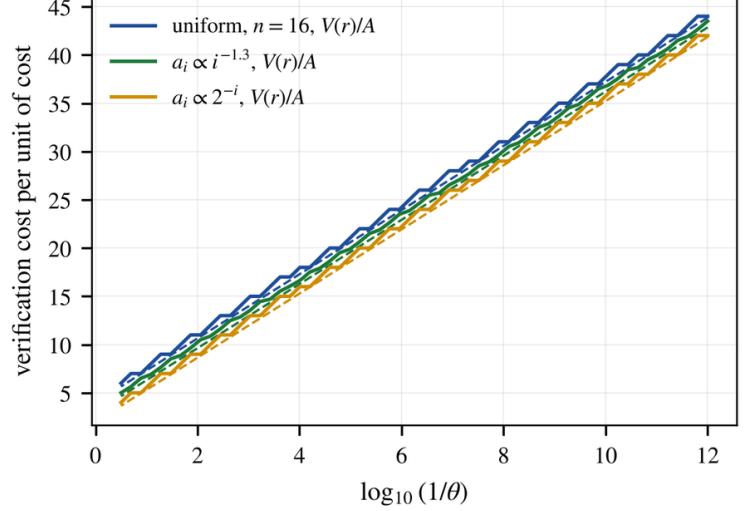
(a) ceiling schedule, $a_i \propto i^{-1.3}$, $\varepsilon = 1/2$ (b) ceiling schedule against $(H(p) + \log \frac{1}{\theta})/L$ (dashed)

FIG. 3. The entropy law. (a) The ceiling schedule $r_i = \lceil \log_2(A/(\theta a_i)) \rceil$ for sixteen candidates with costs $a_i \propto i^{-1.3}$ and $\varepsilon = \frac{1}{2}$, at the budgets $\theta = 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}$: the expensive candidates receive the fewest runs, and each factor 100 of θ adds six or seven runs to every check. (b) The cost of the ceiling schedule divided by A , for the uniform profile on sixteen candidates and the profiles $a_i \propto i^{-1.3}$ and $a_i \propto 2^{-i}$ on sixteen, against the lower bound $(H(p) + \log \frac{1}{\theta})/L$ (dashed). The curves stay in the band of width one above the bound, as Theorem A asserts, and the more skewed the profile the lower the line.

and halts the whole process, rejecting every later candidate, at the first rejection. In a world with incorrect candidates $i_1 < i_2 < \dots$, the j -th of them is kept only if it and the $j-1$ before it are all falsely accepted, which has probability at most θ^j , so $\mathbb{E}_w F \leq \sum_{j \geq 1} \theta^j = \theta/(1-\theta)$. Its cost in the world in which every candidate is correct is $Ar < (A/L) \log \frac{1}{\theta} + A$, without the entropy term. The saving is paid for elsewhere: a single incorrect candidate stops the process and every correct candidate after it is lost. Theorem A is therefore a theorem about protocols that decide each candidate on its own evidence, which is what a system that must keep its correct work has to do; whether an entropy term returns for adaptive protocols that keep every correct candidate and are charged their worst-case cost is a question we leave open.

F. Two-sided checks

Learned verifiers make errors in both directions. Suppose a run of the check of candidate i returns the correct verdict with probability $1-q$ and the wrong one with probability $q < \frac{1}{2}$, for correct and incorrect candidates alike, and the protocol keeps the candidate when a majority of r_i runs, r_i odd, accept. Then there are false acceptances and false rejections, and we ask that the expected number of each be at most θ . Write

$$D = \frac{1}{2} \log \frac{1}{4q(1-q)} = \text{KL}(\text{Ber}(\frac{1}{2}) \parallel \text{Ber}(q)),$$

the Chernoff exponent of the majority test [42, 53].

Proposition V.11 (Two-sided checks). (i) A majority of r runs is wrong with probability at most $(4q(1-q))^{r/2} =$

$$e^{-Dr}.$$

- (ii) For $0 < \theta \leq 1$ the schedule r_i equal to the least odd integer at least $\log(A/(\theta a_i))/D$ has at most θ expected false acceptances and at most θ expected false rejections, and costs at most $(A/D)(H(p) + \log \frac{1}{\theta}) + 2A$.
- (iii) Every decision rule that keeps a candidate whose runs all accept, used with any schedule with at most θ expected false acceptances in every world, costs at least $(A/\log(1/q))(H(p) + \log \frac{1}{\theta})$.

Proof. (i) Let S be the number of wrong verdicts among r runs, a binomial variable with parameters r and q ; the majority is wrong when $S \geq r/2$. For $t > 0$, Markov's inequality applied to e^{tS} gives $\mathbb{P}(S \geq r/2) \leq e^{-tr/2}(1-q+qe^t)^r$. With $e^t = (1-q)/q > 1$ the bracket is $2(1-q)$ and $e^{-tr/2} = (q/(1-q))^{r/2}$, so the bound is $2^r(1-q)^{r/2}q^{r/2} = (4q(1-q))^{r/2}$. (ii) By (i) each candidate is misjudged with probability at most $e^{-Dr_i} \leq \theta a_i/A$, whichever its label, so the expected numbers of both errors are at most θ ; and $r_i < \log(A/(\theta a_i))/D + 2$. (iii) An incorrect candidate whose r_i runs all return the wrong verdict is kept, which has probability q^{r_i} ; in the world in which every candidate is incorrect the expected number of false acceptances is at least $\sum q^{r_i}$, so $\sum q^{r_i} \leq \theta$, and Proposition V.3 with $\beta_i = q^{r_i}$ gives the bound. $\square$

The rate D is the exact exponent of the majority vote: by the method of types, $\mathbb{P}(S \geq r/2) \geq (r+1)^{-1} e^{-r \text{KL}(\lceil r/2 \rceil / r \parallel q)}$ [53], and $\text{KL}(\lceil r/2 \rceil / r \parallel q) \rightarrow D$. So for two-sided checks the entropy law holds with L replaced by D , and the price of a check that can make errors in both directions, compared with a one-sided check of the same per-run error q , is the ratio

$\log(1/q)/D$, which tends to 2 as $q \rightarrow 0$ and to infinity as $q \rightarrow \frac{1}{2}$.

VI. HUFFMAN SCHEDULES

When the soundness of a run is the reciprocal of an integer and the budget is a power of it, the optimal schedule is not only close to the entropy bound but computable exactly, and it is a code.

A. Schedules are codes

A D -ary prefix code for n symbols is a set of n words over an alphabet of D letters, none a prefix of another. Kraft's inequality says that lengths $\ell_1, \dots, \ell_n \in \mathbb{N}$ are the lengths of a prefix code if and only if $\sum_i D^{-\ell_i} \leq 1$ [102], and McMillan's theorem extends the necessity to all uniquely decodable codes [116]; Appendix A 1 recalls the proofs. For a probability vector p write $L_D(p)$ for the least value of $\sum_i p_i \ell_i$ over lengths satisfying Kraft's inequality, and $H_D(p) = H(p)/\log D$.

Lemma VI.1 (Kraft form of soundness). *Let $D > 1$ and $k \in \mathbb{N}$, and let $r \in \mathbb{N}^n$.*

- (i) *If $\sum_i D^{-r_i} \leq D^{-k}$, then $r_i \geq k$ for every i and $\ell_i = r_i - k$ satisfies $\sum_i D^{-\ell_i} \leq 1$.*
- (ii) *If $\sum_i D^{-\ell_i} \leq 1$, then $r_i = \ell_i + k$ satisfies $\sum_i D^{-r_i} \leq D^{-k}$.*
- (iii) $\sum_i a_i(\ell_i + k) = kA + \sum_i a_i \ell_i$.

Proof. (i) Each term satisfies $D^{-r_i} \leq \sum_j D^{-r_j} \leq D^{-k}$, and $x \mapsto D^{-x}$ is decreasing, so $r_i \geq k$; then $\sum D^{-(r_i-k)} = D^k \sum D^{-r_i} \leq 1$. (ii) $\sum D^{-(\ell_i+k)} = D^{-k} \sum D^{-\ell_i} \leq D^{-k}$. (iii) is linearity. $\square$

Proof of Theorem B. With $\varepsilon = 1/D$ and $\theta = D^{-k}$, the θ -sound schedules are the r with $\sum D^{-r_i} \leq D^{-k}$. By Lemma VI.1 they are exactly the vectors $\ell + k$ with ℓ satisfying Kraft's inequality, which by Kraft's theorem are the length vectors of prefix codes. The cost is $kA + \sum a_i \ell_i = A(k + \sum p_i \ell_i)$, and its minimum is $A(k + L_D(p))$. Huffman's algorithm produces a prefix code of expected length $L_D(p)$ (Appendix A 2). $\square$

So the optimal schedule is obtained by building the Huffman tree of the cost profile and adding k runs to every check (Fig. 4). Each run of a check is a letter of the codeword of its candidate; the first k letters buy the budget D^{-k} , the remaining ℓ_i letters distinguish the candidate from the others, and a candidate of high cost gets a short word, as a frequent symbol does.

Corollary VI.2. *Under the hypotheses of Theorem B, $A(k + H_D(p)) \leq V^*(D^{-k}) < A(k + H_D(p) + 1)$, with equality on the left if and only if every p_i is an integer power of $1/D$.*

Proof. Shannon's bounds $H_D(p) \leq L_D(p) < H_D(p) + 1$ [53, 156] are the case $\theta = 1$ of Theorem A, since $L = \log D$ and $(A/L)H(p) = AH_D(p)$; the equality case is that of Proposition V.3. $\square$

The strict inequality on the right is slightly better than Theorem A, and Gallager's bound improves it further when the largest cost share is small [75]: the redundancy $L_2(p) - H_2(p)$ is at most $p_{\max} + 0.086$ when the largest share $p_{\max}$ is below one half. So the additive constant A of the entropy law is, in this case, the redundancy of an optimal code.

B. Examples

Table IV compares three schedules on two profiles: the Huffman schedule, the ceiling schedule of Theorem A, and the lower bound. On a dyadic profile the three agree. On a profile that is not dyadic the ceiling schedule overspends, because the Shannon lengths $\lceil \log_2(1/p_i) \rceil$ are not optimal, and the Huffman schedule lies between it and the bound.

Figure 5 runs the comparison over twelve profiles: uniform, Zipf, geometric and random. The Huffman cost minus k is the Huffman length, which lies in $[H_2(p), H_2(p) + 1]$ and equals $H_2(p)$ only for the dyadic profiles.

Remark VI.3 (Larger alphabets). For $D \geq 3$ Huffman's algorithm merges the D smallest weights at each step, after adding zero weights until $n \equiv 1 \pmod{D-1}$; the dummy candidates receive the longest codewords and cost nothing (Appendix A 2). A check of soundness $1/3$ run r times is a ternary codeword of length r .

VII. THE RÉNYI LAW AND THE DUALITY CRITERION

A. Polynomial checks

A check is *exponential* when r runs drive its false acceptance probability to ε^r , so that the cost of reaching δ is $a \log(1/\delta)/L$. Many checks are slower. A statistical check that averages m independent samples of a statistic and accepts when the average is close to its target is the basic example.

Example VII.1 (Chebyshev checks). Let a candidate determine a random variable X with variance at most σ^2 , with mean 0 if the candidate is correct and mean at least $2t > 0$ if it is incorrect, and let the check accept when the mean of m independent samples is below t . By Chebyshev's inequality an incorrect candidate is accepted, and a correct one rejected,

TABLE IV. Schedules for $\varepsilon = \frac{1}{2}$, $\theta = 2^{-3}$. Costs are V/A .

	$p = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16})$	$p = (0.4, 0.3, 0.2, 0.1)$
Huffman lengths ℓ	(1, 2, 3, 4, 4)	(1, 2, 3, 3)
Huffman schedule r	(4, 5, 6, 7, 7)	(4, 5, 6, 6)
$\sum 2^{-r_i}$	2^{-3}	2^{-3}
cost, Huffman	4.875	4.900
ceiling schedule r	(4, 5, 6, 7, 7)	(5, 5, 6, 7)
cost, ceiling	4.875	5.400
bound $3 + H_2(p)$	4.875	4.846

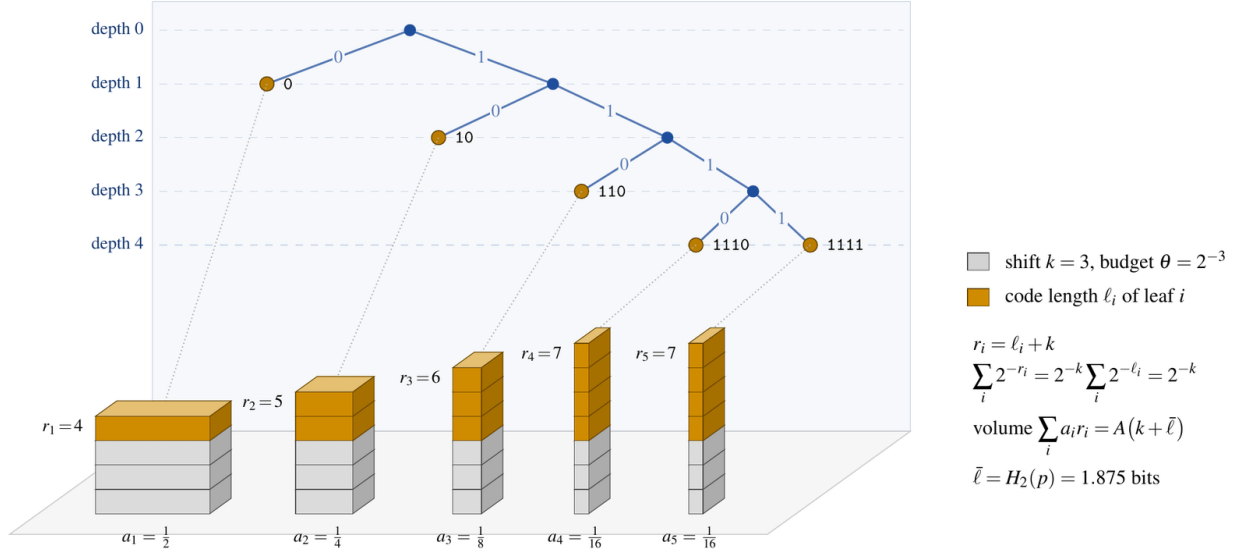
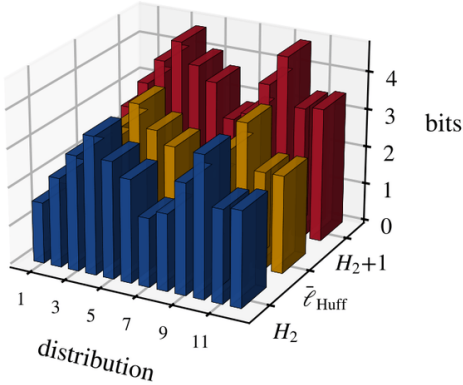


FIG. 4. Theorem B for the profile $p = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{16})$, $\varepsilon = \frac{1}{2}$ and $\theta = 2^{-3}$. Back wall: the binary Huffman tree, with codewords 0, 10, 110, 1110, 1111. Floor: a column for each candidate, of width proportional to its cost a_i , built from $k = 3$ grey blocks, which buy the budget 2^{-3} , and ℓ_i gold blocks, one per letter of its codeword. The heights are the optimal numbers of runs $r_i = 4, 5, 6, 7, 7$, the budget is exactly spent, $\sum 2^{-r_i} = 2^{-3}$, and the volume of the columns is the optimal cost $A(k + \bar{\ell})$ with $\bar{\ell} = H_2(p) = 1.875$.

(a) $V^*(2^{-k})/A - k$ equals the Huffman length



(b) redundancy lies in $[0, 1)$; zero exactly for dyadic p

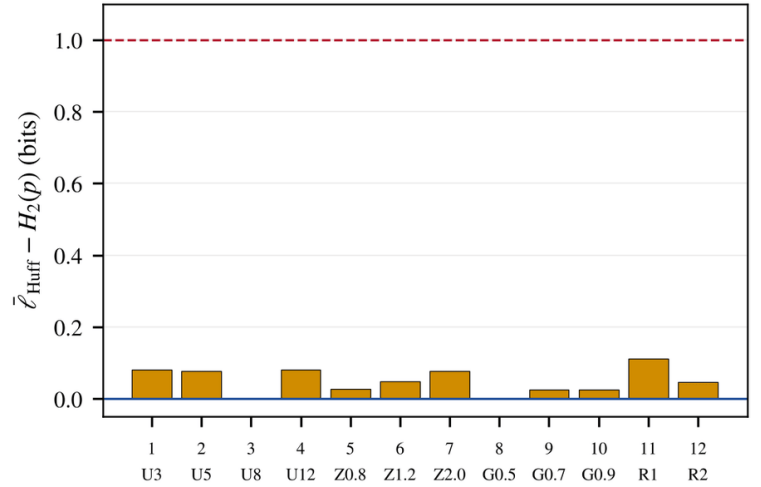


FIG. 5. Theorem B on twelve profiles: uniform on 3, 5, 8 and 12 candidates (U), Zipf with exponents 0.8, 1.2 and 2 on ten (Z), geometric with ratios 0.5, 0.7 and 0.9 on fourteen (G), and two random profiles on nine (R). (a) For each profile, the entropy $H_2(p)$, the optimal schedule cost $V^*(2^{-k})/A - k$, which is the Huffman length, and $H_2(p) + 1$. (b) The redundancy $V^*/A - k - H_2(p)$, which lies in $[0, 1)$; it vanishes on the dyadic profile, the uniform one on eight candidates (green), and is 2.7×10^{-5} for the geometric profile of ratio one half, whose normalised weights differ from powers of two by the factor $2/(2 - 2^{-13})$. The largest redundancy, 0.112, is that of the first random profile.

with probability at most $\sigma^2/(mt^2)$. So false acceptance probability δ costs $m = \sigma^2/(t^2\delta)$ samples: the cost is $a\delta^{-1}$ with a proportional to σ^2/t^2 .

We say a family of checks has *polynomial order* $s > 0$ if the check of candidate i reaches false acceptance probability $\delta_i \in (0, 1]$ at cost $a_i\delta_i^{-s}$, and that an allocation δ is θ -sound if $\sum_i \delta_i \leq \theta$. Chebyshev checks have order 1. Let $V_s^*(\theta)$ be the infimum of $\sum_i a_i\delta_i^{-s}$ over θ -sound allocations. For $0 < \alpha < 1$

the Rényi entropy of order α is

$$H_\alpha(p) = \frac{1}{1-\alpha} \log \sum_i p_i^\alpha,$$

which decreases from $\log n$ at $\alpha \rightarrow 0$ to $H(p)$ at $\alpha \rightarrow 1$ [53, 144].

Proof of Theorem C. Lower bound. Put $\alpha = 1/(1+s)$, so that

$1/\alpha = 1 + s$ and $1/(1 - \alpha) = (1 + s)/s$ are conjugate exponents. Write $a_i^\alpha = (a_i \delta_i^{-s})^\alpha \delta_i^{s\alpha}$ and apply Hölder's inequality:

$$\begin{aligned} \sum_i a_i^\alpha &\leq \left(\sum_i a_i \delta_i^{-s} \right)^\alpha \left(\sum_i \delta_i^{s\alpha/(1-\alpha)} \right)^{1-\alpha} \\ &= \left(\sum_i a_i \delta_i^{-s} \right)^{\frac{1}{1+s}} \left(\sum_i \delta_i \right)^{\frac{s}{1+s}}, \end{aligned}$$

since $s\alpha/(1 - \alpha) = 1$. With $\sum \delta_i \leq \theta$ and raising to the power $1 + s$, $(\sum_i a_i^\alpha)^{1+s} \leq \theta^s \sum_i a_i \delta_i^{-s}$.

Attainment. Let $S = \sum_j a_j^\alpha$ and $\delta_i = \theta a_i^\alpha / S$. Then $\sum \delta_i = \theta$, $\delta_i \leq \theta \leq 1$ because $a_i^\alpha \leq S$, and $\sum_i a_i \delta_i^{-s} = \theta^{-s} S^s \sum_i a_i^{1-s\alpha} = \theta^{-s} S^s \sum_i a_i^\alpha = \theta^{-s} S^{1+s}$, because $1 - s\alpha = \alpha$.

Entropy form. $S = A^\alpha \sum_i p_i^\alpha$, so $S^{1+s} = A (\sum_i p_i^\alpha)^{1+s}$, and $(1 + s) \log \sum_i p_i^\alpha = \frac{1+s}{s} s \log \sum_i p_i^\alpha = s H_\alpha(p)$, since $1 - \alpha = s/(1 + s)$.

The limit $s \rightarrow 0$. For finitely many candidates $\alpha \mapsto \log \sum_i p_i^\alpha$ is smooth and vanishes at $\alpha = 1$ with derivative $\sum p_i \log p_i = -H(p)$ there, so $H_\alpha(p) = H(p) + O(1 - \alpha) = H(p) + O(s)$. Hence $V_s^*(\theta)/A = \exp(s(H_\alpha(p) + \log \frac{1}{\theta})) = 1 + s(H(p) + \log \frac{1}{\theta}) + O(s^2)$. $\square$

The optimal allocation gives the i -th check the share a_i^α/S of the budget, a flattened version of the share a_i/A of the exponential case: when checks are slow, expensive checks must still be made quite reliable, because buying reliability anywhere is costly. The two laws are related by a limit. Since $\frac{1}{L} \log \frac{1}{\delta} = \lim_{s \rightarrow 0} \frac{\delta^{-s} - 1}{sL}$, the cost of an exponential check is the derivative at $s = 0$ of the cost of a polynomial one, and

$$\lim_{s \rightarrow 0} \frac{V_s^*(\theta) - A}{sL} = \frac{A}{L} \left(H(p) + \log \frac{1}{\theta} \right),$$

which is the real relaxation of the entropy law (Proposition V.5). Campbell's coding theorem has the same structure, with exponentially weighted codeword lengths and the Rényi entropy of order $1/(1 + t)$ [39].

Corollary VII.2 (Equal costs). *For n candidates of cost a and checks of order s , $V_s^*(\theta) = an^{1+s}\theta^{-s}$. Chebyshev checks cost an^2/θ , against $(an/L) \log(n/\theta) + O(an)$ for exponential checks.*

So the tax per candidate is $(n/\theta)^s$ for polynomial checks and $\log(n/\theta)/L$ for exponential ones: a statistical check that is not amplified exponentially makes the certification of a long run of modifications quadratically expensive. Figure 6 shows the law and its tangent at $s = 0$.

B. The duality criterion

For infinitely many candidates the question is not the value of the optimum but whether a sound allocation of finite cost exists. The answer, for every cost function, is a single series.

Let $\kappa: (0, 1] \rightarrow [0, \infty)$ be the cost, per unit of a , of making a check accept an incorrect candidate with probability at most

δ ; so $\kappa(\delta) = \log(1/\delta)/L$ for exponential checks with a real number of runs and $\kappa(\delta) = \delta^{-s}$ for checks of order s . Define the *dual cost*

$$\psi(u) = \inf_{0 < \delta \leq 1} (\kappa(\delta) + u\delta), \quad u \geq 0,$$

the least cost of a check when each unit of false acceptance probability is charged u . It is the concave conjugate of κ restricted to $(0, 1]$.

Theorem VII.3 (Duality criterion). *Let $a_n > 0$. The following are equivalent.*

- (a) *There are $\delta_n \in (0, 1]$ with $\sum_n \delta_n < \infty$ and $\sum_n a_n \kappa(\delta_n) < \infty$.*
- (b) *$\sum_n a_n \psi(\lambda/a_n) < \infty$ for some $\lambda > 0$.*
- (c) *$\sum_n a_n \psi(\lambda/a_n) < \infty$ for every $\lambda > 0$.*

When they hold, for every $\theta > 0$ there are $\delta_n \in (0, 1]$ with $\sum \delta_n \leq \theta$ and $\sum a_n \kappa(\delta_n) < \infty$.

Proof. (a) $\Rightarrow$ (c). For every $\lambda > 0$ and n , $0 \leq a_n \psi(\lambda/a_n) \leq a_n \kappa(\delta_n) + \lambda \delta_n$, and the right side is summable. (c) $\Rightarrow$ (b) is trivial. (b) $\Rightarrow$ (a). By the definition of the infimum choose $\delta_n \in (0, 1]$ with $\kappa(\delta_n) + (\lambda/a_n)\delta_n \leq \psi(\lambda/a_n) + 2^{-n}/a_n$. Then $a_n \kappa(\delta_n) + \lambda \delta_n \leq a_n \psi(\lambda/a_n) + 2^{-n}$, a summable bound for two nonnegative terms, so both $\sum a_n \kappa(\delta_n)$ and $\sum \delta_n$ converge. For the last claim choose N with $\sum_{n>N} \delta_n \leq \theta/2$ and replace δ_n by $\theta/(2N)$ for $n \leq N$; this changes finitely many finite terms of the cost. $\square$

Proposition VII.4 (The two dual costs). *The two cost functions of the paper have the following duals.*

- (i) *For exponential checks the cost is $\kappa(\delta) = \log(1/\delta)/L$ and the dual is*

$$\psi(u) = \begin{cases} (1 + \log(uL))/L, & uL \geq 1, \\ u, & uL < 1. \end{cases}$$

- (ii) *For checks of order s , $\kappa(\delta) = \delta^{-s}$, the dual is*

$$\psi(u) = \begin{cases} (1 + s) s^{-s/(1+s)} u^{s/(1+s)}, & u \geq s, \\ 1 + u, & u < s. \end{cases}$$

Proof. Both functions $\delta \mapsto \kappa(\delta) + u\delta$ are convex on $(0, 1]$. In (i) the derivative $-1/(L\delta) + u$ vanishes at $\delta = 1/(uL)$, which lies in $(0, 1]$ exactly when $uL \geq 1$, with value $\log(uL)/L + 1/L$; otherwise the derivative is negative on $(0, 1]$ and the minimum is at $\delta = 1$. In (ii) the derivative $-s\delta^{-s-1} + u$ vanishes at $\delta = (s/u)^{1/(1+s)}$, which lies in $(0, 1]$ exactly when $u \geq s$, with value $(s/u)^{-s/(1+s)} + u(s/u)^{1/(1+s)} = (1 + s) s^{-s/(1+s)} u^{s/(1+s)}$; otherwise the minimum is at $\delta = 1$. $\square$

Corollary VII.5. *Let $a_n \rightarrow 0$.*

- (i) *Exponential checks with real numbers of runs admit a sound allocation of finite cost if and only if $\sum a_n < \infty$ and $\sum a_n \log(1/a_n) < \infty$.*

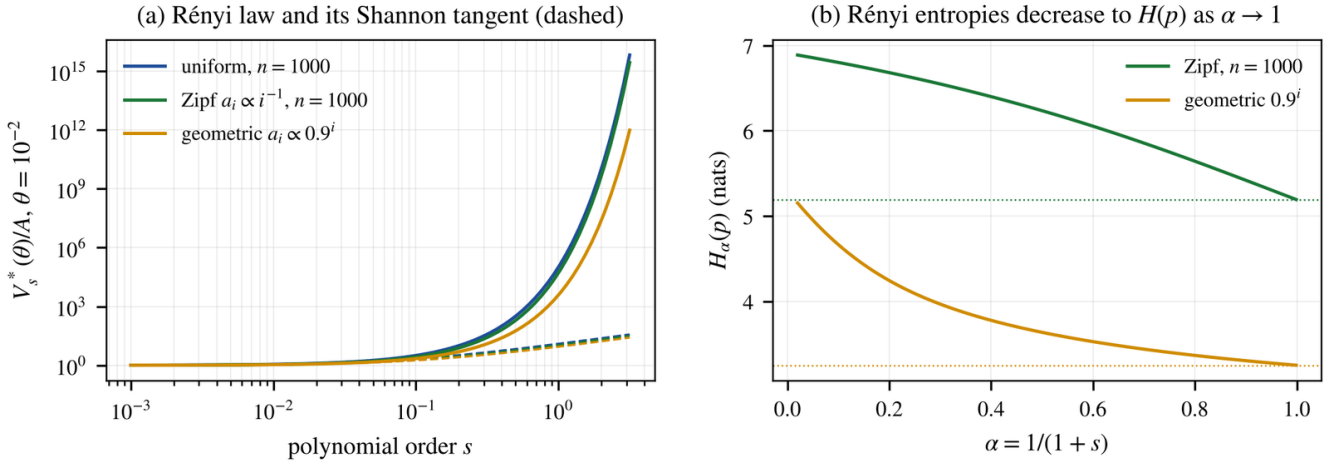


FIG. 6. The Rényi law, Theorem C. (a) $V_s^*(\theta)/A = \theta^{-s} e^{sH_{1/(1+s)}(p)}$ at $\theta = 10^{-2}$ for a uniform profile and a Zipf profile on a thousand candidates and a geometric profile of ratio 0.9, with the Shannon tangents $1 + s(H(p) + \log \frac{1}{\theta})$ (dashed). The tangents are exact as $s \rightarrow 0$ and the flat profile is the most expensive at every order. (b) The Rényi entropies $H_\alpha(p)$ of the Zipf and geometric profiles as functions of $\alpha = 1/(1+s)$, decreasing to the Shannon entropy (dotted) as $\alpha \rightarrow 1$.

(ii) Checks of order s admit one if and only if $\sum a_n^{1/(1+s)} < \infty$.

Proof. For large n , $u = \lambda/a_n$ is large, and by Proposition VII.4 $a_n \psi(\lambda/a_n)$ equals $\frac{a_n}{L} (1 + \log(\lambda L) + \log \frac{1}{a_n})$ in case (i) and a constant multiple of $a_n^{1/(1+s)}$ in case (ii). In case (i) the sum converges if and only if $\sum a_n (1 + \log \frac{1}{a_n})$ does, and both terms are eventually positive. Theorem VII.3 concludes. $\square$

The integer version needs no duality.

Proposition VII.6 (Integer schedules). *Let $0 < a_n \leq 1$ with $\sum a_n < \infty$. There is $r \in \mathbb{N}^{\mathbb{N}}$ with $\sum \epsilon^{r_n} < \infty$ and $\sum a_n r_n < \infty$ if and only if $\sum a_n \log(1/a_n) < \infty$; the schedule $r_n = \lceil \log(1/a_n)/L \rceil$ then works, and adding k runs to every check multiplies $\sum \epsilon^{r_n}$ by ϵ^k .*

Proof. If r is such a schedule, Lemma V.1 with $x = a_n$ and $y = \epsilon^{r_n}$ gives $a_n \log(1/a_n) \leq a_n r_n L + \epsilon^{r_n} - a_n \leq L a_n r_n + \epsilon^{r_n}$, a summable bound for a nonnegative sequence. Conversely $r_n = \lceil \log(1/a_n)/L \rceil$ gives $\epsilon^{r_n} \leq a_n$ and $a_n r_n \leq a_n \log(1/a_n)/L + a_n$. The last claim is $\epsilon^{r_n+k} = \epsilon^k \epsilon^{r_n}$. $\square$

Theorem VII.3 is the infinite form of the entropy and Rényi laws: it says which cost profiles can be certified at all, and Section IX applies it to self-improving chains.

VIII. SELF-IMPROVEMENT CHAINS

A. The chain

A BLLM improves itself in steps. At step i the generator proposes candidates for the i -th modification until one is kept, then moves to step $i+1$. A candidate for step i costs $g_i \geq 0$ to generate, is incorrect with probability ρ_i , independently of everything else, and is checked by r_i runs of a sharp check of soundness ϵ and cost a_i per run, stopping at the first rejection.

Let T_i be the total cost of step i , F the number of steps whose kept candidate is incorrect, $G = \sum g_i$, and write $\beta_i = \epsilon^{r_i}$.

Lemma VIII.1 (One step). *Let $s_i = 1 - \rho_i + \rho_i \beta_i$ be the probability that a candidate is kept and $\pi_i = \rho_i \beta_i / s_i$ the probability that the kept candidate is incorrect. Then*

$$g_i + a_i r_i \leq \mathbb{E} T_i \leq \frac{g_i + a_i r_i}{1 - \rho_i}, \quad \rho_i \beta_i \leq \pi_i \leq \frac{\rho_i \beta_i}{1 - \rho_i}.$$

Proof. The number of candidates proposed at step i is geometric with success probability s_i , so its mean is $1/s_i$, and by Wald's identity $\mathbb{E} T_i$ is $1/s_i$ times the expected cost c_i of one candidate. A candidate costs g_i plus at most r_i runs, so $c_i \leq g_i + a_i r_i$, and $s_i \geq 1 - \rho_i$; this is the upper bound. The kept candidate passed all its r_i runs, so it alone costs $g_i + a_i r_i$; this is the lower bound. The bounds on π_i follow from $1 - \rho_i \leq s_i \leq 1$. $\square$

Theorem F (Chain law). *Let $0 < \rho_* \leq \rho_i \leq \bar{\rho} < 1$ for every i , and let $C^*(\theta)$ be the least value of $\sum_i \mathbb{E} T_i$ over schedules with $\mathbb{E} F \leq \theta$. Then*

$$C^*(\theta) \geq G + \frac{A}{L} \left(H(p) + \log \frac{\rho_*}{\theta} \right),$$

$$C^*(\theta) \leq \frac{1}{1 - \bar{\rho}} \left[G + \frac{A}{L} \left(H(p) + \log \frac{\bar{\rho}}{(1 - \bar{\rho})\theta} \right) + A \right],$$

the upper bound for $0 < \theta \leq \bar{\rho}/(1 - \bar{\rho})$.

Proof. $\mathbb{E} F = \sum_i \pi_i$. If $\mathbb{E} F \leq \theta$, then $\rho_* \sum \beta_i \leq \sum \pi_i \leq \theta$ by Lemma VIII.1, so r is feasible for the budget θ/ρ_* , and Proposition V.3 gives $V(r) \geq (A/L)(H + \log(\rho_*/\theta))$; also $\sum \mathbb{E} T_i \geq G + V(r)$. Conversely let $\theta' = \theta(1 - \bar{\rho})/\bar{\rho} \leq 1$ and let r be the ceiling schedule for θ' . Then $\mathbb{E} F \leq \frac{\bar{\rho}}{1 - \bar{\rho}} \sum \beta_i \leq \theta$, and $\sum \mathbb{E} T_i \leq (G + V(r))/(1 - \bar{\rho})$ with $V(r) \leq (A/L)(H + \log \frac{1}{\theta'}) + A$ by Theorem A. $\square$

So the cost of a sound chain is the entropy law with the budget rescaled by the error rate of the generator, plus the cost of generation, inflated at most by the factor $1/(1-\bar{\rho})$ for the retries. A better generator, one with smaller $\bar{\rho}$, moves the two bounds together.

B. Checking steps or checking outcomes

A chain of n steps can be checked step by step, or run to the end and checked once. Process supervision and outcome supervision are the two names of this choice for language models [107, 175], and the following theorem compares their costs when the checks are given.

Consider n steps, each correct with probability $1-\rho_i$, each costing one unit to generate. *Step gating* checks step i with a check that accepts a correct step with probability $1-\eta$ and an incorrect one with probability δ , and regenerates the step until it is accepted. *Outcome gating* generates all n steps, checks the outcome with a check that accepts a correct chain with probability $1-\eta_o$ and an incorrect one with probability δ_o , and regenerates the whole chain until it is accepted. A chain is correct when all its steps are. The *precision* of a protocol is the probability that the chain it returns is correct.

Theorem VIII.2 (Step and outcome gating). *Let $c = \prod_i (1-\rho_i)$ and $x_i = \rho_i \delta / ((1-\rho_i)(1-\eta))$.*

- (i) *Step gating has expected cost $\sum_i 1/s_i$ with $s_i = (1-\rho_i)(1-\eta) + \rho_i \delta$, and precision $\prod_i (1+x_i)^{-1}$, which lies between $e^{-\sum x_i}$ and $e^{-\sum x_i/(1+x_i)}$.*
- (ii) *Outcome gating has expected cost n/s_o with $s_o = c(1-\eta_o) + (1-c)\delta_o$, and precision $(1 + (1-c)\delta_o/(c(1-\eta_o)))^{-1}$.*
- (iii) *With perfect checks and $\rho_i = \rho$, the ratio of the expected cost of outcome gating to that of step gating is $(1-\rho)^{-(n-1)} \geq \max(1 + (n-1)\rho, e^{(n-1)\rho})$.*
- (iv) *Let $\rho_i = \rho$, $0 < \zeta < 1$ and $\ell = \log \frac{1}{1-\zeta}$. Step gating reaches precision $1-\zeta$ if $\delta \leq (1-\rho)(1-\eta)\ell/(\rho n)$, and only if $\delta \leq (1-\rho)(1-\eta)\ell/(\rho(n-\ell))$ when $n > \ell$. Outcome gating reaches precision $1-\zeta$ if and only if*

$$\delta_o \leq \frac{\zeta}{1-\zeta} \cdot \frac{c(1-\eta_o)}{1-c}, \quad c = (1-\rho)^n.$$

Proof. (i) Step i is accepted at each attempt with probability s_i , so it takes $1/s_i$ attempts on average, and the accepted step is correct with probability $(1-\rho_i)(1-\eta)/s_i = 1/(1+x_i)$. The steps are independent, so the precision is the product. For $x \geq 0$, $x/(1+x) \leq \log(1+x) \leq x$, which gives the two exponential bounds. (ii) The same argument with one step of cost n , acceptance probability s_o and probability $c(1-\eta_o)/s_o$ that the accepted chain is correct. (iii) With $\eta = \delta = 0$ step gating costs $n/(1-\rho)$ and outcome gating $n/(1-\rho)^n$. Bernoulli's inequality gives $(1+\rho)^{n-1} \geq 1 + (n-1)\rho$, and $(1-\rho)^{-1} \geq 1+\rho$, so $(1-\rho)^{-(n-1)} \geq 1 + (n-1)\rho$; and

$1-\rho \leq e^{-\rho}$ gives the exponential bound. (iv) By (i) the precision is $(1+x)^{-n}$ with $x = \rho \delta / ((1-\rho)(1-\eta))$, and it is at least $1-\zeta$ exactly when $n \log(1+x) \leq \ell$. If $x \leq \ell/n$ this holds because $\log(1+x) \leq x$. If it holds, then $n x / (1+x) \leq \ell$, that is $x \leq \ell/(n-\ell)$. For outcome gating, (ii) gives precision at least $1-\zeta$ exactly when $(1-c)\delta_o/(c(1-\eta_o)) \leq \zeta/(1-\zeta)$. $\square$

The theorem separates the two protocols by an exponential. A step check needs false acceptance rate of order $1/n$, which an exponential check reaches with $\log n/L$ runs, so step gating with exponential checks costs order $n \log n$, as Corollary V.4 predicts. An outcome check needs false acceptance rate of order $\zeta(1-\rho)^n$, which costs $n \log \frac{1}{1-\rho}/L$ runs on every attempt, and even a perfect outcome check leaves the factor $(1-\rho)^{-(n-1)}$ of retries. Figure 7 draws both effects, and Fig. 8 the two protocols.

Remark VIII.3 (Relation to learning from outcomes). Jia, Rakhlin and Xie proved that learning a policy from outcome rewards is statistically no harder than learning it from step rewards, up to polynomial factors in the horizon [97]. Theorem VIII.2 is about a different cost, that of certifying a chain with checks that are already given, and the two results are compatible: a learner may extract as much from outcomes as from steps, while a certifier that sees only outcomes pays exponentially in retries and in the reliability its check must have. Rohatgi and coauthors observe the same amplification of verifier error along a generation from the side of sampling [148].

IX. SOUND SINGULARITIES

A. Singular chains

Let the chain of Section VIII run forever: step $n \geq 1$ costs g_n per candidate to generate and a_n per run to check, is incorrect with probability ρ_n , and is checked by r_n runs of a sharp check of soundness ϵ , all steps independent. The total cost is $T_\infty = \sum_n T_n$ and F counts the steps whose kept candidate is incorrect.

Definition IX.1. The chain is *singular* if $\mathbb{E} T_\infty < \infty$, and *sound* if $F < \infty$ almost surely.

If cost is time, a singular chain completes infinitely many improvements in finite expected time, which is the vertical asymptote of the explosion literature [81, 128]; if cost is compute or energy, it completes them on a finite budget. Soundness asks that only finitely many of the improvements be wrong, so that from some step on the system is correct.

Proof of Theorem D. Put $\gamma = 1-\bar{\rho}$. By Lemma VIII.1, $g_n + a_n r_n \leq \mathbb{E} T_n \leq (g_n + a_n r_n)/\gamma$ and $\rho_* \epsilon^{r_n} \leq \pi_n \leq \epsilon^{r_n}/\gamma$. So $\mathbb{E} T_\infty < \infty$ if and only if $\sum (g_n + a_n r_n) < \infty$, and $\sum \pi_n < \infty$ if and only if $\sum \epsilon^{r_n} < \infty$.

(i) Let E_n be the event that the candidate kept at step n is incorrect, so $F = \sum \mathbf{1}_{E_n}$ and the E_n are independent with $\mathbb{P}(E_n) = \pi_n < 1$. If $\sum \pi_n < \infty$, then $\mathbb{E} F < \infty$, so $F < \infty$ almost surely (this is the first Borel–Cantelli lemma [32, 64]), and $\mathbb{P}(F=0) = \prod_n (1-\pi_n) > 0$, because a product of factors in

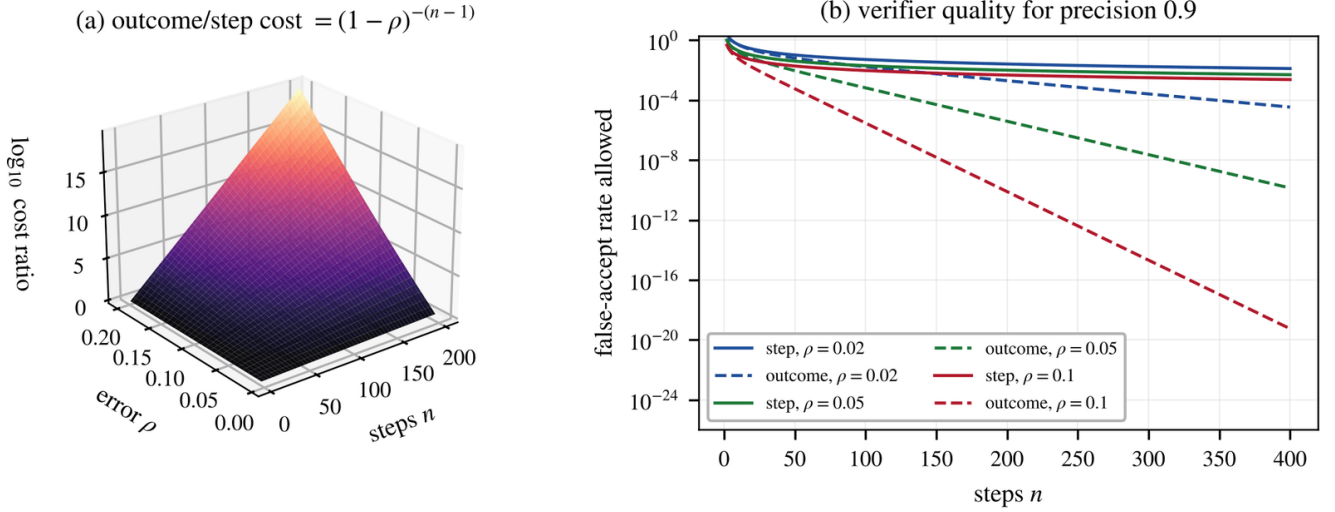


FIG. 7. Theorem VIII.2. (a) The ratio $(1 - \rho)^{-(n-1)}$ of the expected costs of outcome and step gating with perfect checks, on a logarithmic scale, for up to two hundred steps and error rates $\rho \leq 0.2$. (b) The false acceptance rate a check may have to reach precision 0.9 with $\eta = 0$: for step gating (solid lines and shaded bands between the sufficient and the necessary rate of part (iv)) it decreases like $1/n$, for outcome gating (dashed) like $(1 - \rho)^n$.

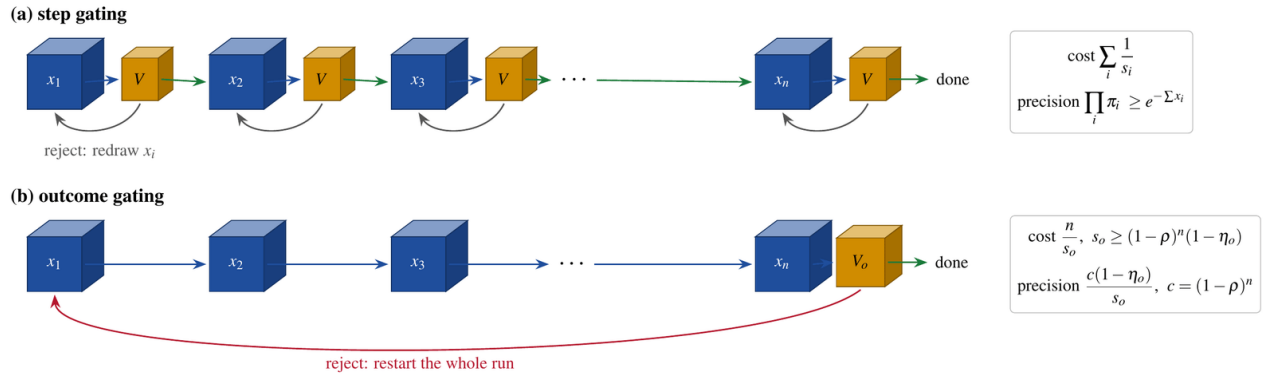


FIG. 8. The two protocols of Theorem VIII.2. (a) Step gating: each step is generated and checked, and regenerated until accepted; the cost is $\sum 1/s_i$ and the precision $\prod \pi_i \geq e^{-\sum x_i}$. (b) Outcome gating: the chain is generated to the end and checked once, and regenerated until accepted; the acceptance probability is at least $(1 - \rho)^n (1 - \eta_o)$, the cost is n/s_o and the precision $c(1 - \eta_o)/s_o$.

$(0, 1]$ converges to a positive limit when the sum of the complements converges. If $\sum \pi_n = \infty$, the second Borel–Cantelli lemma for independent events gives $\mathbb{P}(E_n \text{ infinitely often}) = 1$, so $F = \infty$ almost surely and $\mathbb{P}(F = 0) = 0$.

(ii) If every $r_n \geq 1$, then $\sum a_n r_n \geq \sum a_n$, so a singular chain needs $\sum g_n < \infty$ and $\sum a_n < \infty$; conversely $r_n = 1$ gives $\mathbb{E} T_\infty \leq \sum (g_n + a_n)/\gamma < \infty$.

(iii) A sound singular chain needs $\sum g_n < \infty$, $\sum a_n r_n < \infty$ and $\sum \varepsilon^n < \infty$. The last forces $r_n \geq 1$ for all large n , so $\sum a_n < \infty$, and Proposition VII.6 (applied to the tail on which $a_n \leq 1$) gives $\sum a_n \log(1/a_n) < \infty$. Conversely, under the three conditions, the schedule $r_n = \lceil \log(1/a_n)/L \rceil$ of Proposition VII.6, raised to at least one run, has $\sum a_n r_n < \infty$ and $\sum \varepsilon^n < \infty$. $\square$

The theorem has a simple form: the chain can reach the asymptote if the steps get cheap fast enough, and it can reach it soundly if they get cheap fast enough to pay for the en-

tropy. Since $a \log(1/a) \geq a$ for $a \leq 1/e$, the second condition is strictly stronger and the next corollary shows that the gap between them is not empty.

Corollary IX.2 (The soundness gap). *Let $a_n = 1/(n \log^\beta n)$ for $n \geq 3$, with $\sum g_n < \infty$. A singular chain with every step checked exists if and only if $\beta > 1$, and a sound singular chain exists if and only if $\beta > 2$. For $1 < \beta \leq 2$ every singular chain keeps infinitely many incorrect modifications almost surely.*

Proof. The series $\sum 1/(n \log^c n)$ converges if and only if $c > 1$, by Cauchy’s condensation test: its terms decrease, and 2^k times the term at $n = 2^k$ is $(k \log 2)^{-c}$ (Appendix A 6). So $\sum a_n < \infty$ if and only if $\beta > 1$. For the entropy, $\log(1/a_n) = \log n + \beta \log \log n$, which lies between $\log n$ and $2 \log n$ for all large n , so $a_n \log(1/a_n)$ lies between $1/(n \log^{\beta-1} n)$ and twice it, and $\sum a_n \log(1/a_n) < \infty$ if and only if $\beta - 1 > 1$. Theorem D

concludes; the last claim is part (i), since a singular chain with $1 < \beta \leq 2$ cannot have $\sum \varepsilon^n < \infty$. $\square$

Figure 9(b) shows the condensed partial sums of the two series for three values of β .

B. Power laws

The case that matters for growth models is a power law, $a_n = n^{-p}$ with $p > 1$. Then $A = \zeta(p)$, the profile has entropy

$$H(p) = \log \zeta(p) + p \frac{-\zeta'(p)}{\zeta(p)},$$

by (3), and $\sum a_n \log(1/a_n) = p \sum n^{-p} \log n < \infty$ for every $p > 1$. So every power law admits a sound singularity, and the question is its price.

Proposition IX.3 (Price of a power-law takeoff). *As $p \rightarrow 1^+$,*

$$\begin{aligned} H(p) &= \frac{1}{p-1} + \log \frac{1}{p-1} + O(1), \\ \frac{V^*(\theta)}{A} &= \frac{1}{(p-1)L} + O\left(\log \frac{1}{p-1}\right), \end{aligned}$$

for fixed $\theta \in (0, 1]$ and ε , so that $V^*(\theta) = \frac{1}{(p-1)^2 L} (1 + O((p-1) \log \frac{1}{p-1}))$.

Proof. Put $t = p - 1$. Comparing the decreasing function x^{-p} with its integral, $1/t = \int_1^\infty x^{-p} dx \leq \zeta(p) \leq 1 + 1/t$. The function $f(x) = x^{-p} \log x$ vanishes at 1, increases on $[1, e^{1/p}]$ and decreases afterwards, with maximum $1/(pe)$, so $|\sum_{n \geq 1} f(n) - \int_1^\infty f| \leq 1/(pe)$ (Appendix A 7); and $\int_1^\infty x^{-p} \log x dx = 1/t^2$. Hence $-\zeta'(p) = 1/t^2 + O(1)$ and

$$\frac{-p\zeta'(p)}{\zeta(p)} = \frac{(1+t)(t^{-2} + O(1))}{t^{-1}(1 + O(t))} = \frac{1}{t} + O(1)$$

and $\log \zeta(p) = \log \frac{1}{t} + O(t)$. This is the first formula. The second follows from Theorem A, Proposition V.7 and $A = 1/t + O(1)$. $\square$

The constant term is $1 - \gamma_E$, where γ_E is Euler's constant, as the Laurent expansion $\zeta(p) = 1/t + \gamma_E + O(t)$ shows; at $p = 1.1$ the three terms give 12.725 against the exact $H = 12.745$. So the cheapest sound takeoff along a power law near the critical exponent costs, per unit of the cost of one run per step, about $1/((p-1)L)$, and in total about $1/((p-1)^2 L)$: the soundness of the takeoff is paid for with one extra factor $1/(p-1)$ (Fig. 9(a)).

C. Polynomial checks and the phase diagram

With checks of order s , Corollary VII.5(ii) replaces the entropy condition by $\sum a_n^{1/(1+s)} < \infty$.

Corollary IX.4 (Phase diagram). *Let $a_n = n^{-p}$ and $\sum g_n < \infty$. A singular chain with every step checked exists if and only if $p > 1$. With exponential checks a sound one exists for every $p > 1$; with checks of order s a sound one exists if and only if $p > 1 + s$. For geometric costs $a_n = c^n$, $0 < c < 1$, a sound singular chain exists for every s .*

Proof. The series $\sum n^{-p/(1+s)}$ converges if and only if $p > 1 + s$, and the series $\sum c^{n/(1+s)}$ converges for every s . For the singular part, a check of order s with $\delta_n = 1$ costs a_n . $\square$

Figure 10 draws the phase diagram. Below the line $p = 1$ there is no singularity at all; between $p = 1$ and $p = 1 + s$ there are singular chains and all of them are unsound; above $p = 1 + s$ there are sound ones. Exponential checks are the edge $s = 0$ of the diagram, and the logarithmic family of Corollary IX.2 sits on the line $p = 1$, where the finer scale β splits it again.

X. CHECKING WITHOUT KNOWING THE FUTURE

The schedules of Sections V to IX are computed from the whole cost profile: the number of runs of the i -th check depends on A and on every other cost. A self-improving system does not know its future modifications. When the n -th candidate arrives, only $a_1, \dots, a_n$ are known, and the number of runs of its check must be fixed then. This section shows that such a system loses almost nothing: the entropy law survives with an additive term that grows like the logarithm of the logarithm of the dynamic range, and that term cannot be removed.

A. Online schedules

An *online schedule* is a rule that assigns to each finite sequence $(a_1, \dots, a_n)$ of positive costs a number $r_n \in \mathbb{N}$ of runs for the last candidate. Applied to a sequence $(a_n)_{n \geq 1}$, finite or infinite, it produces the schedule (r_n) , and r_n does not depend on $a_{n+1}, a_{n+2}, \dots$. The rule is θ -sound if $\sum_n \varepsilon^{r_n} \leq \theta$ for every sequence of positive costs. Write

$$S_n = a_1 + \dots + a_n, \quad u_n = \log \frac{S_n}{a_1}, \quad w_n = \frac{a_n/S_n}{2(1+u_n)^2},$$

so that $u_1 = 0$, u_n is nondecreasing and $w_1 = \frac{1}{2}$. On the first N candidates write $A_N = S_N$, $p^{(N)} = (a_n/A_N)_{n \leq N}$, $V_N = \sum_{n \leq N} a_n r_n$ and $V_N^*(\theta)$ for the offline optimum of Theorem A.

Theorem G (Online entropy law). *Let $0 < \varepsilon < 1$ and $0 < \theta \leq 1$. The online schedule*

$$r_n = \left\lceil \frac{1}{L} \log \frac{1}{\theta w_n} \right\rceil = \left\lceil \frac{1}{L} \log \frac{2(1+u_n)^2 S_n}{\theta a_n} \right\rceil \quad (5)$$

is θ -sound, and for every N

$$\begin{aligned} V_N &\leq \frac{A_N}{L} \left(H(p^{(N)}) + \log \frac{1}{\theta} \right) \\ &\quad + \frac{A_N}{L} \left(\log 2 + 2 \log \left(1 + \log \frac{A_N}{a_1} \right) \right) + A_N. \end{aligned} \quad (6)$$

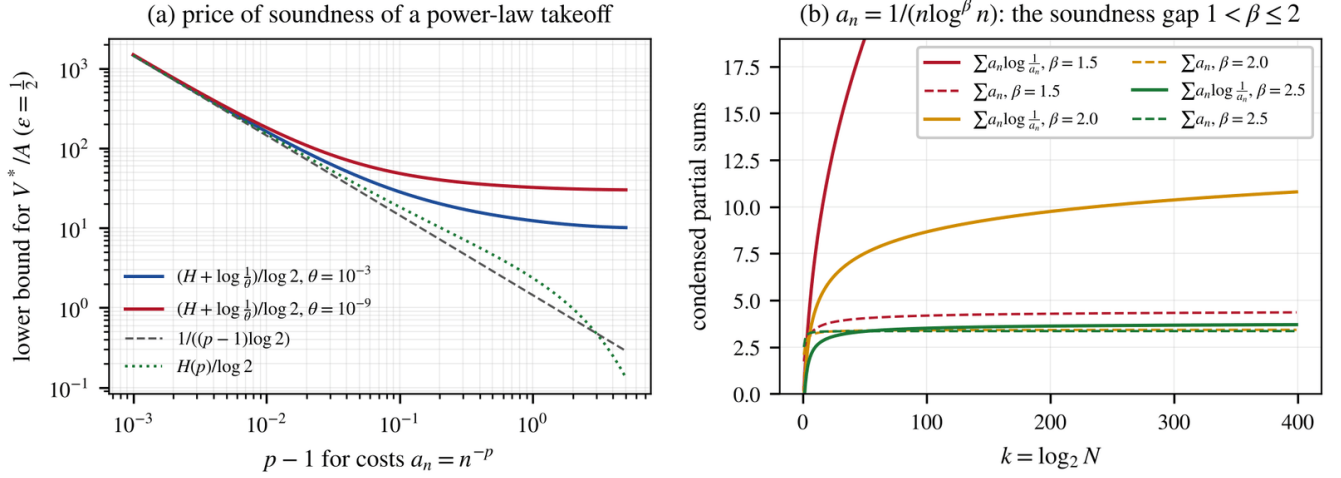


FIG. 9. (a) The lower bound $(H(p) + \log \frac{1}{\theta})/\log 2$ for V^*/A along the power laws $a_n = n^{-p}$, with $\varepsilon = \frac{1}{2}$ and $\theta = 10^{-3}, 10^{-9}$; as $p \rightarrow 1$ it approaches $1/((p-1)\log 2)$ (dashed), the term that dominates the entropy (dotted). (b) The soundness gap of Corollary IX.2: condensed partial sums of $\sum a_n$ (dashed) and $\sum a_n \log \frac{1}{a_n}$ (solid) for $a_n = 1/(n \log^\beta n)$. At $\beta = 1.5$ the cost converges and the entropy diverges; at $\beta = 2$ the entropy still diverges, like $\log \log N$; at $\beta = 2.5$ both converge.

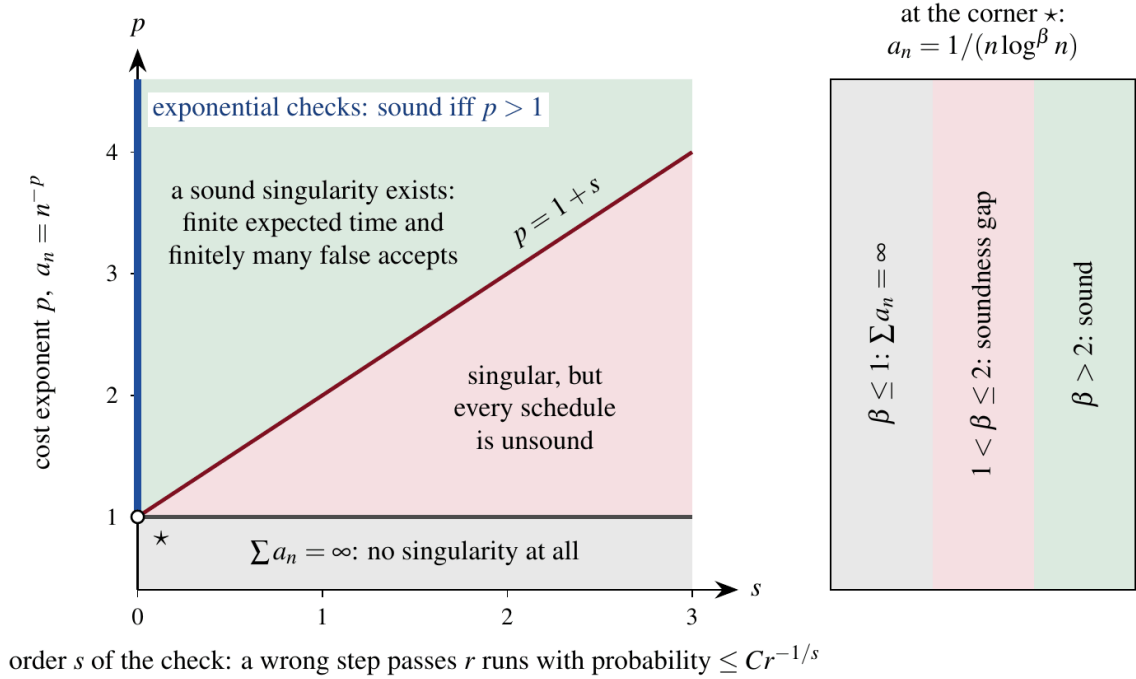


FIG. 10. The phase diagram of Corollary IX.4 in the plane of the polynomial order s of the checks and the exponent p of the cost law $a_n = n^{-p}$. No singularity for $p \leq 1$; singular chains that are all unsound for $1 < p \leq 1+s$; sound singular chains for $p > 1+s$. The axis $s = 0$ is the case of exponential checks, sound for every $p > 1$. The panel at the corner refines the line $p = 1$ by the logarithmic family $a_n = 1/(n \log^\beta n)$ of Corollary IX.2.

Conversely, on the doubling costs $a_n = 2^{n-1}$ every θ -sound online schedule satisfies, for infinitely many N ,

$$V_N - V_N^*(\theta) \geq \frac{A_N}{2L} \log \log_2 \left(1 + \frac{A_N}{a_1} \right) - \left(1 + \frac{2 \log 2}{L} \right) A_N. \quad (7)$$

Together with the lower bound of Theorem A, (6) says that the price of not knowing the future is at most $(A_N/L)(\log 2 + 2 \log(1 + \log(A_N/a_1))) + A_N$, and (7) says that a term of order $(A_N/L) \log \log(A_N/a_1)$ must be paid by every online rule on some sequences. The two bounds differ by a factor of four in the leading term. The dynamic range A_N/a_1 enters only

through a double logarithm: a system whose total checking cost grows by a factor 10^{100} over its first candidate pays at most $2\log(1 + 100\log 10) \approx 10.9$ extra runs per unit of cost, measured in units of L .

Corollary X.1 (Sound singularities without foresight). *If $\sum_n a_n < \infty$ and $\sum_n a_n \log(1/a_n) < \infty$, the online schedule (5) is θ -sound and has finite total cost*

$$V \leq \frac{A}{L} \left(H(p) + \log \frac{1}{\theta} + \log 2 + 2\log \left(1 + \log \frac{A}{a_1} \right) \right) + A.$$

So the criterion of Theorem D does not need an oracle. A chain that only knows its past costs reaches a sound singularity whenever any chain does, and the rule that achieves it is the same for every sequence.

B. Proofs

The proof rests on two elementary facts, both proved in Lean (Appendix B).

Lemma X.2 (Telescoping). *For $0 \leq x \leq y$, $\frac{y-x}{(1+y)^2} \leq \frac{1}{1+x} - \frac{1}{1+y}$. Hence for a nondecreasing sequence $0 \leq u_1 \leq u_2 \leq \dots$,*

$$\sum_{n=2}^N \frac{u_n - u_{n-1}}{(1+u_n)^2} \leq \frac{1}{1+u_1} - \frac{1}{1+u_N}.$$

Proof. The right side of the first inequality is $(y-x)/((1+x)(1+y))$, and $(1+x) \leq (1+y)$. Summing gives a telescoping series. $\square$

Lemma X.3 (Share of the newest cost). *For $0 < s \leq t$, $1 - s/t \leq \log(t/s)$. In particular $a_n/S_n = 1 - S_{n-1}/S_n \leq u_n - u_{n-1}$ for $n \geq 2$.*

Proof. This is $1 - 1/x \leq \log x$ at $x = t/s$, which is $\log y \leq y - 1$ at $y = 1/x$. For the second statement take $s = S_{n-1}$, $t = S_n$ and note $\log(S_n/S_{n-1}) = u_n - u_{n-1}$. $\square$

Proof of Theorem G. Soundness. By Lemmas X.3 and X.2,

$$\begin{aligned} \sum_{n=1}^N w_n &= \frac{1}{2} + \frac{1}{2} \sum_{n=2}^N \frac{a_n/S_n}{(1+u_n)^2} \leq \frac{1}{2} + \frac{1}{2} \sum_{n=2}^N \frac{u_n - u_{n-1}}{(1+u_n)^2} \\ &\leq \frac{1}{2} + \frac{1}{2} \left(1 - \frac{1}{1+u_N} \right) < 1 \end{aligned}$$

for every N and every sequence of positive costs. Since $w_n \leq \frac{1}{2} < 1$ and $\theta \leq 1$, the number inside the ceiling of (5) is positive and $\epsilon^{r_n} \leq \theta w_n$. Hence $\sum_n \epsilon^{r_n} \leq \theta \sum_n w_n \leq \theta$, and r_n depends only on $a_1, \dots, a_n$.

Cost. From $r_n < \frac{1}{L} \log \frac{1}{\theta w_n} + 1$,

$$\begin{aligned} LV_N &< \sum_{n \leq N} a_n \left(\log \frac{1}{\theta} + \log 2 + 2\log(1+u_n) \right) \\ &\quad + \sum_{n \leq N} a_n \log \frac{S_n}{a_n} + LA_N. \end{aligned}$$

For $n \leq N$ we have $S_n \leq A_N$, so $\log(S_n/a_n) \leq \log(A_N/a_n)$ and $u_n \leq \log(A_N/a_1)$. Since $\sum_{n \leq N} a_n \log(A_N/a_n) = A_N H(p^{(N)})$, (6) follows.

Lower bound. Fix a θ -sound online rule and apply it to the infinite sequence $a_n = 2^{n-1}$. Its decisions on the first N candidates are those it takes on the finite sequence $(a_1, \dots, a_N)$. Put $v_n = \epsilon^{r_n}/\theta$, so $\sum_n v_n \leq 1$. Then $v_N \leq 1/N$ for infinitely many N : otherwise $v_N > 1/N$ for all large N and $\sum v_N$ would diverge with the harmonic series. For such N , $r_N L \geq \log(N/\theta)$, and $r_n L \geq \log(1/\theta)$ for every n because $\epsilon^{r_n} \leq \theta$. Hence

$$LV_N \geq A_N \log \frac{1}{\theta} + a_N \log N.$$

On the other side, $p_n^{(N)} = 2^{n-1}/(2^N - 1) \geq 2^{n-1-N}$, so

$$\begin{aligned} H(p^{(N)}) &\leq \log 2 \sum_{n=1}^N (N - n + 1) p_n^{(N)} \\ &= \log 2 \cdot \frac{2^{N+1} - N - 2}{2^N - 1} \leq 2\log 2, \end{aligned}$$

by the identity $\sum_{k=1}^N k 2^{-k} = 2 - (N+2)2^{-N}$, and Theorem A gives $LV_N^*(\theta) \leq A_N(2\log 2 + \log \frac{1}{\theta}) + LA_N$. Subtracting, and using $a_N = 2^{N-1} \geq A_N/2$ and $N = \log_2(1 + A_N/a_1)$, gives (7). $\square$

Proof of Corollary X.1. Soundness is that of Theorem G. For the cost, the argument of the upper bound with A in place of A_N bounds every partial sum V_N by the right side, since $S_n \leq A$ for all n and $\sum_{n \leq N} a_n \log(A/a_n) \leq AH(p)$. The right side is finite because $AH(p) = A \log A + \sum_n a_n \log(1/a_n)$. $\square$

C. Universal codes

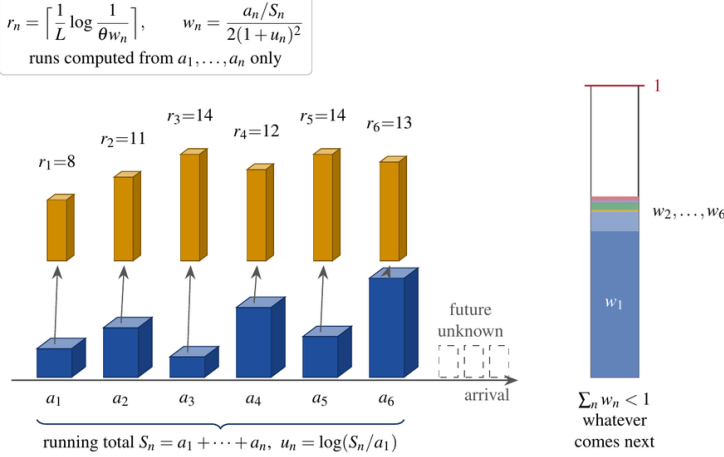
Theorem G is the schedule counterpart of the universal codes for the integers. Elias constructed prefix codes whose length for the integer m is $\log m + O(\log \log m)$ without knowledge of the distribution of m [68], and Rissanen derived from them a universal prior whose code length $\log^* m$ is optimal up to the iterated logarithms [145]. Here the role of the integer is played by the ratio S_n/a_1 , the weight w_n is a code length in disguise, and the double logarithm of (6) is the price of the second level of the Elias code. The constant 2 in front of it can be lowered to $1 + \delta$ for every $\delta > 0$ by replacing $(1+u_n)^2$ with $c_\delta(1+u_n)(1+\log(1+u_n))^{1+\delta}$, as in the Elias ω code; the lower bound (7) shows that no rule removes the double logarithm.

Figure 11 shows the rule at work. On random profiles its cost stays within the band of Theorem G, and on the doubling sequence the excess over the offline optimum grows like $\log \log$ of the dynamic range, as the lower bound requires.

XI. SUBAGENTS, CASCADES AND THE MOVES OF A PROOF

A working BLLM is not one generator and one check. A coordinator hands parts of a problem to subagents, each sub-

(a) checks fixed on arrival



(b) cost against dynamic range

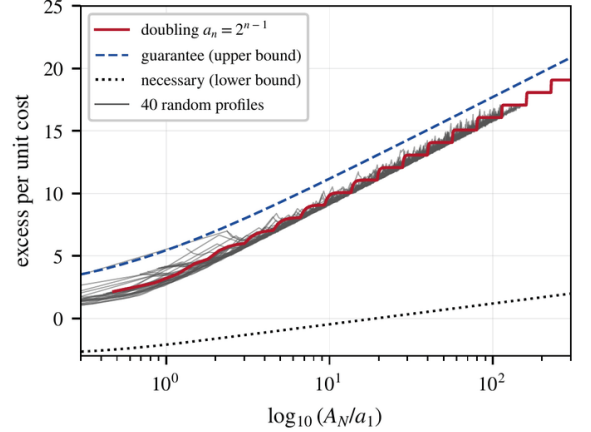


FIG. 11. The online entropy law. (a) Candidates arrive one at a time; the n -th check is given r_n runs computed from the running total S_n alone, and the weights w_n never sum to more than one, whatever comes next. (b) The excess $(V_N - \frac{A_N}{L}(H + \log \frac{1}{\theta}))/A_N$ of the online rule over the offline lower bound, for the doubling sequence $a_n = 2^{n-1}$ (red) and for forty random profiles with log-normal costs (grey), against $\log_{10}(A_N/a_1)$ on a logarithmic axis, with $\varepsilon = \frac{1}{2}$ and $\theta = 10^{-6}$. The dashed curve is the guarantee $(\log 2 + 2\log(1 + \log(A_N/a_1)))/L + 1$ of (6), and the dotted curve is the necessary excess $\frac{1}{2L} \log \log_2(1 + A_N/a_1) - 1 - \frac{2\log 2}{L}$ of (7).

agent proposes candidates by its own method, and the candidates meet checks of very different cost: a test that runs in a millisecond, a randomized identity test, a full proof checker. This section proves the two laws that govern such a system. Splitting the work among subagents costs nothing, provided each receives a share of the budget proportional to its total cost, and a misallocated budget is charged exactly a Kullback–Leibler divergence (Theorem H). Cheap unreliable checks placed before an exact one make the exact check paid, up to a logarithmic premium, only on the correct candidates (Theorem XI.1). The section ends with the six moves by which the subagents of Section G propose candidates.

A. Subagents

Partition the candidates into groups $G_1, \dots, G_m$, one for each subagent. Write $A_k = \sum_{i \in G_k} a_i$, $P_k = A_k/A$ for the share of the k -th subagent and $p^{(k)} = (a_i/A_k)_{i \in G_k}$ for its own cost profile. A coordinator gives the k -th subagent a budget $\theta_k > 0$ with $\Theta = \sum_k \theta_k \leq \theta$, and the subagent schedules its checks with any θ_k -sound schedule. The union of the schedules is θ -sound, since the expected numbers of false acceptances add.

Theorem H (Subagents). (i) Chain rule. $H(p) = H(P) + \sum_k P_k H(p^{(k)})$.

(ii) Price of a split. The sum over the subagents of the lower bounds of Theorem A for their groups exceeds the lower bound for the whole profile by exactly

$$\frac{A}{L} \left(D(P \| q) + \log \frac{\theta}{\Theta} \right), \quad q_k = \frac{\theta_k}{\Theta}, \quad (8)$$

where D is the Kullback–Leibler divergence. It vanishes if and only if $\theta_k = \theta P_k$ for every k .

- (iii) Coordination by one number. If $\theta_k = \theta P_k$, the optimal schedule of Theorem A computed by the k -th subagent from its own costs and the single number θP_k coincides on G_k with the optimal schedule of the whole profile. The union is θ -sound and costs at most $\frac{A}{L}(H(p) + \log \frac{1}{\theta}) + A$.
- (iv) Hierarchies. For a tree of subagents in which every node passes to each child the share of its own budget proportional to the child's total cost, the schedule at the leaves is the optimal flat schedule, and

$$AH(p) = \sum_v A_v H(P^v), \quad (9)$$

the sum over the internal nodes v of the tree, where A_v is the total cost below v and P^v the distribution of A_v among the children of v .

Proof. (i) For $i \in G_k$, $\log(A/a_i) = \log(A/A_k) + \log(A_k/a_i)$; multiply by a_i , sum over $i \in G_k$ and then over k , and divide by A .

(ii) The sum of the lower bounds of the groups is

$$\frac{1}{L} \sum_k \left(A_k H(p^{(k)}) + A_k \log \frac{1}{\theta_k} \right).$$

By (i), subtracting $\frac{A}{L}(H(p) + \log \frac{1}{\theta})$ leaves

$$\begin{aligned} & \frac{1}{L} \left(\sum_k A_k \log \frac{1}{\theta_k} - AH(p) - A \log \frac{1}{\theta} \right) \\ &= \frac{A}{L} \sum_k P_k \log \frac{\theta P_k}{\theta_k} = \frac{A}{L} \left(\sum_k P_k \log \frac{P_k}{q_k} + \log \frac{\theta}{\Theta} \right). \end{aligned}$$

Both terms are nonnegative, the first by Gibbs' inequality (Lemma V.1), and they vanish together exactly when $q = P$ and $\Theta = \theta$, that is, when $\theta_k = \theta P_k$.

(iii) The ceiling schedule of the k -th group at budget θP_k gives $i \in G_k$ the number of runs $\lceil \frac{1}{L} \log \frac{A_k}{\theta P_k a_i} \rceil = \lceil \frac{1}{L} \log \frac{A}{\theta a_i} \rceil$, which is the flat ceiling schedule; its soundness and cost are those of Theorem A.

(iv) By induction on the depth, applying (iii) at the root and the induction hypothesis to each subtree with its proportional budget. The identity (9) follows by applying (i) at every internal node. $\square$

The theorem has a practical reading. A coordinator need not know the costs of the individual checks of its subagents, and a subagent need not know anything about the others: one number per subagent, its share θP_k of the budget, is enough to reach the global optimum, and the identity (9) says that the entropy tax is paid node by node along the tree of delegation, each node paying for the uncertainty of its own split. A coordinator that splits the budget evenly among m subagents instead pays the extra $\frac{A}{L} D(P \| u) = \frac{A}{L} (\log m - H(P))$, where u is uniform, and this is unbounded when a few subagents carry almost all the cost.

B. Weak checks before strong ones

Let a fraction $\rho \in (0, 1)$ of the candidates be incorrect. A *strong check* costs c and is exact: it accepts the correct candidates and rejects the others, as a proof checker does on a formal proof. A *weak check* costs b per run, has perfect completeness and soundness ε : a language model asked to find a flaw, a numerical test at random points, a type check. A *cascade* with r weak runs runs the weak check r times on every candidate and the strong check on the candidates that pass them. It keeps exactly the correct candidates, whatever r , so its soundness is that of the strong check alone.

Theorem XI.1 (Weak before strong). *If the weak check attains its soundness, the expected cost of a cascade with r weak runs per candidate is*

$$E(r) = br + c(1 - \rho) + c\rho\varepsilon^r,$$

and for $b \leq c\rho L$,

$$M \leq \min_{r \in \mathbb{N}} E(r) \leq M + b, \quad M = c(1 - \rho) + \frac{b}{L} \left(1 + \log \frac{c\rho L}{b} \right), \quad (10)$$

the upper bound being attained up to b by $r = \lceil \frac{1}{L} \log \frac{c\rho L}{b} \rceil$. Without weak checks the cost is c .

Proof. A correct candidate passes the r weak runs and reaches the strong check; an incorrect one reaches it with probability ε^r . This gives $E(r)$. Put $y = c\rho\varepsilon^r$, so that $r = \frac{1}{L} \log \frac{c\rho}{y}$ and

$$br + c\rho\varepsilon^r - \frac{b}{L} \left(1 + \log \frac{c\rho L}{b} \right) = \frac{b}{L} \left(\frac{yL}{b} - 1 - \log \frac{yL}{b} \right) \geq 0$$

by $\log x \leq x - 1$. This is the lower bound. For the upper bound let $x = \frac{1}{L} \log \frac{c\rho L}{b} \geq 0$ and $r = \lceil x \rceil$. Then $br \leq bx + b$ and $\varepsilon^r \leq \varepsilon^x = b/(c\rho L)$, so $c\rho\varepsilon^r \leq b/L$. $\square$

The strong check is thus paid on a fraction $1 - \rho$ of the candidates, the correct ones, plus a premium logarithmic in c/b . When most proposals are wrong, as for a language model that guesses the next step of a hard proof, the saving is close to the factor $1/(1 - \rho)$. With a proof checker of cost $c = 1000$, a weak check of cost $b = 1$ and soundness $\frac{1}{2}$, and nine proposals in ten wrong, the best cascade runs nine weak checks per candidate and costs 110.8 against 1000; the lower bound is 110.7.

The proof shows something more important than the saving. The soundness of the cascade does not depend on ε : a weak check, however unreliable, can only remove candidates, never admit one. A learned critic may therefore be used freely as a filter, and only as a filter; the decision belongs to the exact check. The same holds with a randomized strong check of soundness ε_s given r_s runs, whose false acceptances are bounded by $\rho\varepsilon^r\varepsilon_s^{r_s} \leq \rho\varepsilon_s^{r_s}$, so the weak runs may also be counted towards the budget of Theorem A but are never needed for it.

C. The moves of a proof

The formal proofs of Section G are built from a small number of moves, used over and over. We name six, and the BLLM package of Section G has one subagent for each. A move is a way of generating candidates; it carries no authority of its own, and every candidate it produces meets the same checks, so Theorems A to H and XI.1 apply to any mixture of moves.

Construction produces an explicit object, a flow, a lattice, a cycle, a schedule, together with the statement that it has the required property, which is then checked. *Strong then weak* proposes the strongest statement first and, when it is rejected, the next rung of a ladder of weakenings: a special case, a bounded range, a conditional form. Each rung is a candidate and enters the cost profile of Theorem A. *Residual* turns a partial success into a new goal, the part that remains unproved; it is the retry of Theorem F, and when the corrections go on without end, as in a construction corrected to every order, whether the infinite sequence can be certified at finite cost is decided by Theorem D and, without knowledge of the future corrections, by Corollary X.1. *Trivialization* specialises the parameters of a goal until it becomes a finite computation, which a kernel decides exactly, so that its check has soundness zero. *Bypass* replaces a step by a result already kept in memory, whose check has been paid once. *Blow-up* searches for a singular object or a counterexample.

Remark XI.2 (Refutations carry no entropy tax). A counterexample checked by exact evaluation needs one run, whatever the budget, because its check has soundness zero; so do the finite computations produced by trivialization and the results reused by bypass. In a mixed profile the entropy of Theorem A is therefore the entropy of the candidates whose checks are randomized, with total cost A_{rand} , and the exact checks add their cost once. A system that turns more of its work into exact checks lowers both A_{rand} and the entropy it pays on it.

Figure 12 shows the arrangement: a coordinator, six sub-agents, a cascade of weak and strong checks, and a memory that stores every kept result as a note linked to the results it uses.

XII. THE GAP BETWEEN HUMAN AND LLM THINKING, AND HOW A BLLM CLOSES IT

The theorems so far price the checking of what a proposer proposes. They are silent on the proposer itself, and a sound system whose proposer never reaches the right idea is sound and useless. Chen, Zhao and Cohan have measured how far the ideas of language models are from those of human researchers [55].⁵ Other measurements of the same gap, between the language a model produces and the reasoning it is asked for [104] and between human and model reasoning on naturalistic detective problems [105], point the same way. This section turns the measurement into a law: it computes, from the measured profiles, how many proposals a model needs to reach the kind of idea a problem calls for, proves the least possible number, and shows that a selector which learns from the checks reaches that least number.

A. The gap law

Ideas fall into K classes. A problem needs an idea of class j with probability π_j , its *need*, and a proposer draws the classes of its proposals independently from a profile q until it draws the needed class. The expected number of proposals is

$$C(\pi, q) = \sum_j \pi_j \mathbb{E}[\text{draws until } j] = \sum_j \frac{\pi_j}{q_j}. \quad (11)$$

Write $S = \sum_j \sqrt{\pi_j}$, $r_j = \sqrt{\pi_j}/S$ and $H_{1/2}(\pi) = 2 \log S$, the Rényi entropy of order one half [144]; $\chi^2(r||q) = \sum_j (r_j - q_j)^2/q_j$ is the chi-squared divergence.

Theorem I (Gap law). *For every need π and every profile q with $q_j > 0$,*

- (i) $C(\pi, q) \geq S^2 = e^{H_{1/2}(\pi)}$, with equality exactly at $q = r$, square-root sampling;
- (ii) $C(\pi, q) = e^{H_{1/2}(\pi)} (1 + \chi^2(r||q))$;
- (iii) for every $\lambda \in (0, 1]$ the mixture $q_\lambda = (1 - \lambda)q + \lambda r$ has $C(\pi, q_\lambda) \leq e^{H_{1/2}(\pi)}/\lambda$, whatever q is;
- (iv) a proposer that copies the need, $q = \pi$, pays $K \geq S^2$, with equality only for the uniform need.

Proof. (i) By the Cauchy–Schwarz inequality,

$$S = \sum_j \sqrt{\frac{\pi_j}{q_j}} \sqrt{q_j} \leq \left(\sum_j \frac{\pi_j}{q_j} \right)^{1/2} \left(\sum_j q_j \right)^{1/2} = C(\pi, q)^{1/2},$$

with equality if and only if $\sqrt{\pi_j/q_j}$ is proportional to $\sqrt{q_j}$, that is $q_j \propto \sqrt{\pi_j}$. (ii) Since $\pi_j = S^2 r_j^2$,

$$\begin{aligned} C(\pi, q) &= S^2 \sum_j \frac{r_j^2}{q_j} = S^2 \left(1 + \sum_j \frac{r_j^2}{q_j} - 2 \sum_j r_j + \sum_j q_j \right) \\ &= S^2 (1 + \chi^2(r||q)). \end{aligned}$$

(iii) Coordinatewise $q_{\lambda,j} \geq \lambda r_j$, so $C(\pi, q_\lambda) \leq \sum_j \pi_j / (\lambda r_j) = S \sum_j \sqrt{\pi_j} / \lambda = S^2 / \lambda$. (iv) $C(\pi, \pi) = \sum_j 1 = K$, and $S^2 \leq K \sum_j \pi_j = K$ by the Cauchy–Schwarz inequality, with equality only when all π_j are equal. $\square$

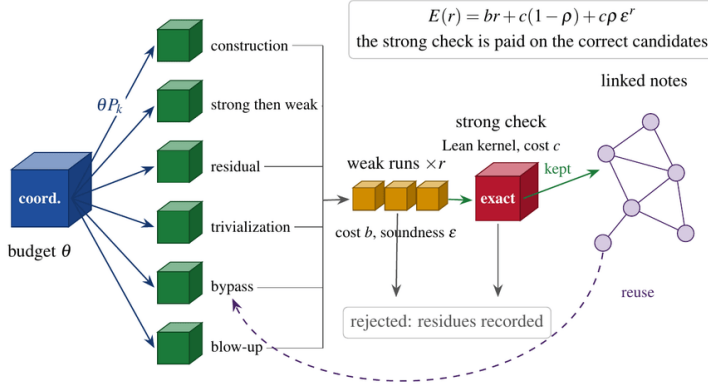
Parts (i) to (iii) are proved in Lean (Appendix B). Square-root sampling is the rule found by Press for the screening of rare cases [138]; here it answers a different question, and its meaning is plain. A proposer that follows the needs too closely wastes its proposals on the common classes, which are found quickly anyway, and a proposer that ignores them never reaches the rare ones; the optimum spends on each class in proportion to the square root of its need. Part (iv) says that even a proposer with perfect human taste, one that proposes as often as humans succeed, is not optimal: it pays K against S^2 . The gap of a proposer is the chi-squared divergence of its profile from square-root sampling, and part (iii) says that it closes at once, by a factor that does not depend on the proposer, as soon as a fixed fraction of the proposals is drawn from the right profile.

B. The gap, measured

We take as need the human profile: the class of idea a problem required is the class its authors used. Table V gives the cost (11) of each model of [55] on both axes, the cost of the half-and-half mixture with square-root sampling, and the least weight λ on square-root sampling that brings the cost within 10% of the optimum. On the axis of opportunities the optimum is $S^2 = 6.503$ proposals per problem and the human profile pays 7; the models pay between 12.69 (Gemini 3.1 Pro) and 27.81 (GPT-OSS-120B), from 1.95 to 4.28 times the optimum. Nearly all of the gap comes from one class: the models propose “bridge opportunities”, the transfer of a method

⁵ Z. Chen, Y. Zhao and A. Cohan, *Measuring the gap between human and LLM research ideas*, preprint (1 July 2026), arXiv:2607.01233 [55]. For each of 11,683 papers of ICLR, ICML and NeurIPS (2023 to 2026) and of Nature Communications (2023 to 2025) they reconstruct a small set of prior works that likely inspired its core idea, ask a language model for a new idea from the titles and summaries of those works, and classify the human idea and the generated ones on two axes of seven classes: the kind of research opportunity (puzzle or contradiction, explanation gap, scope mismatch, evidence gap, bridge opportunity, failure or risk gap, resource bottleneck) and the kind of method (synthesis or unification, relaxing or extending the scope, robustification, formal derivation, empirical mapping, artifact or system, optimisation or search). They conclude that the range of the generated ideas is narrower than, and systematically shifted from, the human one. The shares used here are those of their Figure 3, read from the vector paths of the figure to 0.1 percentage points and checked against the counts of their Table 11 and the distances of their Table 1.

(a) coordinator, subagents, cascade, memory



(b) weak runs before one exact check

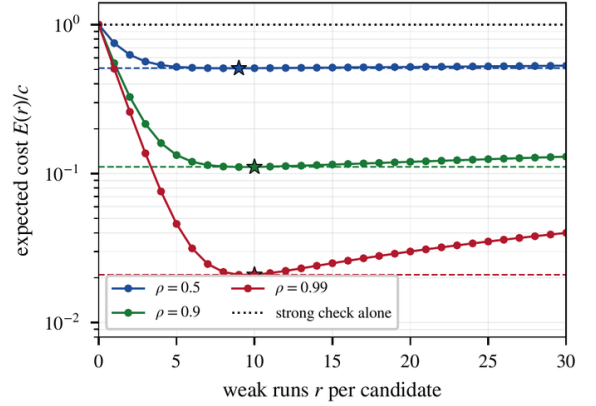


FIG. 12. Subagents and cascades. (a) A coordinator passes to each of six subagents, one for each move of Section XIC, the share θP_k of the budget proportional to its total cost (Theorem H); every candidate meets a cascade of r weak runs and one exact strong check (Theorem XI.1), and the kept results are stored as linked notes that the bypass move reuses. (b) The expected cost $E(r)/c$ of a cascade against the number r of weak runs, for $b/c = 10^{-3}$, $\varepsilon = \frac{1}{2}$ and $\rho \in \{0.5, 0.9, 0.99\}$; stars mark the number of runs of Theorem XI.1, the lower bound M of (10) is dashed and the cost 1 of the strong check alone dotted.

from one field to another, in 47 to 64% of their ideas (Gemini 3.1 Pro 46.9%, GPT-OSS-120B 64.2%) against 12% of the human ones, and evidence gaps and scope mismatches, 28% of the human ideas, in 5 to 11% (Fig. 13(a)). On the axis of methods the models pay from 1.51 to 3.28 times the optimum, with the excess in synthesis and unification.

Two facts in the table are worth stating separately. The ranking by cost is not the ranking by distance: on the axis of opportunities Qwen3-8B is closer to the human profile than Qwen3-32B and DeepSeek V4 Flash and costs 13% and 38% more, and on the axis of methods GPT-5.4 mini is farther from it than DeepSeek V4 Pro and costs less. The cost weighs a class by its need over its share, so a class that a model almost never proposes costs it far more than the distance records. And the mixtures of the table close most of the gap at a moderate price: half the proposals from square-root sampling bring every model within 1.35 times the optimum on opportunities and 1.25 times on methods, and a weight between 0.48 and 0.74 brings it within 10%.

C. Learning the need from the checks

Square-root sampling needs π , and π is not given. A BLLM learns it from its own record: every result its checks keep closes a problem with an idea of a known class, so after n closed problems it has counts c_j with $\sum_j c_j = n$, and it samples from the square root of the estimate $\hat{\pi}_j = (c_j + \alpha)/(n + K\alpha)$, with a prior count $\alpha > 0$. The next proposition says how fast this closes the gap.

Proposition XII.1 (Learning rate). *Let the counts be multinomial with parameters n and π , all $\pi_j > 0$, and let $\hat{r}$ be square-*

root sampling of $\hat{\pi}$. Then

$$\frac{\mathbb{E}C(\pi, \hat{r})}{e^{H_{1/2}(\pi)}} = 1 + \frac{\beta(\pi)}{4n} + o\left(\frac{1}{n}\right),$$

$$\beta(\pi) = \frac{1}{S} \sum_j \frac{1}{\sqrt{\pi_j}} - \frac{K}{S^2} \geq \frac{K(K-1)}{S^2}. \quad (12)$$

Proof. By Theorem I(ii) the ratio is $1 + \mathbb{E}\chi^2(r\|\hat{r})$. Write $\hat{\pi} = \pi + d$. The map $x \mapsto \chi^2(r\|\rho(x))$, with $\rho_j(x) = \sqrt{x_j}/\sum_k \sqrt{x_k}$, is smooth near π , vanishes at π with its gradient, and its second-order part is computed from $\rho_j(\pi + d) - r_j = (u_j - r_j U)/S + O(|d|^2)$, where $u_j = d_j/(2\sqrt{\pi_j})$ and $U = \sum_j u_j$:

$$\begin{aligned} \chi^2(r\|\rho(\pi + d)) &= \frac{1}{S} \sum_j \frac{(u_j - r_j U)^2}{\sqrt{\pi_j}} + O(|d|^3) \\ &= \frac{1}{S} \left(\sum_j \frac{u_j^2}{\sqrt{\pi_j}} - \frac{U^2}{S} \right) + O(|d|^3), \end{aligned}$$

using $r_j/\sqrt{\pi_j} = 1/S$ and $\sum_j r_j^2/\sqrt{\pi_j} = 1/S$. The prior shifts the mean of d by $O(1/n)$, and the covariance of d is $(\text{diag}(\pi) - \pi\pi^\top)/n + O(n^{-2})$; so $\mathbb{E}u_j^2 = (1 - \pi_j)/(4n) + O(n^{-2})$ and $\mathbb{E}U^2 = \frac{1}{4n} (\sum_j (1 - \pi_j) - \sum_{j \neq k} \sqrt{\pi_j \pi_k}) + O(n^{-2}) = \frac{K-S^2}{4n} + O(n^{-2})$. Substituting,

$$\mathbb{E}\chi^2 = \frac{1}{4nS} \left(\sum_j \frac{1 - \pi_j}{\sqrt{\pi_j}} - \frac{K - S^2}{S} \right) + o\left(\frac{1}{n}\right) = \frac{\beta(\pi)}{4n} + o\left(\frac{1}{n}\right),$$

since $\sum_j (1 - \pi_j)/\sqrt{\pi_j} = \sum_j \pi_j^{-1/2} - S$. The remainder is $o(1/n)$: on the event $|d| \leq \delta$ it is bounded by a constant times $\mathbb{E}|d|^3 = O(n^{-3/2})$, and off it $\chi^2(r\|\hat{r})$ is at most a polynomial in n , because $\hat{\pi}_j \geq \alpha/(n + K\alpha)$, while the event

TABLE V. The gap, from the data of [55], with the human profile as need. “Cost” is the expected number of proposals per problem (11); “ratio” is the cost over the optimum $e^{H_{1/2}(\pi)}$, equal to $1 + \chi^2(r||q)$ by Theorem I(ii); “TVD” is the total variation distance from the human profile; “50/50” is the cost of the mixture with weight $\frac{1}{2}$ on square-root sampling, at most $2e^{H_{1/2}(\pi)}$ by Theorem I(iii); “steer” is the least weight on square-root sampling that brings the cost within 10% of the optimum, at most $1/1.1 = 0.909$ for every proposer.

Proposer	research opportunity, optimum 6.503					method, optimum 6.445				
	cost	ratio	TVD	50/50	steer	cost	ratio	TVD	50/50	steer
Human profile	7.00	1.08	0.000	6.62	0.00	7.00	1.09	0.000	6.57	0.00
Claude Sonnet 4.6	14.14	2.17	0.352	7.73	0.64	9.74	1.51	0.211	7.04	0.48
Gemini 3.1 Pro	12.69	1.95	0.349	7.66	0.63	10.96	1.70	0.227	7.15	0.52
GPT-OSS-20B	23.70	3.64	0.457	8.48	0.72	21.15	3.28	0.378	7.89	0.66
GPT-OSS-120B	27.81	4.28	0.522	8.74	0.74	20.94	3.25	0.390	7.90	0.66
GPT-5.4 mini	27.69	4.26	0.511	8.61	0.73	11.84	1.84	0.339	7.45	0.60
Qwen3-8B	23.38	3.59	0.383	8.17	0.69	20.99	3.26	0.368	8.00	0.68
Qwen3-32B	20.67	3.18	0.418	8.26	0.70	19.78	3.07	0.364	7.93	0.67
DeepSeek V4 Flash	16.89	2.60	0.401	8.03	0.68	12.93	2.01	0.246	7.21	0.54
DeepSeek V4 Pro	24.35	3.74	0.437	8.30	0.70	12.11	1.88	0.258	7.38	0.58
Square-root sampling	6.50	1.00	0.099	6.50	0.00	6.44	1.00	0.119	6.44	0.00

(a) shares of seven kinds of research opportunity

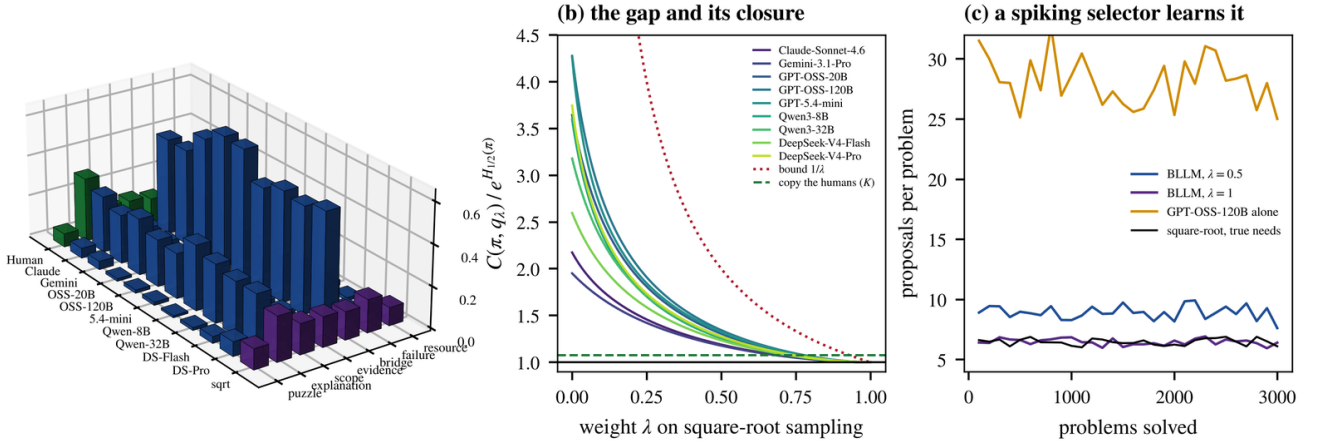


FIG. 13. The gap between human and LLM research ideas and its closure. (a) The shares of the seven kinds of research opportunity in the human papers and in the ideas of nine language models [55], and square-root sampling of the human profile (“sqrt”); the models concentrate on bridge opportunities. (b) The cost of the mixture q_λ of each model with square-root sampling, divided by the optimum $e^{H_{1/2}(\pi)}$, against the weight λ : every curve lies below the bound $1/\lambda$ of Theorem I(iii), dotted, and falls below the cost of the human profile, dashed, at a weight between 0.68 and 0.78. (c) A stream of 3000 problems with the opportunity profile of GPT-OSS-120B, mean proposals per problem over windows of 100 problems and six seeds: the model alone pays 28.1 against the 27.81 of the law; a BLLM that draws half of its proposals from the spiking selector of Section IIID, which learns only from the problems its checks close, pays 8.9, the value 8.74 of the law; with all proposals from the selector it pays 6.45, the optimum 6.50 of square-root sampling with the true needs.

has probability e^{-cn} by Hoeffding’s inequality [91]. Finally $\sum_j \pi_j^{-1/2} \geq K^2/S$ by the Cauchy–Schwarz inequality, which gives the lower bound on β . $\square$

For the human profile of opportunities $\beta = 7.06$, so a BLLM that has closed 100 problems pays 1.018 times the optimum, and after 1000 problems 1.0018 times; simulation gives 1.019 and 1.0018, and at $n = 20$, where the expansion is not yet accurate, 1.083. The correction is of order K^2/n and does not depend on the proposer at all: the gap of a language model is a fixed multiple, the gap of a learning selector disappears like $1/n$.

D. A spiking selector

The selector that the BLLM package uses is a circuit of spiking neurons, chosen because it computes square-root sampling exactly with the elements and the learning rule that nervous systems use for the selection of actions [71, 146, 154] and that neuromorphic hardware implements [63, 114, 117]. There is one leaky integrate-and-fire neuron for each class, or for each agent; the membrane potential u_k of neuron k relaxes with time constant τ to its synaptic drive w_k , and the neuron fires in continuous time with the escape rate $\rho_0 e^{(u_k - \vartheta)/\delta}$ [77]. The first neuron to fire chooses. When a check keeps a result produced by class k , a reward, the third factor of a three-factor

rule [71], potentiates that synapse,

$$w_k \leftarrow w_k + \frac{\delta}{2} \log \frac{c_k + 1 + \alpha}{c_k + \alpha}, \quad c_k \leftarrow c_k + 1, \quad (13)$$

and synaptic scaling [169] then shifts all weights by a common amount that keeps the total rate equal to ρ_0 .

Proposition XII.2 (The selector samples the square root of what it has learned). *Once the membranes have relaxed, the neuron that fires first is k with probability $e^{w_k/\delta} / \sum_l e^{w_l/\delta}$, and after the rewards (13) this probability is $\sqrt{c_k + \alpha} / \sum_l \sqrt{c_l + \alpha}$, square-root sampling of the estimated need $\hat{\pi}$.*

Proof. With relaxed membranes, $u_k = w_k$ and the neurons fire as independent Poisson processes of rates $\lambda_k = \rho_0 e^{(w_k - \vartheta)/\delta}$. The first spike of neuron k is exponential with rate λ_k , and the least of independent exponential times is the k -th with probability $\lambda_k / \sum_l \lambda_l$. The increments (13) telescope, so after c_k rewards $w_k = w_k(0) + \frac{\delta}{2} \log \frac{c_k + \alpha}{\alpha}$, and the scaling adds the same constant to every w_l , which cancels in the ratio. With equal initial weights, $e^{w_k/\delta}$ is proportional to $\sqrt{c_k + \alpha}$. $\square$

In the package the membranes are integrated with a step dt and the race opens after five time constants, so the probabilities hold up to terms of order $\rho_0 dt$ and e^{-5} ; over 40,000 races they agree with $\sqrt{c_k + \alpha} / \sum_l \sqrt{c_l + \alpha}$ to within 0.007. Combined with Proposition XII.1, the circuit closes the gap at the rate $\beta/(4n)$, and it does so with nothing a nervous system lacks: a race to threshold, a reward that potentiates the winner, and a homeostatic scaling of the total rate. Figure 13(c) shows it on the profile of GPT-OSS-120B. The selector never decides a result; it only chooses which class, or which agent, proposes next, and every proposal meets the checks, so the soundness of Theorems A to H is untouched and the gap law is a statement about cost alone.

E. What the gap law says, and what it does not

The law is exact for its model: independent proposals, a need that is a distribution over classes, and a proposer that keeps drawing until the class is right. Reaching the right class is necessary for reaching the right idea, not sufficient, and within a class a proposer may still fail; that part is priced by the checks, and the two costs multiply. The need is taken from the human papers, which is the only meaning the measurement of [55] gives to the right class; if the human profile is itself biased, square-root sampling of a better need is better still, and the law applies unchanged. What the law gives is a closed form for the cost of the gap that was measured, the proof that this cost is a chi-squared divergence from square-root sampling, a universal bound for its closure, and a circuit that reaches the optimum by learning from its checks at the rate $\beta/(4n)$. It does not make a language model think like a person. It makes a system that contains a language model search for ideas, at the level of the classes measured, with fewer proposals than either the model or a proposer with human taste.

XIII. BRANCHING TAKEOFF

A. The tree of improvements

A self-improving system need not move along a chain. Each kept improvement may spawn several candidates for the next, and the improvements form a tree (Fig. 14). The model is a Galton–Watson process with checks.

Definition XIII.1 (Checked branching process). Generation 0 is one kept correct system. Each kept correct improvement of generation $k-1$ spawns an independent copy of a random number $\xi \geq 0$ of candidates, with $\mathbb{E} \xi = \mu$ and $\mathbb{P}(\xi \geq 1) > 0$. Each candidate is correct with probability $\gamma \in (0, 1)$, independently. The check of generation k keeps a correct candidate with probability $1 - \eta$ and an incorrect one with probability $\delta_k \in (0, 1]$. Kept incorrect candidates are quarantined: they spawn nothing. Let Z_k be the number of kept correct improvements of generation k , F_k the number of kept incorrect ones, $F = \sum_{k \geq 1} F_k$, and $m = \mu\gamma(1 - \eta)$.

The number N of kept correct offspring of one kept correct improvement is obtained from ξ by keeping each candidate independently with probability $\gamma(1 - \eta)$, so (Z_k) is a Galton–Watson process with offspring law N and mean $\mathbb{E} N = m$. If ξ is Poisson with mean μ , then N is Poisson with mean m . The quarantine assumption is needed only for the formula for $\mathbb{E} F$; without it F is larger, so the statements that force $F = \infty$ hold a fortiori.

B. Takeoff

The process *takes off* if $Z_k > 0$ for every k . Let $f(z) = \mathbb{E} z^N$ and $q = \mathbb{P}(\text{takeoff})$.

Proposition XIII.2 (Takeoff threshold). *Assume $\mathbb{P}(N = 1) < 1$. Then $q > 0$ if and only if $m > 1$, and $1 - q$ is the least root of $f(z) = z$ in $[0, 1]$. If moreover $\mathbb{E} N^2 < \infty$ and $m > 1$, then*

$$\frac{2(m-1)}{f''(1)} \leq q \leq \frac{2(m-1)}{f''(1-q)}.$$

For Poisson offspring, $f''(1) = m^2$, $f''(1-q) = m^2(1-q)$ and, for $1 < m \leq 2$, $|q - 2(m-1)/m^2| \leq 4e^4(m-1)^2$.

Proof. The first two claims are the classical extinction theorem [4, 183]; Appendix A 8 recalls the proof. For the bounds, Taylor’s formula at 1 with $f(1) = 1$ and $f'(1) = m$ gives $f(1-q) = 1 - mq + \frac{1}{2} f''(\xi) q^2$ for some $\xi \in (1-q, 1)$. Since $f(1-q) = 1-q$, this is $(m-1)q = \frac{1}{2} f''(\xi) q^2$, so $q = 2(m-1)/f''(\xi)$; and f'' is increasing on $[0, 1]$, being a power series with nonnegative coefficients, so $f''(1-q) \leq f''(\xi) \leq f''(1)$. For Poisson offspring $f(z) = e^{m(z-1)}$, $f''(z) = m^2 e^{m(z-1)}$, and $e^{-mq} = 1-q$ at the fixed point. The two bounds then differ by $\frac{2(m-1)}{m^2} \cdot \frac{q}{1-q}$. Extinction has probability at least $\mathbb{P}(N = 0) = e^{-m} \geq e^{-2}$, so $1-q \geq e^{-2}$ and $q \leq 2e^2(m-1)$; the difference is at most $2(m-1)e^2 \cdot 2e^2(m-1)$. $\square$

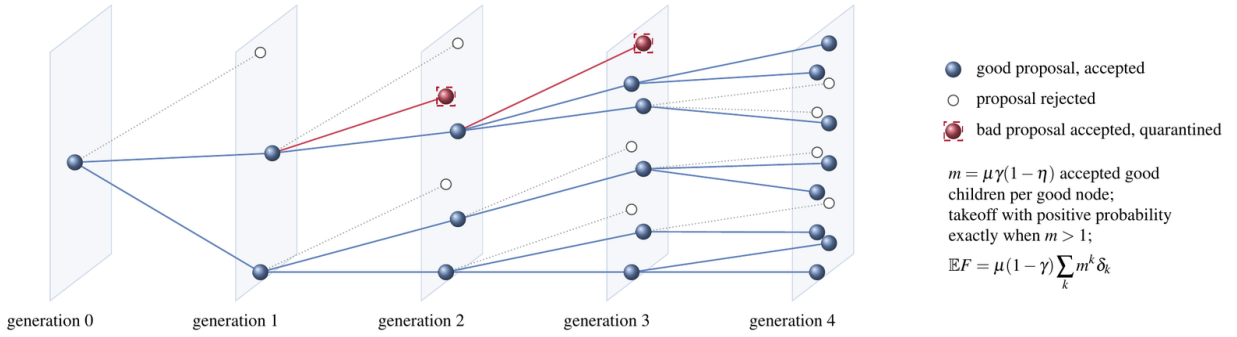


FIG. 14. A checked branching process, Definition XIII.1, over four generations drawn as sheets. Blue nodes are kept correct improvements, which spawn the next generation; grey nodes are rejected candidates; red nodes are incorrect candidates that the check kept, which are quarantined. The kept correct improvements grow like m^k in mean, and the expected number of kept errors is $\mu(1 - \gamma) \sum_k m^{k-1} \delta_k$ (Theorem E).

Near the threshold the probability of takeoff is linear in $m - 1$, with slope $2/f''(1)$, which tends to $2/\sigma^2$ at criticality, σ^2 the variance of the offspring law there; Table VI gives the numbers for Poisson offspring and Fig. 15(a) the curves. A takeoff is never certain: for $m \leq 2$ it fails with probability at least e^{-2} , and it is a matter of luck in its first generations.

C. Errors in a takeoff

Proof of Theorem E. Threshold. This is Proposition XIII.2.

Expected errors. Let $\mathcal{F}_k$ be the information of generations $0, \dots, k$. Given $\mathcal{F}_{k-1}$, the Z_{k-1} kept correct improvements spawn on average μZ_{k-1} candidates, each incorrect and kept with probability $(1 - \gamma)\delta_k$, so $\mathbb{E}[F_k | \mathcal{F}_{k-1}] = \mu(1 - \gamma)\delta_k Z_{k-1}$. With $\mathbb{E}Z_{k-1} = m^{k-1}$ and monotone convergence, $\mathbb{E}F = \mu(1 - \gamma) \sum_{k \geq 1} m^{k-1} \delta_k$. If the series converges, $F < \infty$ almost surely.

Divergence. Let $m > 1$, $\mathbb{E}\xi \log \xi < \infty$ and $\sum m^{k-1} \delta_k = \infty$. Since $N \leq \xi$, also $\mathbb{E}N \log N < \infty$, and the Kesten–Stigum theorem [4, 103] gives $Z_k/m^k \rightarrow W$ almost surely with $W > 0$ almost surely on takeoff. Let $B_k = \{F_k \geq 1\} \in \mathcal{F}_k$ and $u_k = (1 - \gamma)\delta_k$. Given $\mathcal{F}_{k-1}$ the parents act independently, and one parent with ξ candidates keeps no incorrect one with probability $\mathbb{E}(1 - u_k)^\xi \leq 1 - u_k \mathbb{P}(\xi \geq 1)$, because $1 - (1 - u)^j \geq u$ for $j \geq 1$. With $c_0 = \mathbb{P}(\xi \geq 1) > 0$,

$$\begin{aligned} \mathbb{P}(B_k | \mathcal{F}_{k-1}) &\geq 1 - (1 - c_0 u_k)^{Z_{k-1}} \geq 1 - e^{-c_0 u_k Z_{k-1}} \\ &\geq \frac{1}{2} \min(c_0 u_k Z_{k-1}, 1), \end{aligned}$$

TABLE VI. Probability q of takeoff for Poisson offspring of mean m , with the two bounds of Proposition XIII.2.

m	$2(m-1)/m^2$	q	$2(m-1)/(m^2(1-q))$
1.02	0.0385	0.0390	0.0400
1.05	0.0907	0.0937	0.1001
1.10	0.1653	0.1761	0.2006
1.20	0.2778	0.3137	0.4048
1.50	0.4444	0.5828	1.0653
2.00	0.5000	0.7968	2.4608

the last step by $1 - e^{-x} \geq \frac{1}{2} \min(x, 1)$. On takeoff, $Z_{k-1} \geq \frac{1}{2} W m^{k-1}$ for all large k , so up to finitely many terms

$$\sum_k \mathbb{P}(B_k | \mathcal{F}_{k-1}) \geq \frac{1}{2} \sum_k \min\left(\frac{c_0(1 - \gamma)W}{2} m^{k-1} \delta_k, 1\right).$$

The right side diverges, since $\sum m^{k-1} \delta_k$ diverges and a series $\sum \min(cx_k, 1)$ with $c > 0$, $x_k \geq 0$ converges only if $\sum x_k$ does. Lévy’s conditional Borel–Cantelli lemma [64] states that $\{B_k \text{ infinitely often}\}$ and $\{\sum_k \mathbb{P}(B_k | \mathcal{F}_{k-1}) = \infty\}$ agree up to a null set (Appendix A5). So $F = \infty$ almost surely on takeoff.

Exponential checks. With $\delta_k = \varepsilon^{r_k}$ and $r_k = \lceil k \log(2m)/L \rceil$, $r_k L \geq k \log(2m)$, so $m^k \varepsilon^{r_k} \leq 2^{-k}$ and $\sum m^{k-1} \delta_k \leq \frac{1}{m} \sum 2^{-k} < \infty$. Conversely, if $\sum m^{k-1} \varepsilon^{r_k} < \infty$, its terms tend to 0, so $r_k L - (k - 1) \log m \rightarrow \infty$, hence $r_k L \geq k \log m$ for all large k . A constant $\delta_k = \delta > 0$ gives $\sum m^{k-1} \delta = \infty$ for $m \geq 1$. $\square$

Corollary XIII.3 (The entropy tax of a takeoff). *In a sound takeoff with exponential checks and $m > 1$, every kept correct improvement of generation k has been checked by at least $k \log m / L = \log(\mathbb{E}Z_k) / L$ runs, for all large k .*

The tax is the logarithm of the size of the population, divided by L , exactly as for n candidates of equal cost in Corollary V.4: a growing population of improvements can stay sound only if each new improvement is checked harder than the last by a fixed number of runs per generation, and a fixed false acceptance rate, however small, lets errors multiply with the population (Fig. 15(b)).

XIV. TOWERS OF VERIFIED LAYERS

A BLLM can act on another BLLM. The proposer of Section XV that builds a graph is one engine; an engine that chooses among strategies of that proposer and keeps the ones that build faster is a second, standing on the first; an engine that runs several copies of the second and keeps the better half after each round is a third. We call a stack of n such engines a *tower of depth n* , write it B^n , and ask what depth costs. The

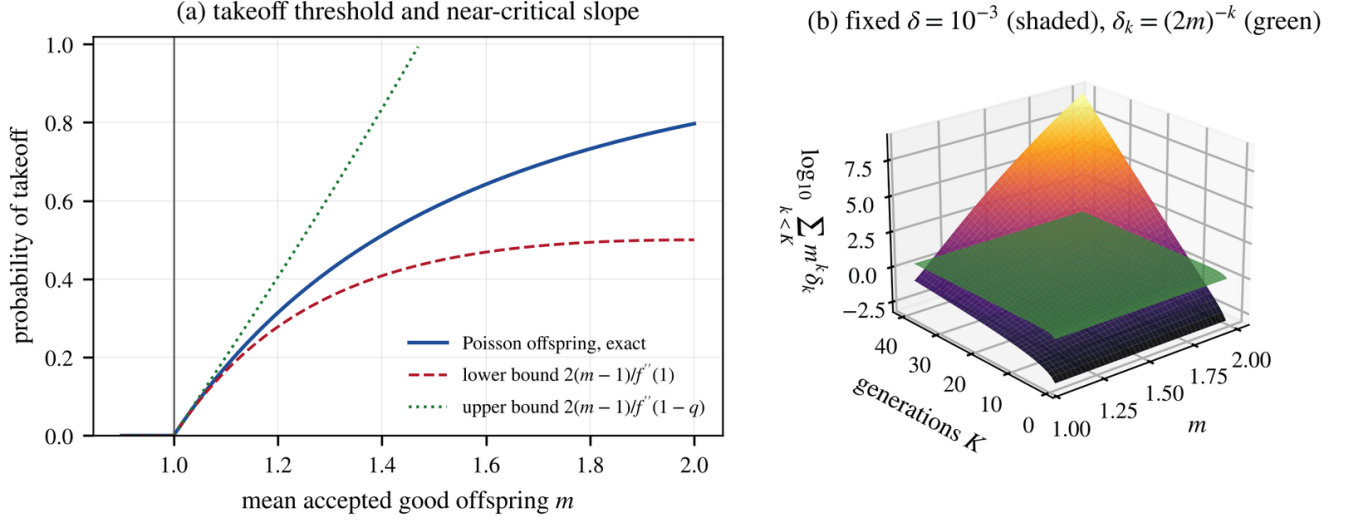


FIG. 15. Theorem E. (a) The probability of takeoff for Poisson offspring as a function of the mean m of kept correct offspring, with the two bounds of Proposition XIII.2; the threshold is $m = 1$ and the slope there is 2. (b) The expected number of kept errors in the first K generations, $\log_{10} \sum_{k < K} m^k \delta_k$, for a fixed rate $\delta = 10^{-3}$ (shaded surface), which grows without bound, and for the amplified rate $\delta_k = (2m)^{-k}$ (green), which stays below $\log_{10} 2$.

answer is a corollary of the entropy law, and it bounds what any implementation of depth can gain.

A. The model

Let a tower have layers $0, \dots, n-1$. Layer k passes an object up only after it has been certified, and a false acceptance at layer k contaminates every layer above it, since the layers above build on what it passed. So the layers share one budget: the false acceptance probabilities β_k of the n certifications satisfy $\sum_{k < n} \beta_k \leq \theta$, and the certification cost of the tower is $\sum_{k < n} \log(1/\beta_k)$ nats of soundness. A layer certified by r_k independent runs of a check of soundness ε has $\beta_k = \varepsilon^{r_k}$ and costs r_k runs. The search cost S_n of the tower, the work its layers do before anything is certified, is left arbitrary except for a lower bound B that every stage pays.

The same model covers a *mesh*, in which the layers read and write one shared library and the connections between them are strengthened when they are used. Every object enters the shared library only after the independent check of Section XV G, so a false acceptance still contaminates every user of the library, and the count of certified passes, not the wiring, is what enters the bounds below.

B. What depth costs

Theorem XIV.1 (Tower law). *Let $0 < \theta \leq 1$ and $n \geq 1$.*

- (i) *If $\beta_k > 0$ and $\sum_{k < n} \beta_k \leq \theta$, then $\sum_{k < n} \log(1/\beta_k) \geq n \log(n/\theta)$; with exponential checks of soundness ε , a tower needs at least $n \log(n/\theta) / \log(1/\varepsilon)$ runs in all.*

- (ii) *Adding a layer raises the bound of (i) by at least $\log((n+1)/\theta)$, and these increments tend to infinity.*

- (iii) *If every stage costs at least B to search and the certification of a depth- n tower costs at least $cn \log(n/\theta)$ with $c > 0$, then for every T_1 there is a depth N such that every tower of depth $n \geq N$ costs more than T_1 ; in particular, past a finite depth every tower costs more than the tower of depth one.*

- (iv) *Along an infinite tower whose budgets $\beta_k > 0$ are summable, the certification cost $\log(1/\beta_k)$ of a single stage tends to infinity.*

Proof. (i) is the lower bound of Theorem A for n candidates of unit cost: with $a_k = 1$ the entropy of the profile is $\log n$, and the bound reads $\sum_k r_k \geq (n/L)(\log n + \log \frac{1}{\theta})$; in nats of soundness this is $\sum_k \log(1/\beta_k) \geq n \log(n/\theta)$. (ii) With $f(n) = n \log(n/\theta)$, $f(n+1) - f(n) = \log((n+1)/\theta) + n \log((n+1)/n) \geq \log((n+1)/\theta)$, and $\log((n+1)/\theta) \rightarrow \infty$. (iii) For $n \geq 3$ and $\theta \leq 1$ we have $n/\theta \geq 3 > e$, so $\log(n/\theta) \geq 1$ and the cost is at least $B + cn$, which exceeds T_1 as soon as $n > (T_1 - B)/c$. (iv) A summable sequence of positive terms tends to 0, so $1/\beta_k \rightarrow \infty$. $\square$

The six statements are proved in Lean in the file *Tower*, listed in Table XX.

The theorem is a ceiling, not a recipe. Depth is worth paying for only while it lowers the search cost S_n by more than the increment of (ii), and (iii) says that this stops at a finite depth whatever the search does. A tower that seems to gain without limit from depth is either uncertified somewhere or spending its soundness budget faster than it says. Section XVIII B measures S_n for $n = 1, 2, 3$, ladder and mesh, at equal budgets of exact checks.

Remark XIV.2 (Postnikov towers). A Postnikov tower of a connected CW complex X is a sequence of fibrations $\cdots \rightarrow X_n \rightarrow X_{n-1} \rightarrow \cdots \rightarrow X_1$ with maps $X \rightarrow X_n$ that are isomorphisms on π_k for $k \leq n$, where $\pi_k(X_n) = 0$ for $k > n$ and each fibre of $X_n \rightarrow X_{n-1}$ is an Eilenberg–MacLane space $K(\pi_n(X), n)$ [84, Section 4.3]. We use the picture in one precise sense. Stage n of the tower is the engine cut at depth n , and passing from stage $n-1$ to stage n adds exactly one layer, whose cost is the increment of Theorem XIV.1(ii); that increment is the only part of the picture the paper uses, and it is proved. That the passage is a fibration, that it has a fibre with one nonzero invariant, or that the stages are glued by k -invariants has no counterpart in the engine, and we do not claim one.

XV. ANAXIA: THE ENGINE

The theorems of the preceding sections price the checking of candidates and bound the number of proposals any proposer needs. They say nothing about where candidates come from. This section describes Anaxia, the engine we built to supply them, in the order in which a candidate meets its parts. In the crossing network, known constructions are typed nodes, new candidates are crosses of old ones, every cross is checked and then refuted if possible, and every outcome is written to a memory that steers the next cross (Fig. 16). Construction mode writes down a new object, a graph, together with the argument for what it forces, and has it checked again by code that shares nothing with the search. The tower of Section XIV stacks construction mode into layers. A registry records what each part takes from earlier systems, and the whole is released as one package.

A. Constructions as typed nodes

A *construction* c is a record of three items: $\text{gives}(c)$, the type of the object it produces; $\text{needs}(c)$, the list of types it consumes; and $\text{source}(c)$, the published result or computation it comes from. The single-link law of $\text{SU}(2)$ gives a link law and needs nothing; Dobrushin’s uniqueness criterion [62] gives a range of couplings and needs a dependency graph and an influence bound; Shioda’s inductive structure [157] gives a splitting of a Fermat character and needs a character table. If $\text{gives}(b) \in \text{needs}(a)$, the *cross* $a \leftarrow b$ is the construction with the first such input of a filled by b ; its parts are the union of the parts of a and b . A composite with no unfilled input is *closed*, and only closed composites are checked. Typing removes ill-formed candidates before any check: on the Yang–Mills library, crossing without regard to types wastes 45.4 of 48 crosses per run on ill-typed pairs (Table XII).

The library of a benchmark is the set of its constructions. Five libraries are used: Yang–Mills (38 nodes), Hodge (28), the standard conjecture of Lefschetz type (17), Chowla’s conjecture (20) and Erdős problem 1183 (13). Of these 116 nodes, 54 are literature constructions written out with their key lemma and the barrier at which they stop, for instance the Kotecký–Preiss criterion, the

Brydges–Yau and Gawędzki–Kupiainen block-spin flows, Holley–Stroock and Stroock–Zegarliński for Yang–Mills; van Geemen, Ribet, Kuga–Satake, Varesco and Charles–Markman for Hodge; Friedlander–Iwaniec, Heath-Brown, Green–Tao and Matomäki–Radziwiłł–Tao for Chowla.⁶

B. Checks and the refuter

The check of a closed composite is a program attached to its output type. It returns *accepted* with a measurement, *rejected* with the measurement that failed, or *stopped* when no checkable form of the composite exists. On Yang–Mills the check of a Dobrushin composite counts the active partners of a link in the dependency graph the composite names and computes the range $\beta_W < 1/(\text{partners} \cdot \frac{1}{4})$ from the influence bound; on the Fermat fourfolds it enumerates the zero-sum splittings of the characters in an orbit; on Erdős problem 1183 it computes the largest monochromatic sublattice by the search of Lemma XVII.1. An accepted composite then meets the *refuter*, which breaks it if its range is dominated by one already known, if it reproduces an accepted statement without gain, or if it rests on an unrefereed claim. Only composites that pass both count as survivors. A survivor whose statement is new is formalised in Lean before it is reported; none of the five pass lines has such a survivor.

C. Boosters

Every ordered pair of library heads carries a synapse with weight w , initially 1. When a cross fires and its offspring survives, its synapse rises by η , and the synapses that built its parents rise by $\eta\gamma^k$ at distance $k \leq 4$; a rejected or refuted offspring lowers its synapse by $\eta/2$, to no less than 0.05. After each generation the weights leaving each node are rescaled to mean one, and a share ε of the crosses is drawn uniformly. We use $\eta = 1$, $\gamma = \frac{1}{2}$ and $\varepsilon = 0.2$. This is a Hebbian rule with path credit, and Section XVII D measures what it adds.

D. Memory

The memory is one SQLite store shared by every proposer and every round. It holds the nodes, one event per checked composite (its parts, outcome, measurement, reason, synapse weight, round and proposer), an index from parts to events, the synapse weights, the blocks (for each problem and step,

⁶ The list is the file `knowledge.json` of the package. Each entry carries the problem it addresses, its typed inputs and output, the published source, the lemma that does the work, and a verdict: *known* when the construction proves a published statement, *stops* when it is known to fail at the benchmark’s blocking step, with the reason. A literature node is never counted as progress: a composite that reaches a known statement is accepted as known and rewarded at a quarter of the rate of a new one, and a composite through a node marked *stops* is stopped with that node’s reason.

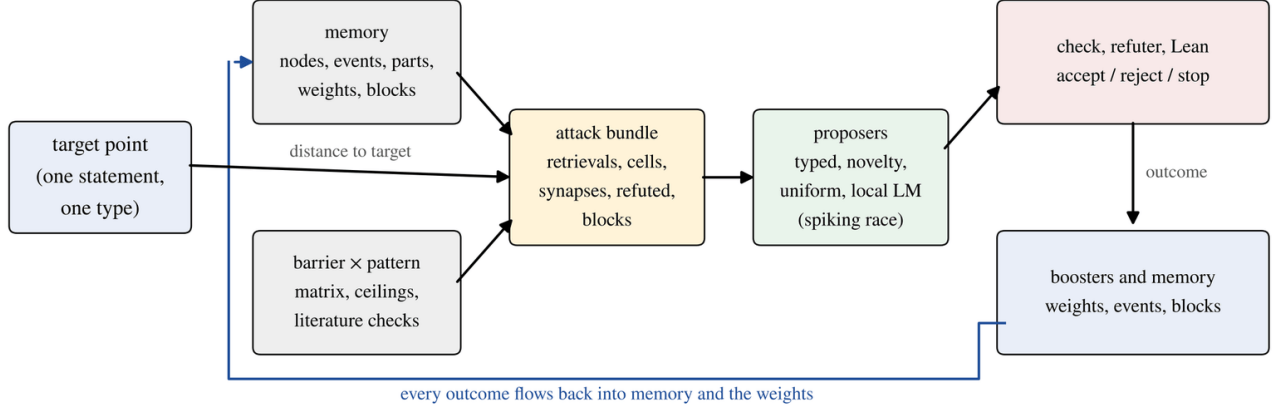


FIG. 16. One round of the attack loop. The target point of a benchmark fixes a type; every composite gets a distance to that type. The attack bundle collects, for each candidate cross, the most similar past attempts in memory, the cells of the barrier and pattern matrix that touch its parts, the weights of its synapses, the refuted offspring of its parts and the blocks recorded on its problem. The proposers draw from the weighted candidates; the check, the refuter and, for a new statement, Lean decide the outcome, which updates the boosters and the memory before the next round.

the assumption that stopped it and the source of the stopping composite), the literature checks, and a log of every retrieval. Two uses follow.

- (i) *Decided composites are skipped.* A composite whose parts already have an event is not checked again; this is what removes the repeated checks of Fig. 22(b).
- (ii) *Retrieval steers the draw.* For a candidate with part set Q , the $k = 5$ past events with the largest score $J(Q, S)$ are retrieved, where S is the part set of the event, J the Jaccard similarity and w the synapse weight at the time. The candidate's weight is multiplied by

$$\pi_{\text{mem}}(Q) = \min\left(3, \max\left(\frac{1}{4}, 1 + \sum_{\text{survivors}} J - \frac{1}{2} \sum_{\text{others}} J\right)\right), \quad (14)$$

which raises it near past survivors and lowers it near failures.

The store of a five-round study over all five libraries holds 42 events, 11 blocks and 26 weights in 156 kilobytes, so the memory is small and the check, not the bookkeeping, is the cost.

E. Target points and the attack channel

A *target point* is one statement whose proof would cross the wall of a benchmark, together with its type. For Erdős problem 1183 it is a family of colourings of $2^{[n]}$ whose monochromatic sublattices have at most $n/2 + C$ members for every n ; for Yang–Mills, an effective action at one coupling $\beta_w > \frac{1}{3}$ whose blocked measure satisfies a Dobrushin-type condition uniformly in the spacing; for Hodge, the algebraicity of the class of the character $(1, 16, 26, 31, 36)$ of level 55 on the

Fermat fourfold of degree 110; for Chowla, local U^2 uniformity of the Liouville function on almost all intervals of length $\exp((\log X)^{2/5-\delta})$, one step below the scale of [137]; for the standard conjecture, an algebraic inverse of L on H^2 of one threefold of general type with $h^{2,0} > 0$. A backward search over the library gives every type a distance: 0 for the target type, and $1 + d$ for each input of a construction whose output is at distance d . A candidate's weight is multiplied by e^{-d} , where d is the distance of its output, so the forward search by crossing and the backward search from the target meet at the composites of distance 0 and 1.

The *attack channel* is the single pipeline that hands everything the engine holds to the proposers. For each candidate it builds a bundle: the retrievals, the cells of the barrier and pattern matrix that touch its parts (Section XVII H), the weights of its synapses, the refuted offspring among the retrievals, and the blocks of its problem. The bundle multiplies the weight by

$$\pi_{\text{ch}} = \min\left(5, \max\left(0.05, 2^{-n_{\text{fail}}} \left(\frac{3}{2}\right)^{n_{\text{part}}} \left(1 + \sum_{\text{survivors}} J\right)\right)\right), \quad (15)$$

where n_{fail} and n_{part} count the cells of the bundle whose recorded outcome is a failure or a partial result. Every bundle that flows is written to the memory, so that a run with the channel can be compared with one without.

F. Proposers and agents

Four proposers share the weighted candidates: *typed* draws in proportion to the weight; *novelty* takes the candidate least similar to anything in memory; *uniform* draws uniformly; and *local model* shows the eight heaviest candidates and the retrieved notes to a local language model with 5×10^8 parameters [142] and takes the one it names. Which proposer moves

is decided by the spiking selector of Proposition XII.2, rewarded on survivors, so that the committee samples the square root of each proposer’s record of success, the optimum of Theorem I.

An *agent* holds one target point, its own library, its own synapse weights and its slice of the shared memory. Each step it attacks its library through the channel. When a step adds no survivor it runs a sub-attack whose target is the nearest open input type of its target point, a lemma at distance one. Agents trade through the memory: a result passes from one agent to another only where one benchmark is an input of another, which among the five happens once, since the standard conjecture for X is a case of the Hodge conjecture on $X \times X$. Both are package code, not language-model agents: an agent is an object with a library, weights and a memory slice, and its step is a call of the attack.

G. Construction mode

The crossing network only combines results that someone has already proved. Construction mode builds an object that no one has written down, on problems where the object is finite: planar graphs whose proper 3-colourings are constrained at a few *contact* vertices on one face. Such a graph is a *gadget*; it is a not-all-equal forcer if no proper 3-colouring gives its three contacts one colour, and its *profile* is the list of distances between its contacts. Gadgets of this kind are the building blocks of the refutation of Steinberg’s conjecture by Cohen-Addad, Hebdige, Král’, Li and Salgado, a planar graph with no cycles of length 4 or 5 that is not 3-colourable [43], and of every attempt on Erdős’s question whether a planar graph with no cycles of length 4, 5 or 6 is 3-colourable.

The builder grows a gadget together with its forcing argument. It assumes that the contacts share a colour γ and applies four moves: *attach* a new vertex next to a γ vertex, which is then not γ ; *link* two vertices that are not γ , whose colours then differ; *derive* a new vertex next to two vertices whose colours are known to differ, which is then γ ; and *close* with an edge or a triangle that leaves no colouring. A library gadget may also be inserted as a lemma, with its contacts sent to vertices of the graph. Between moves the builder propagates the forced colours to a fixed point and adds structure only when nothing more follows. Every move passes a cascade of tests before it is kept: no cycle of a forbidden length through a new edge, contact distances at least those of the profile, and planarity with the contacts on one face; only then does an exact satisfiability check decide whether the contacts can still share a colour. When they cannot, the graph is minimised and offered to the gate.

The *gate* is code that shares nothing with the builder: planarity by the left–right algorithm of `networkx`, every cycle of forbidden length by depth-first search, the contact distances by breadth-first search, and the absence of a colouring with equal contacts by plain backtracking with no satisfiability solver. A gadget counts only when the gate accepts it, and the whole counterexample assembled from a gadget is checked again by three satisfiability solvers, CaDiCaL, Glucose and MiniSat. A

second search writes the forcing condition as an exists-forall problem: a satisfiability instance over the edge variables, with planarity built in by a 2-page book embedding and every 3-colouring that defeats a candidate turned into a clause that every forcer satisfies. The cost of every run is counted in exact satisfiability checks, so that different searches can be compared at the same budget.

H. The tower of builders

The tower B'' of Section XIV is built over construction mode. Layer 1 is the builder with one fixed strategy: the order in which it tries the four moves, how much noise it adds to that order, and how often it tries a lemma. Layer 2 acts on layer 1: it keeps a population of strategies, runs each for a slice of checks, keeps the better half by gadgets found per check and refills the population with mutations of the survivors, the evolve-and-evaluate loop of FunSearch and AlphaEvolve [123, 143] with a fixed space of strategies in place of program text. It also sets a curriculum, as AlphaProof trains on variants of a target before the target [90]: a weaker contact profile is attacked first, and the gadget it yields joins the library as a lemma for the harder one. Layer $k \geq 3$ acts on layer $k - 1$ by successive halving [98]: it runs several copies of layer $k - 1$ with different settings and keeps the better half after each round.

Two wirings are compared. In the *ladder* each copy of a layer has its own library and sees only what the layer directly below returns. In the *mesh* every layer reads and writes one library, so a gadget found at the bottom of one branch is a lemma everywhere at once, and the move kinds carry Hebbian synapses [87]: the moves that built a gadget the gate accepted are strengthened by $\eta = 1$ shared among them, every synapse decays by a factor 0.95 at each success, and a strategy’s priority for a move is lowered by 1.5 times its synapse, so that moves that have worked are tried earlier by every layer. Whatever path a gadget took, it counts only after the gate, and the budget of a run is the number of exact checks it spends.

I. What the engine takes from other systems

Every system, corpus and method that Anaxia draws on is a record of one registry: its reference, its status (refereed, preprint, software or unrefereed claim), what was taken, the module that implements it, what Anaxia adds and what was left out. The registry has 35 entries in eight groups, Google DeepMind, methods, OpenAI, The Omega Institute, memory, agents, corpora, and provers and models; Appendix D lists them. Three corpora are indexed as literature and searched before a target is attacked, to see whether it is already settled: the 5,035 items of the public repositories of Google DeepMind, among them 1,471 open and 2,437 solved statements of Formal Conjectures; the 782 problem dossiers of OpenRefute, each a published conjecture with a Lean claim and a kernel-checked resolution [127]; and the 722 manuscripts of OpenAI’s collection [125], every one marked unrefereed.

Nothing in an index is used as a lemma by a check, and nothing fetched is executed. No model weights are bundled with the engine.

J. The package

Anaxia is released as the Python package eBLLM, with the command `anaxia`, a Lean 4 library and the container image `ghcr.io/creelie/blm-engine`, which is rebuilt on every change and every release and tells a running copy when a newer one is published; release 3.0.0 is archived under [doi:10.5281/zenodo.23464390](https://doi.org/10.5281/zenodo.23464390)[24]. Its first version implemented the verifier-gated engine of [23]. It contains no source code of any language model; hosted models are called through their public interfaces with the user’s key and local models through `llama.cpp`. Its parts are the schedules of Sections V to XIV; the proposers; the libraries and the literature constructions; the checks, which are the Lean kernel, exact evaluation, randomized polynomial identity tests [151, 191], the searches of Lemma XVII.1 and of its grid version in C, the gate of construction mode and arbitrary commands with checked output; the memory, the boosters, the channel and the agents of this section, with every kept result also written as a Markdown note linked to the results it uses, so that the memory can be opened as an Obsidian vault and searched with the neighbours of each hit; construction mode and the tower; the registry and the three indexes; the five benchmarks with their pass lines; and a Lean development whose finite parts back the theorems of the paper. One command runs the tests, the memory study and the agent loop, and the hardware each part needs is measured in Appendix C.

XVI. FIVE OPEN PROBLEMS AS A BENCHMARK

A proposer is tested by what it finds, and on solved problems a network that recombines known constructions can only rediscover. We therefore test on five open problems, each with a pass line fixed in advance (Table VII). The Yang–Mills mass gap and the Hodge conjecture are two of the Millennium problems (Table VIII); the standard conjecture of Lefschetz type, Chowla’s conjecture and Erdős problem 1183 are smaller and have partial results that a computation can extend. A benchmark is passed when its whole open statement is proved, the proof survives the refuter, and its finite part is checked in Lean. Between passes each benchmark reports a partial score whose definition was fixed before the runs, and the step that blocks.

A. Splitting a conjecture

The partial scores of the standard conjecture and of Chowla’s conjecture come from splitting each into the cases along which it can be checked: degree, dimension and class for the standard conjecture, order and averaging for Chowla. A piece counts as closed only when a published proof settles it, and the source of every closed piece is recorded (Ta-

ble IX). The split matters for choosing targets. An attack on the standard conjecture for Fermat varieties is wasted, since the conjecture passes to products and to smooth ample divisors [99] and so holds for every hypersurface; the open pieces of Chowla’s conjecture are open because the method that settles the known ones does not reach them.

In degree i the standard conjecture asks that the inverse of $L^{d-i}: H^i \rightarrow H^{2d-i}$ be induced by an algebraic cycle; on a threefold the only nontrivial degree is $i = 2$, where the inverse is automatic on the span of the divisors and the missing object is one cycle on $X \times X$ that inverts L on the transcendental lattice. The known operations, products, divisors, fibrations and motives of abelian type, produce no threefold of general type with $h^{2,0} > 0$ whose transcendental lattice is not of abelian type. Since the conjecture for X is the algebraicity of one Hodge class on $X \times X$, its open residue is a case of the Hodge conjecture. For Chowla’s conjecture the entropy decrement argument of [168] needs logarithmic weights, and the even orders $k \geq 4$ need uniformity of the Liouville function on intervals of every length $H \rightarrow \infty$, while the best uniformity known holds for linear phases on intervals of length $\exp((\log X)^{2/5+\varepsilon})$ [137] and for nilsequences on intervals of length X^θ [119]. The two-point theorem with ordinary averages [126] extends the expansion method of [92] from logarithmic to ordinary averages by bounding the operator of a dilation graph through closed walks; the method compares two points, and no k -point graph with such a bound is known. Neither open piece reduces to a finite residue with explicit constants, and no monotone quantity is known that classifies the residue, so neither the finite reduction nor the residue trivialization of the package applies.

B. Why the walls hold

Each blocking step resists for a reason that can be stated before any run, and the reason predicts what a recombining proposer can and cannot do.

Yang–Mills. Every rigorous method for lattice gauge theory works in a regime with a small parameter: cluster and polymer expansions, Dobrushin’s criterion and its gauge-fixed forms at strong coupling, Bataban’s renormalisation group at weak coupling. Hypothesis R needs control at order-one coupling, uniformly in the spacing, where neither parameter is small. Single-link Dobrushin arguments have a ceiling at $\beta_W = \frac{1}{3}$ for every forest gauge fixing (Proposition XVII.5), and block arguments need the boundary sensitivity of a box law, an integral over $\mathrm{SU}(2)^{3n^4}$ that no method bounds at the size required.

Hodge. The known sources of algebraic cycles are divisors, products, correspondences from abelian varieties, the Kuga–Satake construction and the semiregular sheaves of Markman. The open families are exactly those that none of them reaches: very general Weil sixfolds and eightfolds, squares of K3 surfaces whose real multiplication is a similitude and not an isometry, and the Fermat fourfold of degree 110, whose leftover character has no zero-sum splitting in its orbit and a lattice point in the Fine interior of its Newton polytope, so that no block family of curves covers it.

TABLE VII. The five benchmarks. The partial score is defined before the runs; “blocking step” is the first step that no construction of the library and no published result supplies. None is passed. The range $\beta_W < \frac{1}{3}$ is proved in Theorem XVII.4 and has not been refereed.

Benchmark	Pass line	Partial score	Blocking step
Yang–Mills	a lattice gap for four-dimensional SU(2) uniform in the spacing (Hypothesis R)	3 of 8 leaves; $\beta_W < \frac{1}{3}$, which is 15.2% of the range up to the crossover at 2.2 (refereed: $\frac{2}{9}$, 10.1%)	renormalisation steps at order-one coupling between Balaban’s fine-scale control and a convergent expansion; single-link arguments stop at $\frac{1}{3}$ after every forest gauge fixing (Proposition XVII.5)
Hodge	(F2) and (F3’), equivalently the Hodge conjecture [17]	2 of 10 leaves; families covered by dimension: Weil sixfolds 5/9, Weil eightfolds 0/16, squares of K3 surfaces with real multiplication 4/8	no source of algebraic cycles for Hodge classes off the known families; the Fermat orbit of degree 110 is indecomposable
Standard conjecture B	every smooth projective variety	11 of 14 pieces (Table IX)	an algebraic inverse of L on transcendental H^2 ; it reduces to the Hodge conjecture on $X \times X$
Chowla	every k , ordinary averages	4 of 8 pieces, one more claimed (Table IX)	the parity barrier; local Gowers uniformity on intervals of every length $H \rightarrow \infty$
Erdős 1183	$f(n)$ for every n , or an answer to Question 9.3: is $f(n) = O(n)$?	$f(n)$ exact for $n \leq 20$ (Theorem XVII.2)	a family of colourings for every n

TABLE VIII. The seven Millennium problems [48]: what is proved, the first step that is open, and, for the two problems with a benchmark tree in the package, the number of its leaves that are proved after the first headline round (Section XVII C). The leaves of a tree are the pieces into which the package splits a problem; a proved leaf may be a published theorem (fine-scale control at weak coupling, (F2) in dimension at most five) or a step accepted by a run (the strong-coupling gap, its extension to $\beta_W < \frac{1}{3}$, the reduction for Fermat fourfolds). The range $\beta_W < \frac{1}{3}$ is proved in Theorem XVII.4 and has not been refereed. The Navier–Stokes development is formal but not refereed.

Problem	Status	What is proved	First open step	Leaves
P versus NP	open	relativizing, natural and algebrizing arguments cannot separate P from NP [12, 22, 147]	a superpolynomial circuit lower bound for an explicit function in NP	–
Riemann hypothesis	open	more than 5/12 of the nontrivial zeros lie on the critical line [139]; the hypothesis holds up to height $3 \cdot 10^{12}$ [141]	every nontrivial zero	–
Birch and Swinnerton-Dyer	open	for analytic rank at most one the rank equals the analytic rank and the Tate–Shafarevich group is finite [83, 101]; the average rank is bounded and a positive proportion of curves have rank zero [36, 37]	analytic rank at least two	–
Navier–Stokes	formal	breakdown with forcing, alternatives (C) and (D), for every viscosity [124]	the correspondence of paper and formal proof [20]	–
Yang–Mills mass gap	open	lattice gap for four-dimensional SU(2) when $\beta_W < \frac{2}{9}$ [18], and when $\beta_W < \frac{1}{3}$ (Theorem XVII.4)	order-one coupling, uniformly in the spacing	3/8
Hodge conjecture	open	equivalence with (F2) and (F3’) [17]	(F2) for CM fields of degree at least four off CM points; (F3’) outside the class $\mathcal{A}$	2/10
Poincaré conjecture	proved	Ricci flow with surgery [133–135], in Lean 4 [190]	nothing	–

Standard conjecture B. On a threefold the conjecture asks for one algebraic cycle on $X \times X$ inverting L on the transcendental part of H^2 , which is a case of the Hodge conjecture. Products, ample divisors, fibrations and motives of abelian type do not produce threefolds of general type with $h^{2,0} > 0$ whose transcendental lattice is not of abelian type, so every known route stops there.

Chowla. The parity barrier of sieve theory prevents any sieve from distinguishing integers with an even number of prime factors from those with an odd number. The entropy decrement method of [168] needs logarithmic averages and, for even $k \geq 4$, local uniformity of the Liouville function on intervals of every length $H \rightarrow \infty$, which is known only at the scales of [119, 137].

Erdős 1183. A random colouring has no monochromatic sublattice larger than $O(n \log n)$, and the first-moment count cannot do better, since the expected number of monochro-

matic chains of length N is at least $(N+1)^n 2^{1-N}$. A bound $O(n)$ therefore needs a structured family of colourings for every n , and no such family is known. Size colourings reduce the question to a finite Ramsey problem on rank sets (Lemma XVII.1), which a computer can settle for each n and not for all.

In each case the step needs an input that is not in any library: a convergent method at order-one coupling, a new source of cycles, a new uniformity estimate, a colouring family. A network that recombines known constructions can reach the edge of what they prove and certify where they stop; it cannot supply the missing input unless a cross produces it, and the experiments of Section XVII measure how often crosses do.

TABLE IX. The standard conjecture of Lefschetz type (B) and Chowla’s conjecture split into pieces. A piece is closed when a published proof settles it; the 2026 two-point theorem with ordinary averages has a Lean proof that was audited but not rebuilt here and is counted as claimed. The share of closed pieces measures the split, not the distance of the conjecture from a proof.

Standard conjecture B: 14 pieces, 11 closed (78.6%)		Chowla: 8 pieces, 4 closed (50.0%), 1 claimed	
curves, surfaces; divisor classes	[99, 106]	$k = 1$	prime number theorem
abelian varieties	[106]	$k = 2$, logarithmic averages	[168]
products, smooth ample divisors	[99]	$k = 2$, logarithmic averages, quantitative savings	[92, 136]
threefolds with $h^{2,0} = 0$	Lefschetz (1, 1)	odd k , logarithmic averages	[170, 171]
threefolds with $\kappa < 3$	[166, 167]	$k = 2$, ordinary averages	claimed [126]
uniruled threefolds, unirational fourfolds, Hilbert schemes of surfaces, moduli of bundles on curves type K3 ^[n]	[7]	even $k \geq 4$, logarithmic averages	open
generalised Kummer type, low degrees	[49]	$k \geq 3$, ordinary averages	open
generalised Kummer type, dimension $4d$	[73]	every k , ordinary averages	open
O’Grady’s ten-dimensional examples of Laza–Saccà–Voisin type	conditional [74]		
O’Grady’s six-dimensional examples	[2]		
threefolds of general type with $h^{2,0} > 0$	[7, 72]		
every smooth projective variety	open		

XVII. EXPERIMENTS

Every experiment of this section fixes its measured quantity before the runs, compares each arm with controls that spend the same kind of budget, and counts a step only after the check that the run itself performs. The unit of cost is one check of one closed composite. The experiments are, in order: a router of strategies on the two Millennium benchmarks; the crossing network on Erdős problem 1183, where the only advance came from a change of representation; the crossing network on Yang–Mills and Hodge, where it reaches the range $\beta_W < \frac{1}{3}$; the booster layer; the memory, the channel and the agents; and the barrier and pattern matrix.

A. A router of strategies on Hodge and Yang–Mills

In its first form the package represented each of the two open problems as a tree whose root is the statement and whose leaves are the pieces it splits into: a faithful Lean statement, the lattice gap at strong coupling, fine-scale control at weak coupling, the crossover window at order-one coupling with two attempts on it, and the continuum limit for Yang–Mills; a faithful Lean statement, (F2) in dimension at most five, for CM abelian varieties, for split Weil eightfolds and through a semiregular object on E_0^6 , and (F3’) for the square of a K3 surface with real multiplication and for Fermat fourfolds for Hodge. A run picks a leaf and one of eight moves (reduction, special case and lift, certificate, construction, counterexample, flow with surgery, descent, physics first), and a move is accepted only when a computation of the package checks it. These are the trees of six and eight leaves on which the router was run; Section XVII C adds the leaves the crossing network opened. Five policies choose the move: uniformly; by the prior of the Lean-formalised manuscripts of [125]; by a router that averages that prior with the share of the settled Erdős problems solved by each move and is reinforced by ac-

cepted steps; by the profile of a local language model, the softmax of the next-token logits of a model with 1.5×10^9 parameters over the letters of the eight moves [142]; and by the mixture of the last two with weight $\frac{1}{2}$.

Over 200 seeds of 12 rounds every policy finds the same two accepted steps, the reduction of the Hodge conjecture for a Fermat fourfold of degree m to the degrees with leftover characters, by Shioda’s criterion for zero-sum pairs and triples, and the strong-coupling gap $m \geq -\log(9\beta_W/2)$ for $\beta_W < \frac{2}{9}$. The mean number of attempts before both are found is 36.3 for the uniform policy, 38.3 for the corpus prior, 35.4 for the router, 26.3 for the profile of the model and 33.5 for the mixture; of 65 to 67 attempts per run, between 50.9 and 52.9 go to moves that have no checkable formulation at their leaf. The model’s profile is the best of the five, by a factor of 1.38 over the uniform policy, and no policy moves any open leaf. Both accepted steps are results of [17, 18]; the benchmark measures how fast a policy finds a step that a computation can check, and on these trees the only such steps are known ones. Moving a leaf needed a different proposer, one that combines known constructions instead of choosing among moves, and that is the crossing network of the next subsection.

B. A chain reaction of constructions

The library of the package records known constructions with their provenance. The crossing network makes each of them a typed node and breeds new constructions from pairs of old ones, keeping an offspring only if it passes a check, so that every survivor is a checked construction that may seed the next generation. The experiment asks whether learning which crosses survive produces more new constructions than crossing at random.

The domain. Let $f(n)$ be the largest N such that every 2-colouring of the subsets of $[n] = \{1, \dots, n\}$ has a monochromatic family of N sets closed under union and intersection,

TABLE X. The arms of the crossing network, five runs of 1200 checks each. “Pass” is the share of offspring that pass the check; “new” is the number of new colourings per run (mean $\pm$ standard deviation) and “optimal” the number of them with $m(n) = f(n)$ for $n = 6, \dots, 9$. Ill-typed offspring are those of the blind arm that fail the typing.

Arm	Pass	New	Optimal	Ill-typed
<i>free</i>				
network	58.7%	2.8 ± 0.4	1.8	0
select	38.1%	2.8 ± 0.4	2.0	0
random	3.6%	3.2 ± 1.0	0.8	0
blind	2.7%	5.6 ± 1.6	1.6	927
<i>novelty rule</i>				
network	1.3%	3.0 ± 0.0	2.0	0
select	1.4%	3.0 ± 0.0	2.0	0
random	1.4%	3.4 ± 1.5	1.0	0
blind	2.2%	5.2 ± 1.5	1.4	1002

Erdős problem 1183 [28]. A maximal chain has $n + 1$ members and each of its colour classes is closed under both operations, so $f(n) \geq \lceil (n+1)/2 \rceil$, and Chojecki proved $f(n) \leq (3 + o(1))n \log_2 n$ [44].⁷ A construction is a family of colourings, one for each n . Its genome is a typed expression tree: the leaves are words indexed by sizes (Thue–Morse, Rudin–Shapiro and Sturmian words, Legendre symbols modulo 5, 7, 11, 13, periodic words) and families of colourings (weighted sums of coordinates modulo 3 and 4, the parity of the least element), and the internal nodes are operations on words (sum modulo two, shift, reversal, dilation, interleaving) and on families (the level colouring of a word, sum modulo two, splitting the ground set, blowing up blocks, complementation, symmetrisation). Two parents breed by exchanging subtrees of the same type. The check computes the largest monochromatic sublattice $m(n)$ for $n = 6, 7, 8, 9$ with an exact search and keeps the offspring when $m(n) \leq f(n) + 1$; a colouring is new when its key, the colour profile of the sizes and of the elements, minimised over complementation of the sets and exchange of the colours, has not been met.

The arms. Each arm breeds 40 offspring in each of 6 generations, over 5 seeds, so every arm spends 1200 checks. The *network* draws parents by fitness, proposes eight typed crosses and evaluates the one ranked first by a logistic model of sur-

vival refitted every generation; *select* draws parents by fitness and evaluates one typed cross; *random* draws parents uniformly and evaluates one typed cross; *blind* draws parents uniformly and crosses without regard to types. The experiment was run twice, freely and under a novelty rule that skips, before any check, an offspring whose colouring the arm has already evaluated (Fig. 17).

The result (Table X). Without the novelty rule the network passes 58.7% of its offspring against 3.6% for random typed crossing, but almost all of them are colourings it has already found under another genome: it finds 2.8 new colourings per run against 3.2 for random crossing and 5.6 for blind crossing. Under the novelty rule the pass rates fall to between 1.3% and 2.2% and the order of the new colourings is unchanged. The learned model therefore learns to repeat survivors, and it does not beat the random controls in the quantity that matters. Blind crossing does best because an untyped exchange occasionally grafts a family where a word was expected and so reaches combinations of the operations that typed crossing never proposes; the typing removes ill-formed genomes and, with them, part of the reachable set. None of the bred colourings improves a known value: the best, from the random arm, has $m(n) = 7, 7, 8, 8, 9$ for $n = 10, \dots, 14$, which equals $f(n)$ at $n = 11, 13$ and exceeds it by one at $n = 10, 12, 14$.

What did move the frontier was a change of representation, not breeding. The optimal colourings found for $n = 7, 9, 10$ by a counterexample-guided search over all colourings are level colourings, in which the colour of S depends only on $|S|$; such a colouring is a word $c_0 c_1 \dots c_n$, with colour classes $L_0, L_1 \subseteq \{0, \dots, n\}$. A *weighted poset* is a finite poset P with a map $w: P \rightarrow \{1, 2, \dots\}$; $J(P)$ is its lattice of down-sets and $w(I) = \sum_{p \in I} w(p)$.

Lemma XVII.1. *A level colouring has a monochromatic sublattice of colour class L with N members if and only if there are a weighted poset (P, w) and an integer $t \geq 0$ with $t + w(P) \leq n$, $|J(P)| = N$ and $t + w(I) \in L$ for every $I \in J(P)$.*

Proof. Let D be a sublattice of $2^{[n]}$ with N members whose sizes lie in L . It is a finite distributive lattice, so by Birkhoff’s theorem it is isomorphic to $J(P)$ for the poset P of its join-irreducible members, by the map that sends $A \in D$ to the set of join-irreducibles below it. The function $A \mapsto |A|$ is strictly increasing on D and modular, since $|A \cup B| + |A \cap B| = |A| + |B|$. Write p_* for the unique lower cover of a join-irreducible p and $w(p) = |p| - |p_*| \geq 1$. If I is a down-set and p a minimal element of $P \setminus I$, then $I \cup \{p\}$ is the join of I and the principal down-set of p , and their meet is that principal down-set without p , which corresponds to p_* , so modularity gives $|I \cup \{p\}| = |I| + w(p)$; induction along a maximal chain of down-sets gives $|A| = t + w(I)$ with $t = |\min D|$, and $t + w(P) = |\max D| \leq n$. Conversely, given (P, w, t) , choose disjoint sets T and B_p ($p \in P$) in $[n]$ with $|T| = t$ and $|B_p| = w(p)$; the map $I \mapsto T \cup \bigcup_{p \in I} B_p$ is injective, carries unions and intersections of down-sets to unions and intersections of sets, and gives I the size $t + w(I) \in L$. $\square$

The lemma replaces the search over sublattices of $2^{[n]}$ by a search over weighted posets, which is small: the elements of P

⁷ D. Bhattacharjee, P. Mandal and U. Bhattacharya, *Resolving Erdős–Ulam monochromatic union-closed family conjectures*, arXiv:2610.02833v1 (2 October 2026), doi:10.48550/arXiv.2610.02833, with Lean 4 archive doi:10.5281/zenodo.23094057. The paper proves both conjectures of Erdős and Ulam on families closed under union only, in Lean, bounds $f(n)$ by $O(n \log^2 n)$ in its Theorem 1.4, and asks in its Question 9.3 whether $f(n) = O(n)$ and whether $f(n) = \lceil (n+1)/2 \rceil$ for every n . Its fifth version, archived with its Lean development under doi:10.5281/zenodo.23374973, proves in its Theorem 7.6 that every 2-colouring of $2^{[n]}$ has a red and a blue sublattice with at least $n + 1 + 2q + e$ members together, where $n + 1 = 20q + r$ with $0 \leq r \leq 19$ and $e = 1$ exactly when $r \geq 12$, so that $f(n) \geq \lceil (n + 1 + 2q + e)/2 \rceil \geq 11n/20$; the proof ends in a finite check of a window of twenty levels, run by four independent programs and formalised in Lean only for colourings that depend on the sizes of the sets. Theorem XVII.2 below gives the exact values for $n \leq 20$; the first question remains open.

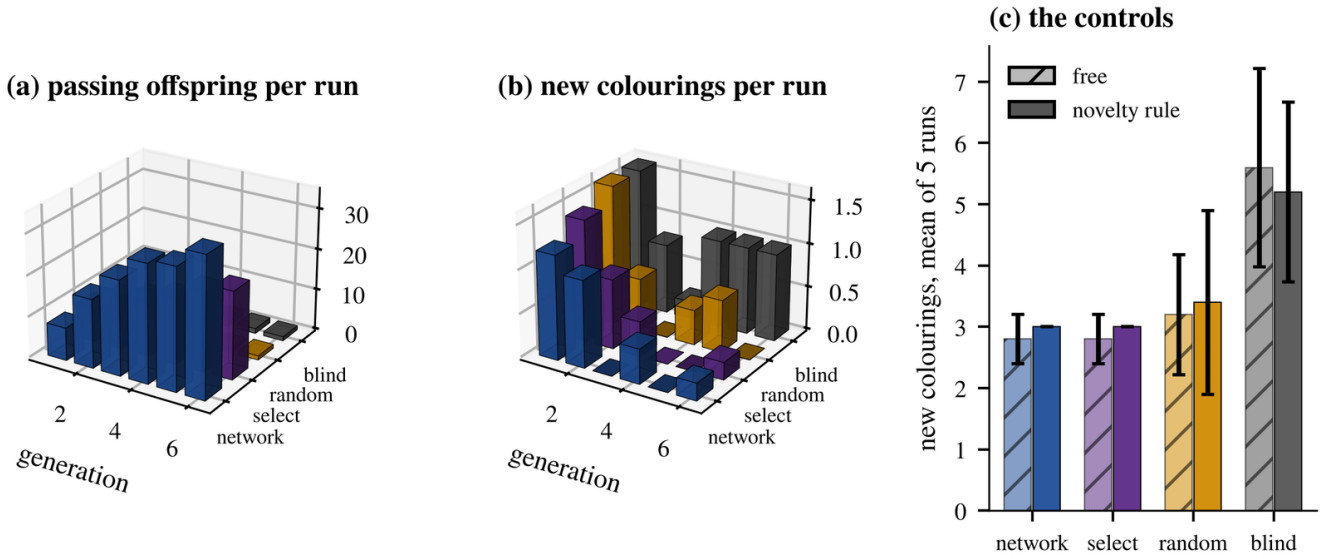


FIG. 17. The crossing network against its controls on Erdős problem 1183. (a) Offspring that pass the check, per generation, mean of five runs without the novelty rule. (b) Colourings not met before among them. (c) New colourings per run with and without the novelty rule, with one standard deviation. The network passes the most offspring and finds no more new colourings than the controls; the untyped arm finds the most.

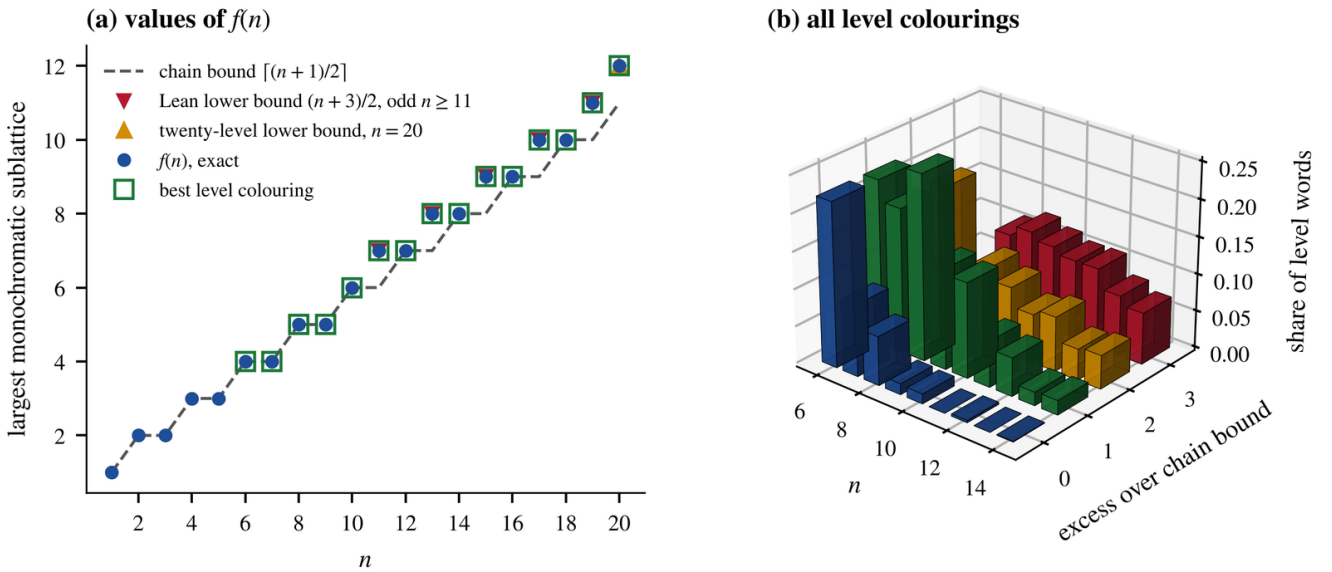


FIG. 18. Monochromatic sublattices of colourings of $2^{[n]}$. (a) The exact values of $f(n)$ for $n \leq 20$ (Theorem XVII.2), the chain bound $\lceil (n+1)/2 \rceil$, the Lean lower bound $(n+3)/2$ for odd $n \geq 11$ [129], the twenty-level lower bound 12 at $n = 20$, and the best level colouring, which attains $f(n)$ for every $n \leq 20$. (b) The distribution of the largest monochromatic sublattice over all level colourings for $6 \leq n \leq 14$, by its excess over the chain bound; the share of words that attain the bound falls quickly with n , and from $n = 11$ on no odd word does.

are added in the order of the weights of their principal down-sets, each new element is placed above a down-set already built and creates the down-sets $I \cup \{p\}$ for every I above that one, and a branch stops as soon as a new size leaves L . On 120 random level colourings with $3 \leq n \leq 9$ the poset search and the general search agree on the largest monochromatic sublattice in every case, and on 30 at $n = 10$ they agree on every value below the cap of the poset search. A grid version, for colourings that depend on the sizes of S in the blocks of

a fixed partition of $[n]$, holds with weights in $\mathbb{N}^k \setminus \{0\}$ by the same proof.

Theorem XVII.2. For $1 \leq n \leq 19$,

$$f(n) = \begin{cases} \lceil (n+1)/2 \rceil & \text{if } n \leq 10 \text{ or } n \text{ is even,} \\ (n+3)/2 & \text{if } n \geq 11 \text{ is odd,} \end{cases}$$

and $f(20) = 12$. For $11 \leq n \leq 20$ the value is attained by the level colouring of the word in the table below.

n	word $c_0 \cdots c_n$	n	word $c_0 \cdots c_n$
11	110111001000	16	10011001011100100
12	1010011100100	17	110100111001101000
13	10111001101000	18	0011001011100110100
14	011010011100100	19	00110010111000101101
15	1100111010011000	20	000101100111001011100

Proof. The lower bounds are the chain bound and, for odd $n \geq 11$, the theorem that every colouring then has a monochromatic sublattice with $(n+3)/2$ members, proved in Lean 4 by the OpenRefute project [129]. For $n \leq 10$ the upper bound is attained by explicit colourings with no monochromatic sublattice of $\lceil (n+1)/2 \rceil + 1$ members, each checked by an exact search and by a satisfiability query that shares no code with it. For $11 \leq n \leq 19$ the words of the table have, by Lemma XVII.1 and the exhaustive poset search, no monochromatic sublattice larger than the stated value; for $n = 11$ this is confirmed by the exact search over all sublattices and by a satisfiability query. For $n = 20$ the poset search finds 59 level colourings of $2^{[20]}$, up to reversal and exchange of the colours, with no monochromatic sublattice of 13 members, and the lower bound $f(20) \geq 12$ is the twenty-level window of the footnote, which at $n = 20$ gives a red and a blue sublattice with 23 members together. $\square$

The theorem answers the second part of the question of the footnote in the negative from $n = 11$ on, as OpenRefute already did, and leaves the first open. Among level colourings (Fig. 18) the chain bound is attained for every even $n \leq 18$ and exceeded by one for every odd n from 11 to 19; up to reversal and exchange of the colours there are 9, 7, 6, 8, 3, 2 optimal words for $n = 8, 10, 12, 14, 16, 18$, and none for $n = 20$, where the best level colouring has 12 members against the chain bound 11, and 12 is $f(20)$. No optimal word for $n = 8$ extends, two letters at a time on the right, through optimal words to $n = 18$. The lemma shows the obstruction: three sizes of one colour in arithmetic progression already give a sublattice with one member more than the sizes, and a progression of length $d + 1$ gives 2^d members, so a good word must avoid long monochromatic progressions in each colour class while the number of its colour changes grows with n . At $n = 20$ no colouring at all attains the chain bound.

What the experiment shows. The chain reaction was asked to produce new constructions faster than random crossing, and it did not; its survival model learned to propose what had already survived. The one advance on the problem came from a proof, Lemma XVII.1, that changed what could be checked: it turned the general search, which reaches $n = 11$, into a search over weighted posets, which reaches $n = 20$. In the terms of Theorem I, the network sharpened the profile of a proposer whose support did not contain the needed idea; the lemma enlarged the support.

C. Yang–Mills and Hodge: crossing on the open steps

The crossing network of the package (Section XVII B describes it on a calibration problem) is aimed here at the open leaves of the two trees. Every construction of its library is a

typed node: it gives an output of one type and needs inputs of others. The Yang–Mills library holds the single-link law of $SU(2)$, which is a von Mises–Fisher law on S^3 ; the sharp quarter inequality $W_1(v_b, v_{b'}) \leq \frac{1}{4}|b - b'|$ of [18]; Dobrushin’s uniqueness criterion [62]; three dependency graphs of the links (no gauge fixing, temporal gauge, and the maximal-tree gauge of Creutz [51] with one fixed link in every plaquette); reflection positivity and the transfer matrix [130]; the gauge Dobrushin–Shlosman criterion and the Bakry–Émery bound of [165]; the Polchinski flow and the multiscale log-Sobolev inequality [16]; Bałaban’s renormalisation group [14]; and a finite subgroup of $SU(2)$. The Hodge library holds Markman’s theorem on Weil classes of abelian fourfolds [115], the reduction of Moonen and Zarhin in dimension at most five [122], Schoen’s products [152], Shioda’s structure of Fermat varieties [157], the Hodge characters of the Fermat fourfold of degree 110 left by the census of [17], Buskin’s theorem on Hodge isometries of K3 surfaces [38], and an unrefereed claim of the Hodge conjecture for CM abelian varieties [125]. Two nodes cross when the output of one is an input of the other. An offspring that needs nothing more is closed and goes to its check, a computation that accepts, rejects with the measurement that stopped it, or stops when no checkable form exists. Checked offspring join the pool and seed the next generation.

The first round. Over five seeds of five generations of 40 crosses, the network bred 180 offspring per run, checked 64 and accepted 8; select bred 8.8, checked 3.6 and accepted 0.6; random bred 200, of which 191.2 were ill-typed, and accepted 0.6 (Fig. 19a). Every run of the network reached the composite of Dobrushin’s criterion, the sharp quarter, the von Mises–Fisher law and the maximal-tree gauge, whose check counts 12 active partners for every active link and accepts the range $\beta_W < \frac{1}{3}$; no run of the controls reached it. The network enumerates every typed pair, so its runs do not depend on the seed. The composite was first proposed by the language-model proposer of the package, which suggested fixing a gauge to cut the number of partners; a satisfiability search found the pattern of Lemma XVII.3, Lean 4 checked its finite part, and the cross was then entered in the library, from which the network rederives it. The network therefore reproduces this step; it did not originate it.

The step. We use the normalisation of [18, 165]: the Wilson measure of $SU(2)$ on a finite set of links has density $\exp(2\beta \sum_p \text{Re Tr } U_p)$ with respect to Haar measure, where U_p is the ordered product around the plaquette p , and $\beta_W = 4\beta$ is the coupling of the physics literature. A link of $\mathbb{Z}^4$ is written (x, μ) for the pair $\{x, x + e_\mu\}$; the plaquette at x in the plane $\mu < \nu$ consists of the links (x, μ) , $(x + e_\mu, \nu)$, $(x + e_\nu, \mu)$, (x, ν) .

Lemma XVII.3. *Let $\sigma: (\mathbb{Z}/2)^4 \rightarrow \{0, 1, 2, 3\}$ take the values 0, 0, 2, 3, 3, 1, 2, 1, 1, 2, 1, 3, 3, 2, 0, 0 at $x_0 + 2x_1 + 4x_2 + 8x_3 = 0, 1, \dots, 15$, and let F be the set of links $(x, \sigma(x \bmod 2))$, $x \in \mathbb{Z}^4$. Then*

- (i) every plaquette of $\mathbb{Z}^4$ contains exactly one link of F ;
- (ii) F is a forest, and so is $F \cap E(\Lambda)$ for every set Λ of sites, $E(\Lambda)$ being the links with both ends in Λ ;

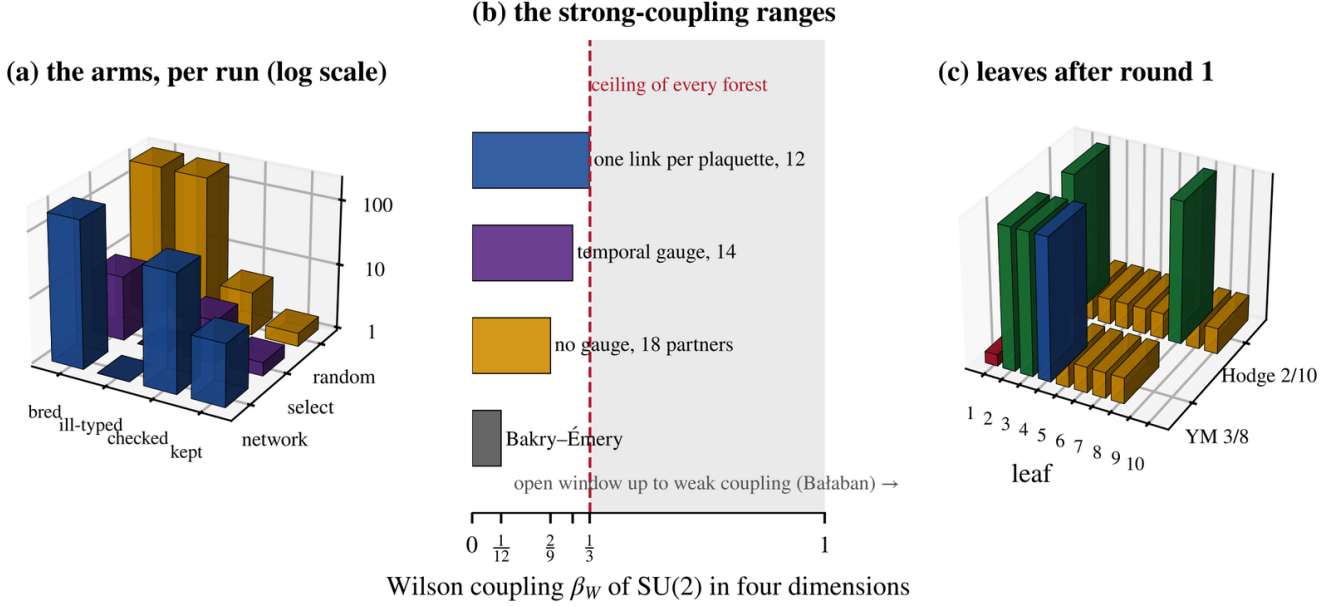


FIG. 19. The first headline round. (a) Offspring bred, ill-typed, checked and kept per run of each arm, on a logarithmic scale. The network enumerates the typed crosses; select draws pairs at random and keeps those whose types fit; random crosses whatever the types. (b) The ranges of four-dimensional SU(2) on which a lattice mass gap is proved: Bakry–Émery [165], single-link Dobrushin without gauge fixing [18], and after the temporal and the one-per-plaquette gauge fixings (Theorem XVII.4). The dashed line is the ceiling of every forest gauge fixing (Proposition XVII.5); beyond it the window up to weak coupling is open. (c) The leaves of the two trees after the round, in the order of Table XIII: proved (green, the new step in blue), open (amber) and blocked (red).

(iii) every link $e \notin F$ shares a plaquette with exactly 12 links not in F , two in each of its six plaquettes.

Proof. (i) The plaquette at x in the plane $\mu < \nu$ contains $\#\{\sigma(x) \in \{\mu, \nu\}\} + [\sigma(x + e_\mu) = \nu] + [\sigma(x + e_\nu) = \mu]$ links of F , a number that depends only on $x \bmod 2$ and the plane; inspection of the $16 \times 6 = 96$ cases of the table gives 1 in each.⁸ (ii) Every site x has exactly one link of F leaving it in the positive direction, the link to $x + e_{\sigma(x)}$, and that link raises $h(x) = x_0 + x_1 + x_2 + x_3$ by one. In a cycle of k links of F on k sites, each link is the positive link of one of its ends and each site has at most one positive link, so each site of the cycle is the lower end of exactly one of its links and the cycle is directed; then h increases strictly around it, which is impossible. A subgraph of a forest is a forest. (iii) A link lies in six plaquettes; by (i) each of them has exactly one link of F , which is not e , so it has two active links besides e . Two distinct plaquettes meet in at most one link, so the 12 partners are distinct. $\square$

On a periodic box $(\mathbb{Z}/L\mathbb{Z})^4$ the set F contains cycles, since h is not defined modulo L , and the argument below never uses a periodic box. Let γ^F be the specification on the links of

$\mathbb{Z}^4$ not in F with a priori Haar measure and formal density $\exp(2\beta \sum_p \text{Re Tr } U_p^F)$, where U_p^F is the plaquette product with the links of F replaced by 1; for a box Λ let μ_Λ^F be the finite-volume measure on the active links of Λ with the plaquettes of Λ only, and W_Λ the Wilson measure on $E(\Lambda)$ with free boundary.

Theorem XVII.4 (Gauge-fixed strong coupling). *For four-dimensional SU(2) with $0 \leq \beta_W < \frac{1}{3}$:*

- (i) γ^F has exactly one Gibbs measure μ_F , and its truncated correlations of Lipschitz local functions decay exponentially with every rate $\lambda < -\log(3\beta_W)$ in the distance of their supports;
- (ii) for every gauge-invariant local function O the limit $\omega(O) = \lim_\Lambda W_\Lambda(O)$ exists along every sequence of boxes that exhausts $\mathbb{Z}^4$, and $\omega(O) = \mu_F(O^F)$, where O^F is O with the links of F set to 1;
- (iii) ω is invariant under translations and the hypercubic group and reflection positive, and its transfer matrix has a spectral gap $m \geq -\log(3\beta_W)$.

Proof. (i) The conditional law of an active link e under γ^F is the von Mises–Fisher law ν_b with $b = 4\beta$ times the sum of the six staples of e , each staple being the product of the other links of a plaquette through e , with the links of F equal to 1. Moving an active link e' moves at most one staple of e , since two links share at most one plaquette, and moves it by an isometric image of the displacement, since multiplication by unit quaternions is an isometry of $\mathbb{R}^4$. By the sharp quarter

⁸ The inspection is a finite check. The eBLLM package carries it out in Lean 4, in the theorem `one_fixed_per_plaquette` of its file `EBLLM/GaugeFix.lean`, proved by `decide`, and repeated independently on a box of side six by `verify_gauge.py`. The table was found by a satisfiability search over the 4^{16} functions σ ; it has eight solutions of period two, and none of period three.

inequality [18, Theorem A] the Dobrushin coefficient in the chordal metric of S^3 satisfies $C_{ee'} \leq 4\beta \cdot \frac{1}{4} = \beta$, and $C_{ee'} = 0$ unless e' is one of the 12 partners of Lemma XVII.3(iii). Every row and column of C therefore sums to at most $\alpha = 12\beta = 3\beta_W < 1$, and the Kantorovich–Dobrushin comparison lemma of [18, Appendix] gives uniqueness and decay at every rate $\lambda < -\log \alpha$, exactly as in the proof of [18, Theorem E].

(ii) Fix a box Λ . By Lemma XVII.3(ii) the set $F_\Lambda = F \cap E(\Lambda)$ is a finite forest. Root each of its trees and, given the values U_F on F_Λ , define $g: \Lambda \rightarrow \text{SU}(2)$ by $g = 1$ at the roots and, along each tree, $g_y = g_x U_{xy}$ for every link (x, y) of the tree; this is consistent because the tree has no cycle, and it makes $g_x U_{xy} g_y^{-1} = 1$ on F_Λ . For fixed U_F the map $U_e \mapsto g_x U_e g_y^{-1}$ on the active links is a product of left and right translations and preserves Haar measure, and the Wilson action is gauge invariant. Hence under W_Λ the gauge-transformed active links are independent of U_F and have law μ_Λ^F , and $W_\Lambda(O) = \mu_\Lambda^F(O^F)$ for gauge-invariant O . No holonomy is left over, because F_Λ has no cycle. Now let $\Lambda \subset \Lambda'$ be boxes and compare μ_Λ^F with $\mu_{\Lambda'}^F \otimes \text{Haar}$ on the active links of Λ' . Both are invariant under the single-link resampling of $\mu_{\Lambda'}^F$ at every active link all of whose plaquettes lie in Λ ; the remaining links form a set S within distance one of the complement of Λ . The comparison lemma of [18, Appendix], with $\theta = \alpha e^\lambda < 1$, couples the two measures so that the expected distance on the support T of O^F is at most $\frac{2\theta}{1-\theta} \sum_{z \in S} e^{-\lambda \text{dist}(z, T)} \leq c_T R^3 e^{-\lambda R}$, with $R = \text{dist}(T, \mathbb{Z}^4 \setminus \Lambda)$ and c_T independent of Λ' . As O^F is Lipschitz, $|W_{\Lambda'}(O) - W_\Lambda(O)| \leq c_O R^3 e^{-\lambda R}$. Any two boxes that contain the support at distance R lie in a common larger box, so $W_\Lambda(O)$ converges along every exhausting sequence to one limit $\omega(O)$. The same coupling with μ_F in place of $\mu_{\Lambda'}^F$, which is invariant under all single-link resamplings of γ^F , gives $\omega(O) = \mu_F(O^F)$.

(iii) For a translation or a symmetry τ of the lattice, $W_\Lambda(O \circ \tau) = W_{\tau\Lambda}(O)$ by the symmetry of the Wilson action, and the boxes $\tau\Lambda$ also exhaust $\mathbb{Z}^4$; by (ii) the limit is $\omega(O)$. Only the boxes move, not F . For a reflection ϑ in a lattice hyperplane through sites or bisecting links, the free-boundary measures W_Λ on boxes symmetric under ϑ are reflection positive [130], and by (ii) ω is their limit, so $\omega(\vartheta(\bar{G})G) \geq 0$ for every local G supported on one side. Finally, for a gauge-invariant Lipschitz G supported at times in $[0, s]$, part (i) applied to $\vartheta(\bar{G})^F$ and $(\tau_t G)^F$, whose supports are at distance at least $t - s - 1$, gives $0 \leq \omega(\vartheta(\bar{G})\tau_t G) - |\omega(G)|^2 \leq C_G e^{-\lambda t}$. Such functions are dense in $L^2(\omega)$ on each bounded time interval, so [18, Proposition gapdecay] gives a gap at least λ for every $\lambda < -\log(3\beta_W)$. $\square$

The theorem extends [18, Theorem F], where the range is $\beta_W < \frac{2}{9}$ and the state is the limit of periodic boxes, to $\beta_W < \frac{1}{3}$, with the state the limit of free boxes. It has been checked here and not refereed. A search of the literature on strong-coupling expansions found no earlier statement of this range: [130] and the cluster expansions that followed give convergence at strong coupling without an explicit Dobrushin range, [165] gives $\beta_W < \frac{1}{12}$, and the Lean project of strong-coupling

lattice gauge theory cited in [18] has $\beta < 1/(32n(d-1))$.⁹ The gain over [18] comes from the dependency graph and not from the metric: after a forest gauge fixing a plaquette has three active links instead of four, and each link loses six of its 18 partners.

Proposition XVII.5 (The ceiling of forest gauge fixings). *Let F' be any set of links of $\mathbb{Z}^4$ whose intersection with every box is a forest. Then the single-link Dobrushin coefficients of $\gamma^{F'}$, in the chordal or the geodesic metric of S^3 , have $\sup_e \sum_{e'} C_{ee'} \geq 12\beta$. Hence no single-link Dobrushin argument after a forest gauge fixing proves uniqueness for $\beta_W \geq \frac{1}{3}$.*

Proof. In a box of side n there are n^4 sites, $4n^4 - O(n^3)$ links and $6n^4 - O(n^3)$ plaquettes, and a forest has at most $n^4 - 1$ links, so at least $3n^4 - O(n^3)$ links are active. Let a_p be the number of active links of the plaquette p ; then $\sum_p a_p = 6 \cdot \#\{\text{active links}\} - O(n^3)$, since every link lies in six plaquettes, and the mean $\bar{a}$ of a_p is at least $3 - o(1)$. An active link e has $\sum_{p \ni e} (a_p - 1)$ active partners, so the sum of the partner counts over the active links is $\sum_p a_p(a_p - 1) \geq \sum_p a_p(\bar{a} - 1)$ by the Cauchy–Schwarz inequality, and the mean partner count is at least $6(\bar{a} - 1) \geq 12 - o(1)$. Partner counts are integers, so some active link has at least 12 partners. For each partner e' of such a link, the staples of e involve pairwise disjoint sets of other links and can be chosen with sum zero while the staple through e' is arbitrary, and the proof of [18, Proposition ceiling] gives $C_{ee'} \geq \beta$. $\square$

Passing to a weighted Dobrushin condition does not help either, since the spectral radius of a nonnegative matrix is at least its mean row sum over a box, up to the boundary.

The amplified round. The second round adds to the library the crosses aimed at the assumption of each ceiling, the partners of the gauge-fixed specification, and the nodes obtained by transferring the pattern of the step that moved, *remove redundant variables by a symmetry, then apply a local criterion*. It also passes every accepted offspring through a refuter before it counts, and adds the booster layer described below. Up to depth four the library breeds 21,515 composites, of which 1,466 are closed. Of these, 934 stop without a checkable form, 433 are rejected by their check, and 99 are accepted. The refuter breaks 35 of the accepted ones, 26 because a stronger range is already known and 9 because they reproduce an accepted range with no gain, and 64 survive (Table XI).

- *Plaquette variables* [15]. In these variables the Bianchi identity of each cube fixes every plaquette variable, up to parallel transport, by the other five plaquettes of the cube. The single-variable kernels are therefore point masses, so Dobrushin’s coefficients equal one and the cross is rejected.

⁹ The search covered the arXiv listings in math-ph, hep-lat and math.PR, the citations of [165], the gauge-fixed formulations of Balaban’s renormalisation group, and the expository literature on maximal-tree gauges after [51]. The monograph of Seiler on gauge theories as statistical mechanics could not be read from the session, and its strong-coupling results are cluster expansions without an explicit range of the Dobrushin type.

TABLE XI. The amplified round on the two libraries, up to depth four. “Survived” counts accepted composites that the refuter could not break. The surviving Yang–Mills composites are 18 forms each of the gap for $\beta_W < \frac{2}{9}$, $< \frac{2}{7}$ and $< \frac{1}{3}$, and 3 of uniqueness in each range; the Hodge survivor is (F2) in dimension at most five, which was known.

	Closed	Stopped	Rejected	Refuted	Survived
Yang–Mills	1,460	932	430	35	63
Hodge	6	2	3	0	1
Total	1,466	934	433	35	64

- *Blocks of active links.* A block coefficient needs the Kantorovich sensitivity of the joint law of a block at finite β , and the sharp quarter does not control it; the cross stops.
- *Path coupling.* The single-link Glauber dynamics contracts on exactly the Dobrushin range, so the cross gains nothing.
- *Bakry–Émery after the forest gauge fixing.* The Hessian of ReTr of a product of k active links is at most $2k$ times the squared norm of the tangent vector. With three active links per plaquette instead of four, the curvature condition of [165] improves from $\beta_W < \frac{1}{12}$ to $\beta_W < \frac{1}{9}$, which the Dobrushin range dominates.
- *Dobrushin–Shlosman on a box of the gauge-fixed specification.* This needs the sensitivity of a box law to its boundary at order-one coupling, an integral over $\text{SU}(2)^{3n^4}$; the estimate of [18] puts the box side near 90, and the cross stops.
- *Pattern transfer to Fermat fourfolds.* Shioda’s criterion is invariant under $\mu_m^6 \rtimes S_6$ and under the Galois group. The leftover character $(1, 16, 26, 31, 36)$ at level 55 has an orbit of 8 members under the units of $\mathbb{Z}/55$, and none of them splits into zero-sum pairs and triples, so the quotient by the symmetries gives nothing new.
- *Pattern transfer to K3 squares.* For a K3 surface S with real multiplication by $\mathbb{Q}(\sqrt{d})$, the element $\sqrt{d}$ is a Hodge similitude $T(S) \rightarrow T(S)(d)$ and not an isometry, so Buskin’s theorem does not apply to it. Composites through K3 surfaces S' with $T(S') \cong T(S)(d)$ are similitudes too. Varesco reduces the algebraicity of Hodge similitudes to the Kuga–Satake Hodge conjecture [176], and he proves it on the first four-dimensional families. The transferred node therefore reproduces a known route, and it stops at the very general member of the larger families.

D. Boosters

The booster layer of Section XV was tested on the Yang–Mills library at a budget at which no arm can enumerate every typed pair: 6 crosses in each of 8 generations, over 200 seeds. Two forms were run, one synapse per pair of pool members and one per pair of library heads, so that a weight learned on one composite carries to every composite with the same heads; in a second experiment the boosted arm keeps its weights for four rounds (Fig. 20).

TABLE XII. Boosted, unboosted and random crossing at a budget of 48 crosses, 200 runs per arm. “Reached $\frac{1}{3}$ ” counts runs whose survivors include $\beta_W < \frac{1}{3}$, with the mean generation at which they reach it; survivors and refuted offspring are means per run.

Synapse	Arm	Reached $\frac{1}{3}$	Generation	Survivors	Refuted
pair	boosted	14	5.86	0.61	1.14
head	boosted	12	5.58	0.62	1.16
–	unboosted	21	5.43	0.68	1.25
–	random	0	–	0.08	0.05

Boosters do not beat the unboosted network (Table XII). Within one round the boosted arms reach the step in 12 and 14 of 200 runs against 21 for the unboosted network; the difference is within two standard errors. Across four rounds the boosted share moves from 0.067 to 0.12 and the unboosted share from 0.113 to 0.067, again within two standard errors in every round. Both are far above random pairing, which wastes 45 of its 48 crosses on ill-typed pairs and never reaches the step. The reason is the one met in Section XVII B: a booster strengthens synapses that have already fired, and the step that moved needed a synapse that had never fired, between Dobrushin’s criterion and the maximal-tree gauge. The typing, not the learning, carries the advantage of the network.

E. Gap numbers and the construction graph

Gap numbers. Each round reports two numbers. For Yang–Mills it is the largest coupling below which the lattice gap is proved, and its share of the interval up to $\beta_W \approx 2.2$, where Monte Carlo simulations of four-dimensional $\text{SU}(2)$ see the crossover from strong to weak coupling [52]; this is a physical marker and not the threshold of any proof, since Baĭaban’s control is not explicit in the coupling. The number moved from $\frac{2}{9}$, or 10.1%, before the rounds to $\frac{1}{3}$, or 15.2%, in the first round, and stayed there in the second. For Hodge it is, for each open family, the largest dimension of a family on whose general member the classes are known to be algebraic, against the dimension of the largest family of its kind (Table XIII). No Hodge number moved.

What blocks. On Yang–Mills the step that blocks is Hypothesis R: a lattice gap uniform in the spacing, for which Baĭaban’s effective action at a fixed physical scale must land in a region where a convergent method applies. The strong-coupling side now reaches $\beta_W = \frac{1}{3}$, the ceiling of single-link arguments after any forest gauge fixing, and going further needs multi-link blocks whose joint laws are controlled at finite coupling. On Hodge every open leaf is blocked by the absence of a construction of the classes: of Weil classes off the products of lower Weil varieties, of the classes of $\sqrt{d}$ on squares of K3 surfaces outside Varesco’s families, and of the leftover class of the Fermat fourfold of degree 110.

The construction graph. The package records every node, every crossing with its parents, check and refuter verdict, the synapse weights of every run, and the map of the other workstreams that feed the library (Fig. 21). Four workstreams feed

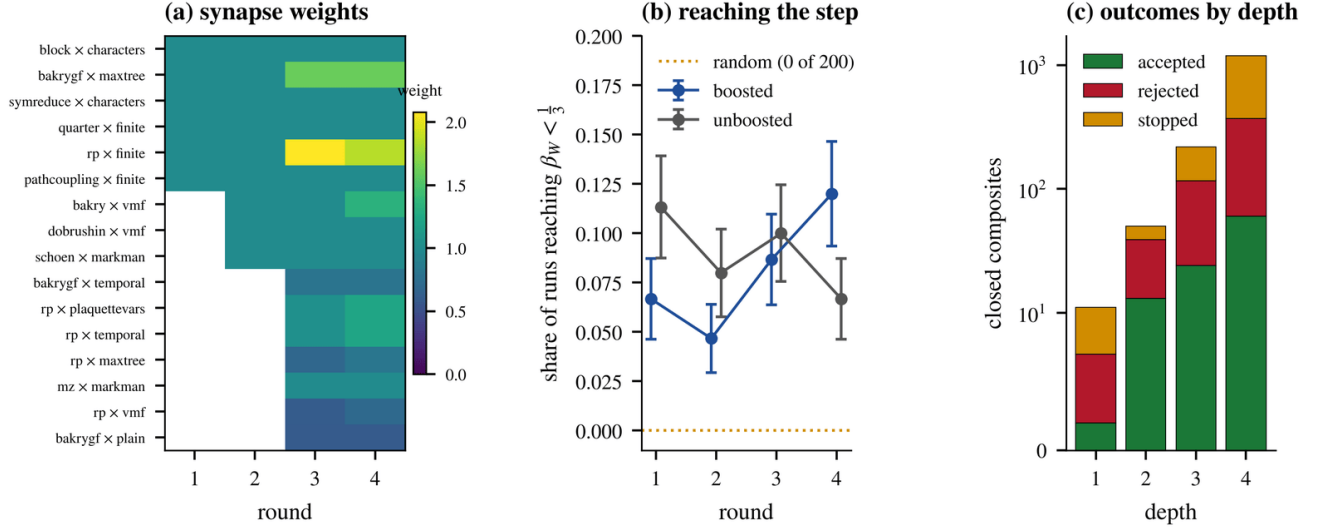


FIG. 20. The booster layer. (a) The weights of the synapses of one boosted run over four rounds, one row per synapse; a blank cell is a synapse not yet created. (b) The share of 200 runs that reach $\beta_W < \frac{1}{3}$ within 48 crosses, for the boosted arm that keeps its weights from round to round and for the unboosted network, with one standard error; random pairing never reaches it. (c) The outcomes of the 1,466 closed composites of the amplified library, by depth.

it. The Yang–Mills archive [18] gives the sharp quarter, reflection positivity and the box criterion; the Hodge papers [17] give the split into (F2) and (F3') and the open families; the census of Fermat varieties gives the characters left at 110 and 220; and the weighted-poset lemma of Section XVII B gives the pattern of a change of representation, which the amplified round transfers to plaque variables, blocks and the quotient of Fermat characters. The work on the lattice D_4 and the 24-cell gives the pattern of a finite certificate, which the box nodes use. The Lehmer and Earth–Moon workstreams feed neither tree.

F. Memory

The memory study runs the whole package over the five libraries for 5 rounds and 60 seeds, with 8 generations of 10 crosses per round, in five arms: the full system (boosters, memory, retrieval, the committee of proposers); the full system without retrieval, which keeps the memory but does not let it steer; boosters alone; no boosters; and random pairing. Every arm without memory starts each round from the library and keeps only its synapse weights. The measured quantities are the distinct survivors after each round and the checks spent on them (Fig. 22).

Every arm except random pairing ends between 6.0 and 6.4 distinct survivors, because the five libraries hold that many checkable statements within reach of the crosses; all of them are known results. The arms differ in cost (Table XIV). With memory the system reaches the plateau with 46.4 checks, against 95.2 for boosters alone, and spends none on composites already decided; without memory a third of the checks after the first round are repeats. Retrieval steers the draw but

changes little, 46.4 checks against 49.8: the saving comes from not checking a decided composite twice, which is the soundness bookkeeping of Section VIII applied across rounds. Memory therefore halves the cost of reaching what the libraries can reach, and it does not move the plateau.

G. The channel and the agents

The channel study runs the five agents on one shared memory for 6 rounds and 60 seeds in three arms: the channel with the target weights, the target weights alone, and neither. The three arms spend 211.8, 218.0 and 216.8 checks per seed over the six rounds, with standard errors below one check, and end with 24.05, 24.15 and 24.23 distinct survivors. The channel saves three per cent of the checks, because a bundle that carries a failed cell halves the weight of its candidate, and it costs a tenth of a survivor for the same reason. In none of the 180 runs does a survivor reach a target point. The channel delivers everything the engine holds, and what the engine holds are outputs of library constructions; no library holds the input that a target point needs (Section XVI B).

The continuous loop runs the five agents and, beside them, a single-loop control: one attack over all five libraries with boosters and no memory, 6 generations of 40 crosses per round. The agents use the channel, the target weights and the local model. After eleven rounds the agents hold 24 distinct survivors for 250 checks and the control 27 for 797 (Fig. 22(c)). The agents reach 23 survivors in the second round, for 142 checks, and 24 in the fifth; the control needs five rounds and 348 checks to reach 23 and passes the agents in the sixth round, still adding a known result every few rounds at about 70 checks per round. No survivor of either reaches a

TABLE XIII. The leaves of the two benchmark trees after the second round, with the result that settles a proved leaf, the check that stopped the last attempt on an open one, and, for the open families of the Hodge tree, the gap number: the largest dimension of a family on whose general member the classes are known to be algebraic, against the dimension of the largest family of its kind. Yang–Mills: 3 of 8 leaves proved (37.5%); Hodge: 2 of 10 (20%).

Leaf	Status	Source, or what blocks	Gap
<i>Yang–Mills</i>			
1 a faithful Lean statement of the Clay problem	blocked	no formal statement of the Wightman axioms for a gauge theory exists to state it against	–
2 lattice gap at strong coupling	proved	$\beta_W < \frac{2}{9}$ [18]	–
3 fine-scale control at weak coupling	proved	Balaban’s renormalisation group [14]	–
4 the window from below	proved	$\beta_W < \frac{1}{3}$, Theorem XVII.4; the ceiling of the method, Proposition XVII.5	15.2%
5 a finite-box mixing certificate in the window	open	the box needs side about 90; no computation over $SU(2)^{4n^4}$ at that size exists	–
6 a lift of the two-dimensional gap through dimension	open	no family of theories interpolating in dimension with a uniform gap is known	–
7 a multiscale log-Sobolev inequality at order-one coupling	open	the method of [16] needs a Polchinski flow with uniformly convex effective potentials; for gauge fields none is known, and gauge invariance removes the Gaussian reference measure	–
8 the continuum limit with the Wightman axioms	open	needs a lattice gap uniform in the spacing, Hypothesis R, which is the Clay problem on the lattice	–
<i>Hodge</i>			
1 a faithful Lean statement	blocked	Hodge classes and algebraic cycles of complex projective varieties are not yet formalised	–
2 (F2) in dimension at most five	proved	[115, 122]	–
3 (F2) for CM abelian varieties	open	the only claim is unreferenced [125]; the refuter rejects it	–
4 Weil classes on Weil eightfolds of nontrivial discriminant	open	no construction of the classes is known	0/16
5 a semiregular object on E_0^6 meeting the polarised criterion	open	every convolution tested in [17] has too large an Ext^2	–
6 Weil classes on Weil sixfolds	open	products of Weil fourfolds and Weil surfaces fill a family of dimension 5 in the moduli of dimension 9; a very general Weil sixfold is simple	5/9
7 (F3′) for $S \times S$, S a K3 surface with real multiplication	open	$\sqrt{d}$ is a Hodge similitude; Varesco’s reduction to the Kuga–Satake Hodge conjecture [176] covers the first four-dimensional families, the largest family has dimension 8	4/8
8 Fermat fourfolds, degrees with split characters	proved	Shioda’s zero-sum criterion [157] with the census of [17]	–
9 the Fermat fourfold of degree 110	open	the leftover character has a lattice point in its Fine interior, so no block family covers it, and its orbit under the symmetries has no split member	one orbit
10 the Fermat fourfold of degree 220	open	the classes lifted from degree 110	–

TABLE XIV. The memory study: five rounds, 60 seeds, means per seed. “Repeated” counts checks of composites decided in an earlier round.

Arm	Checks	Repeated	Survivors	Refuted
full system	46.4	0.0	6.07	5.2
no retrieval	49.8	0.0	6.02	5.3
boosters only	95.2	33.9	6.18	10.6
no boosters	96.3	33.6	6.42	10.6
random	2.5	0.1	0.47	0.3

target point. Agents are cheaper for the same reason as memory, since each agent skips what its slice of the store has decided, and they reach no further, since a target point needs an input that no library holds; the control’s three extra survivors are known results that the agents’ narrower libraries do not contain.

H. The barrier and pattern matrix

The last experiment is done by hand and recorded in the package. For each blocking step, every pattern that broke a comparable barrier elsewhere is transferred, and the transferred statement, its needed input and the outcome are written as one cell: *closes* when the pass line is crossed, *partial* when a weaker statement follows or a new fact is computed, *fails* when the needed input is false or unknown, with the reason. There are 38 cells. None closes; 7 are partial and 31 fail. The partial cells are the forest gauge fixing and its ceiling, the box Dobrushin–Shlosman criterion as a finite check not yet run, size colourings and Theorem XVII.2, the forced arithmetic progressions in level colourings, the Kuga–Satake reduction of Varesco, the averaged Chowla theorem, and the grid search at $n = 20$. The failures fall into three kinds: the input is known to be false (product colourings and m-sequence colourings for Erdős problem 1183, plaquette variables, abelian duality for $SU(2)$), the input is the open problem itself or another benchmark (motivated cycles, which are algebraic exactly when the standard conjecture holds; the inverse of L as a Hodge class on $X \times X$), or the input needs a small parameter that is not small

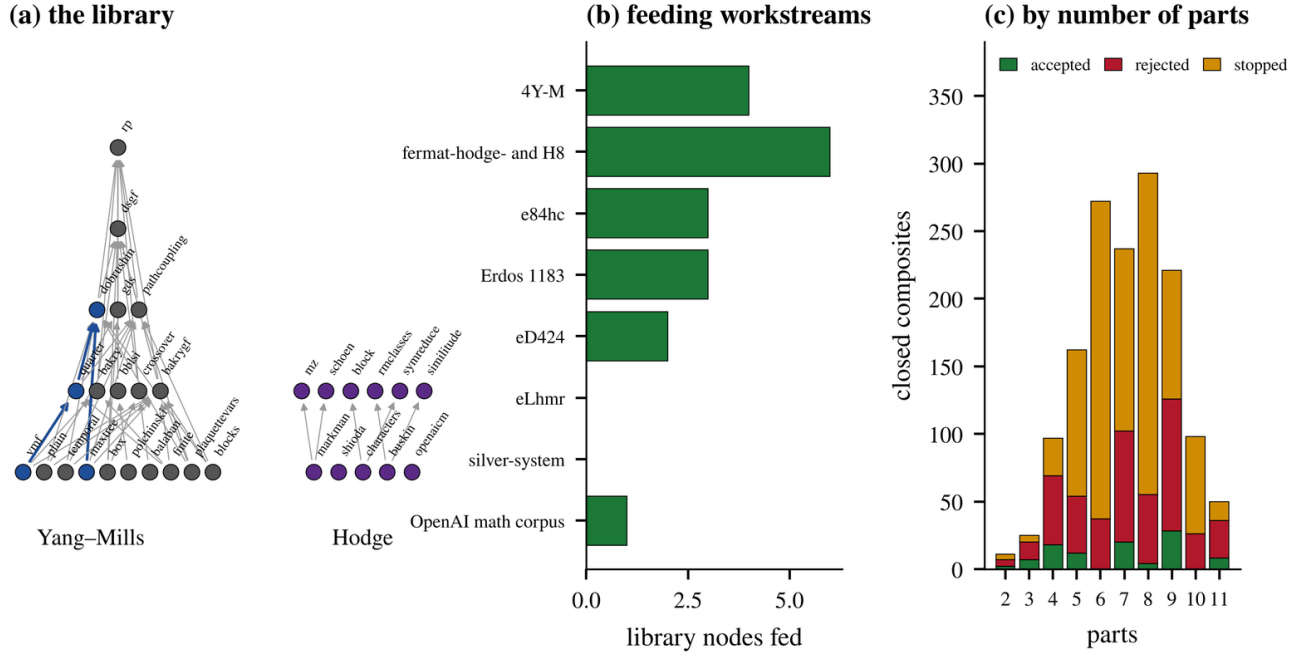


FIG. 21. The construction graph. (a) The library of the amplified round, with an arrow from b to a when b can be fed into a , layered by the length of the longest chain of inputs; Yang–Mills nodes on the left, Hodge nodes on the right, and the four parents of the composite that proves $\beta_W < \frac{1}{3}$ in blue. (b) The number of library nodes fed by each workstream. (c) Closed composites of the amplified library by number of parts and outcome.

at the blocking step (Balaban, Brydges–Yau, Kotecký–Preiss). The matrix is the content of the cells π_{ch} of (15) reads, so a pattern recorded as failing on a benchmark is drawn half as often there.

Two cells are new computations and one is a new negative result.

Proposition XVII.6. *Let c_1, c_2 be level colourings of $2^{[a]}$ and $2^{[b]}$ attaining $f(a)$ and $f(b)$, and colour $S \subseteq [a+b]$ by $\phi(c_1(S \cap [a]), c_2(S \setminus [a]))$ for a Boolean function ϕ of two variables that depends on both. For $a = b = 4$ the largest monochromatic sublattice has 16 members when ϕ is exclusive or or its negation, 32 for disjunction and 48 for conjunction, against $f(4) + f(4) = 6$.*

The grid search of Lemma XVII.1, run with two blocks, computes these values. They show why subadditivity $f(a+b) \leq f(a) + f(b) + C$ cannot come from such products: a monochromatic sublattice of the product projects to sublattices of the factors, but the colour of a product point is a function of both factors, so the projections need not be monochromatic, and the product lattice of two chains is a grid with $(a+1)(b+1)$ members.

At $n = 20$ the grid colourings with two blocks of sizes 18 and 2, 57 grid points, were searched by simulated annealing on the number of pairs (colour, base point) from which a monochromatic sublattice of 12 members is reachable. Of 24 restarts, 16 end with exactly one such pair and none with zero; every colouring obtained from the best one by changing one or two grid points keeps a bad pair. Refining the best

colouring to three blocks of sizes (16, 2, 2) and (17, 1, 2) and annealing again, four restarts, also ends at one bad pair. None of the runs reaches zero, and none can: by the twenty-level window cited in Theorem XVII.2, every colouring of $2^{[20]}$ has a monochromatic sublattice of 12 members.

The colourings with the shortest monochromatic progressions known come from finite fields, and they were the natural candidates for an answer to Question 9.3. They fail.

Proposition XVII.7. *Let α be a primitive element of $\mathbb{F}_{2^p}$, $\beta \neq 0$, $n \leq 2^p - 2$, and colour $S \subseteq [n]$ by $\text{Tr}(\beta \alpha^{|S|})$. (i) If p is prime, every monochromatic sublattice isomorphic to a Boolean lattice has at most 2^{p-1} members. (ii) If $\alpha^e = 1 + \alpha^d$ and $\text{Tr}(\beta \alpha^{t+jd}) = 0$ for $0 \leq j \leq k$, with $t + k \max(d, e) \leq n$, there is a monochromatic sublattice with 3^k members. If α is a root of the trinomial $x^p + x + 1$ and $n = 2^p - 2$, a suitable β allows $d = 1$, $e = p$, $t = 0$ and $k = p - 2$, which gives $3^{p-2} = \frac{1}{9}(n+2)^{\log_2 3}$ members.*

Proof. A Boolean sublattice of rank k is $J(P)$ for an antichain $P = \{1, \dots, k\}$ with weights w_i and base t , and its members have colours $\text{Tr}(\beta x)$ for x in $X = \{\alpha^t \prod_{i \in I} \alpha^{w_i}\}$, whose span is $\alpha^t \prod_i \text{span}\{1, \alpha^{w_i}\}$; each factor is a plane, since $1 \leq w_i \leq 2^p - 2$. For prime p the field $\mathbb{F}_{2^p}$ has no subfield other than $\mathbb{F}_2$ and itself, so the theorem of Hou, Leung and Xiang [89] gives $\dim(UV) \geq \min(p, \dim U + \dim V - 1)$ for nonzero subspaces U, V , and the span has dimension at least $\min(p, k+1)$. Colour 0 puts it in a hyperplane, so $k \leq p - 2$. For colour 1 the differences $\alpha^t(1 + \alpha^{w_j}) \prod_{i \in I} \alpha^{w_i}$, $I \not\ni j$, have trace 0 and span a product of $k - 1$ planes times one element, so

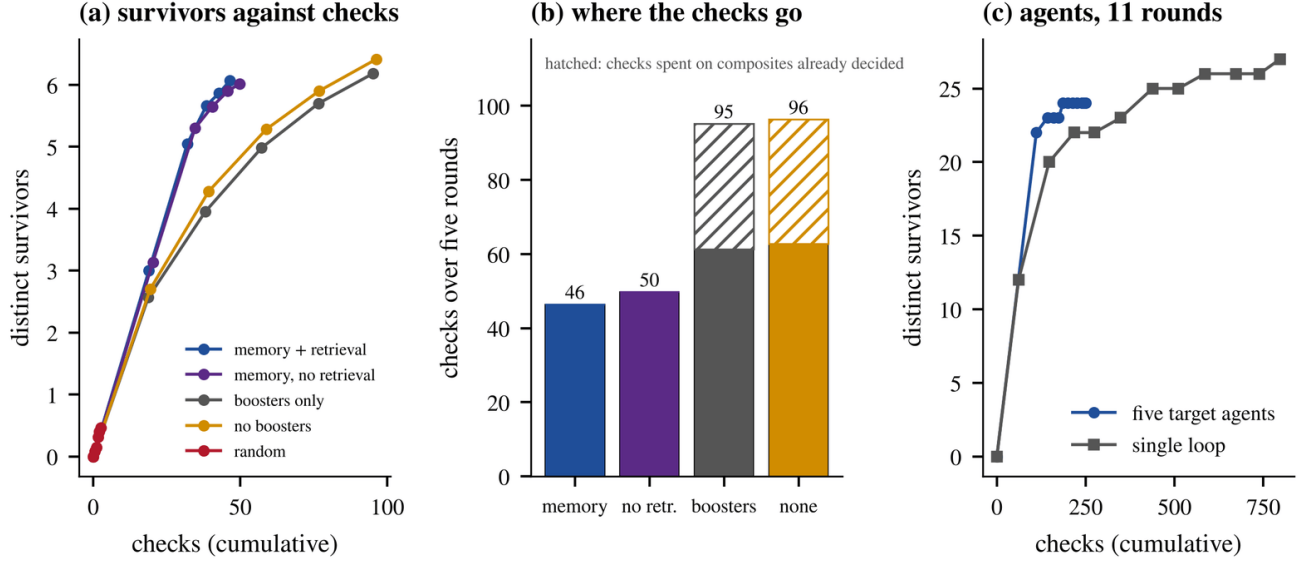


FIG. 22. Memory and agents. (a) Distinct survivors against cumulative checks over five rounds, mean of 60 seeds; each marker is one round. (b) Checks over the five rounds; the hatched part is spent on composites that an earlier round of the same seed had already decided. (c) The five target agents against the single-loop control in the continuous loop, with the local model among the proposers; each marker is one round.

$k \leq p - 1$. This is (i). For (ii) take k disjoint chains, each with ranks $0, d$ and e . Their down-sets have ranks $t + ad + be$ with $a + b \leq k$, and $\alpha^{t+ad+be} = \alpha^t (\alpha^d)^a (1 + \alpha^d)^b$ lies in the span of $\alpha^t, \alpha^{t+d}, \dots, \alpha^{t+kd}$, so every member has trace 0, and $t + k \max(d, e) \leq n$ keeps all ranks in range. The sublattice has 3^k members. For the trinomial, $\alpha^p = 1 + \alpha$, and the condition asks for a run of $k + 1$ zeros in the m -sequence $\text{Tr}(\beta \alpha^i)$; each period has a run of $p - 1$ zeros, choosing β moves it to $t = 0$, and $(p - 2)p \leq 2^p - 2$. $\square$

For the colourings defined by $x^5 + x^2 + 1$, $x^6 + x + 1$ and $x^7 + x + 1$ at $n = 2^p - 2$, the largest sublattices of the form (ii) have 27, 81 and 243 members, or 0.84, 1.27 and 1.90 times $n + 2$, with $(d, e) = (2, 5)$, $(1, 6)$ and $(1, 7)$. The linearity that keeps the progressions of both classes at length $p - 1$ turns every progression whose difference has a small partner e into a triangle of ranks, so an answer to Question 9.3 by level colourings needs classes that avoid both.

XVIII. CONSTRUCTION MODE

The crossing network combines what its library holds, and on every open step of the five benchmarks the library holds nothing that combines into the next object. Construction mode is the part of the engine that writes a new object down. It builds the object together with the argument for what the object forces, and it keeps the object only after the gate of Section XV G, which checks it again from scratch. This section tests it on a problem whose answer people found, so that a success can be recognised, and then asks whether towers of builders find answers with fewer checks.

A. Steinberg's problem from its rules

Steinberg conjectured in 1976 that every planar graph with no cycle of length 4 or 5 is 3-colourable [43, 163], and Erdős asked for the least k such that every planar graph with no cycle of length 4 to k is 3-colourable [21, 43]. Borodin, Glebov, Raspaud and Salavatipour proved that $k \leq 7$ [21]. Cohen-Addad, Hebdige, Král', Li and Salgado refuted Steinberg's conjecture in 2016, so $k \geq 6$ [43]; whether $k = 6$ or $k = 7$ is open. Their construction has three steps. A *gadget* G_1 on 15 vertices has three contacts on one face, at distances 3, 3 and 4, and no 3-colouring gives the three contacts one colour; three copies of G_1 and a triangle make a gadget G_2 on 42 vertices whose contacts are pairwise at distance 4; and four copies of G_2 in a planar frame give a graph with no cycle of length 4 or 5 that is not 3-colourable. The step that needs invention is the first: once a gadget of the right profile exists, the second and third steps are assemblies.

The engine was given the rules (planar with the contacts on one face, no cycle of length 4 or 5, the contact distances, and the requirement that no 3-colouring gives the contacts one colour) and nothing about the answer. Table XV lists the methods and what each found.

The builder. The builder starts from three contacts that share a colour and adds a vertex or an edge only when a forcing rule uses it: a vertex joined to two vertices of the shared colour receives one of the two other colours, a vertex joined to two vertices of different colours other than the shared one is forced, and two adjacent vertices forced alike close the argument. After each move it applies the cheap tests of the cascade (forbidden cycles, distances, planarity with the contacts on one face) and only then one exact satisfiability check; it stops when the check says the contacts can no longer share a

TABLE XV. Construction mode on Steinberg’s problem. Every gadget in the table passed the gate: planarity with the contacts on one face, every cycle length up to 7, the contact distances, and the forcing property by backtracking and by three satisfiability solvers. “Isomorphic” means isomorphic to the gadget of [43] with the contacts matched.

Target	Method	Outcome
gadget, distances 3, 3, 4	random planar frames filled with library gadgets	none in 2.45 million frames
the same	goal-directed builder, rules only	15 vertices, 23 edges, isomorphic to G_1 ; try 116, 11 s
the same	exists-forall search over two-page book embeddings	a 15-vertex, 23-edge gadget in 2.5 s (1,172 rounds)
gadget, distances all ≥ 4	builder, rules only	nothing: from contacts this far apart no rule applies
the same	builder with the rebuilt G_1 and a triangle gadget as lemmas	42 vertices, 72 edges, isomorphic to G_2 , on two seeds (tries 13 and 9); on a third seed 48 vertices with distances 5, 4, 5 (try 8, 3.5 s)
the whole graph	assembly of [43] on the rebuilt G_1	166 vertices and 300 edges, planar, cycles of length at most 6 only of lengths 3 and 6, not 3-colourable by CaDiCaL, Glucose and MiniSat
Erdős’s question: no cycle of length 4, 5 or 6	all of the above	nothing; open

colour, and then deletes vertices while the property holds. On the profile 3, 3, 4 it found on its 116th try, after 11 seconds, a gadget on 15 vertices and 23 edges that is isomorphic to G_1 with its contacts, and in the same run a gadget on 18 vertices with distances 3, 3, 5.

Why the rules stop at distance four. With no cycle of length 4, the neighbours of one contact that are not of the shared colour form a matching; a vertex joined to both ends of an edge of that matching closes a 4-cycle, and an edge between the neighbourhoods of two contacts puts them at distance 3. So from three contacts pairwise at distance at least 4 no forward rule derives anything, and the long distances of G_2 come from using G_1 as a lemma, which is a case split, not a forward step. The engine therefore keeps every gadget it finds as a lemma: a library gadget whose two contacts are forced to the shared colour forces its third contact to differ, and the builder may add such a copy as one move. With the rebuilt G_1 and the triangle gadget as lemmas it built a gadget isomorphic to G_2 on two seeds.

The exact search. The second method searches for a graph and a proof together: an exists-forall loop proposes a graph on n vertices from a satisfiability instance and refutes it with a colouring, which becomes a clause. Requiring the graph with one extra vertex joined to the contacts to have a two-page book embedding along a fixed spine makes every candidate planar with the contacts on one face, so the crossing rule is a fixed list of clauses and no Kuratowski cuts are needed. On Steinberg’s profile it found a 15-vertex gadget in 2.5 seconds, where the same loop with lazy Kuratowski cuts had not finished this size after 30 minutes. A “none” from this search covers only the graphs that are two-page along the spines tried.

Erdős’s question. The same assemblies reduce the question to one gadget: a gadget with no cycle of length 4, 5 or 6, planar with the contacts on one face, forcing the contacts not all equal, with distances 3, 3 and at least 5, would give through G_2 and the frame a planar graph with no cycle of length 4, 5 or 6 that is not 3-colourable, and so $k = 7$. The book search found no such gadget on 16 vertices along the spines tried, a search that treats each triangle gadget as one node found none

up to 13 nodes, and the builder with the lemmas of this case found none. A gadget built by hand from triangle gadgets has every property except planarity, on 24 vertices with distances 3, 3, 6; once three of its vertices are placed, one contact and the other two lie on different faces, and no later gadget can join the two sides. The wall for $k = 6$ is the planar arrangement of the forcing argument, not the forcing argument.

The calibration therefore passes in the sense that was fixed before it: the engine invented the gadget that people invented, from the rules, and assembled the counterexample from it. It is a rediscovery of a published construction, and on the question that remains open it found nothing.

B. Towers at equal cost

Theorem XIV.1 bounds what depth can gain; this bench measures what it gains on the problem of Section XVIII A. Every configuration ran on three targets, five seeds each, with the same budget of exact satisfiability checks, the unit in which the builder above is counted. The budgets were 10,000 checks for the profile 3, 3, 4 of G_1 from the rules alone, and 25,000 for the profile 4, 4, 4 of G_2 with the triangle gadget as the only lemma in the library. Erdős’s question, the profile 3, 3, 5 with no cycle of length 4 to 6, got 20,000, with the triangle gadget and a 15-vertex gadget whose contacts are at distances 3, 3, 3 and which has no cycle of length 4 to 6. A run reaches the goal when a gadget with the goal’s profile passes the gate of Section XV G, which is the same for every configuration and shares no code with the search.

Layer one of a tower is the builder. Layer two acts on layer one: it keeps a population of strategies of the builder (the priorities of the moves, the noise, how often a lemma is tried), runs each for a slice of the budget, keeps and mutates those that reach gadgets with fewer checks, and attacks an easier profile first, keeping the gadget it yields as a lemma for the goal. Layer three runs several copies of layer two with different settings and keeps the better half after each round. In the ladder each copy has its own library; in the mesh all layers

TABLE XVI. The tower bench. Reached: seeds, of five, whose run produced a gadget with the goal's profile that passed the gate. Checks: the exact checks spent before that gadget, median with the range in brackets, over the seeds that reached it. Time: mean seconds per run, on the machine of Appendix C. The time-limited crossing loop ran for 120 s on G_1 and 400 s on the other targets; the entry gives the checks it spent. On Erdős's profile no configuration found a gadget of any profile. At depth one the ladder and the mesh coincide.

Configuration	G_1 : distances 3, 3, 4			G_2 : distances all ≥ 4			Erdős: 3, 3, 5	
	reached	checks	time	reached	checks	time	reached	time
crossing loop	0/5	none	9	0/5	none	24	0/5	21
crossing loop, time only	0/5	none in $\sim 128,000$	120	0/5	none in $\sim 410,000$	400	0/5	400
depth 1	5/5	1,635 [486–2,602]	5	0/5	none	159	0/5	334
ladder, depth 2	5/5	766 [355–2,924]	5	3/5	12,419 [2,645–24,984]	216	0/5	302
mesh, depth 2	5/5	766 [355–2,924]	5	4/5	13,257 [2,451–22,934]	201	0/5	299
ladder, depth 3	5/5	759 [577–2,136]	4	3/5	14,958 [10,240–21,704]	251	0/5	274
mesh, depth 3	5/5	759 [577–2,136]	4	3/5	9,182 [7,022–22,022]	228	0/5	277

read and write one library, and Hebbian weights on the kinds of move, strengthened by the moves of every gadget found and decaying otherwise, bias the strategies of all layers. At depth two there is one library either way, so the two wirings differ only by the weights; at depth three the mesh also hands a lemma found by one copy of layer two to all the others. The crossing loop of Table XV, random planar frames filled with library gadgets, ran with the same budget, counted in the same checks, and again limited only by time, for longer than any tower run on the target. Table XVI and Figure 23 give the results; the whole bench took 5.6 hours of processor time, 1.9 hours on three cores.

Crossing. The crossing loop reached no goal and found no gadget of any profile, also when it was run for 120 or 400 seconds and spent 13 to 20 times the budget of the towers.

Depth on G_1 . Every tower reached G_1 on every seed, always as the gadget on 15 vertices and 23 edges. The median number of checks falls from 1,635 at depth one to 766 at depth two and 759 at depth three, but the spread between seeds is as large as the change: the geometric means are 1,176, 957 and 944, and seed by seed the second layer saved checks on three seeds, changed nothing on one and cost more on one. Five seeds do not separate the depths on this target.

Depth on G_2 . Here the second layer decides. With the triangle gadget as its only lemma the builder never reached the profile 4, 4, 4; as Section XVIII A shows, no forward rule applies to three contacts pairwise at distance at least 4. Towers of depth two reached it on seven of ten runs and towers of depth three on six of ten, with gadgets on 42, 45, 48 or 60 vertices. Every one of these runs first built a gadget with the profile 3, 3, 4 on 15 vertices and 23 edges, the curriculum of layer two, kept it as a lemma and built the goal from it; every run that did not reach the goal had failed to build that lemma within the budget, except one ladder run. The lemma took between 2,298 and 22,642 checks, in six of seven runs more than G_1 ever took as a target of its own; on this target the builder searches with larger size limits, 80 vertices and 400 steps against 40 and 200, and with the triangle gadget among its moves. The third layer did not help: it reached the goal on six runs against seven, with the same median of about 12,500 checks over the runs that reached it, and it reached the goal on one seed that depth two missed and missed two seeds that depth two reached at least once.

Ladder and mesh. On G_1 and on Erdős's profile the ladder and the mesh spent exactly the same checks on every seed. The design forces this: the weights of the mesh change only when a gadget is found, and on G_1 the goal is the first gadget, while on Erdős's profile no gadget was found. On G_2 the two wirings found the lemma at the same check on every seed and parted after it. Once the lemma existed, the mesh reached the goal on all seven runs and the ladder on six; the mesh needed a median of 308 checks from the lemma to the goal and the ladder 2,342, and the mesh needed fewer on six of the seven pairs (Figure 23(d)). At depth two, where the two wirings differ only by the weights, the difference comes from the weights learned from the moves of the lemma, either through the moves they favour or only because they send the two searches down different random paths; seven pairs are too few to tell these apart, and the direction is the one the mesh was built for.

Erdős's question. No configuration reached the profile 3, 3, 5 without cycles of length 4 to 6, and none found a gadget of any profile on the way, in 20,000 checks per run or in 400 seconds of crossing.

On this bench depth paid once, at the step where layer two adds what layer one lacks, a curriculum that turns an easier gadget into a lemma; the third layer did not pay for itself, and the mesh shortened the step after the lemma. This is consistent with Theorem XIV.1, which allows depth to pay only up to a finite depth, and the bench is too small to say more: three targets, five seeds and depths up to three. The towers rebuild the gadgets that people found and find nothing on the open question.

XIX. ONE PROBLEM ONLY: THE HODGE ROUND

The runs of Section XVII spread the network over five problems. This section points the whole of it at one, the Hodge conjecture in the form of [17]: the conjecture is equivalent to two statements about abelian varieties, (F2), that every Hodge class on an abelian variety outside the part $R(Y)$ reached from the known constructions is algebraic, and (F3'), that on every smooth projective variety the Hodge classes are those coming from abelian varieties, $\text{Hdg}^p = \text{Ab}^p$. The paper proves that the locus on which either statement holds is dense in every family

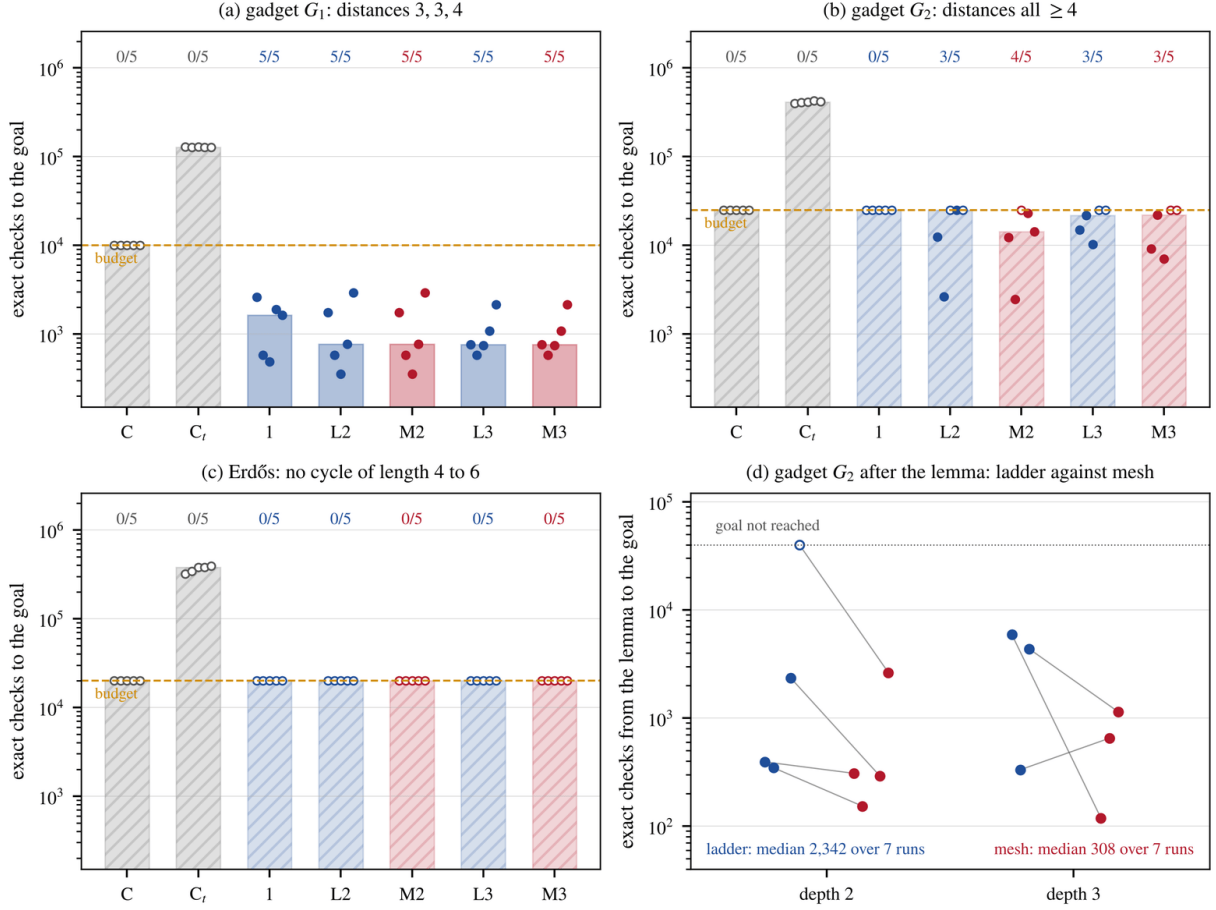


FIG. 23. The tower bench. (a)–(c) The exact checks each run spent before its first gadget with the goal’s profile, by configuration: C is the crossing loop with the budget of the towers, C_t the crossing loop limited only by time, 1 the builder alone, L the ladder and M the mesh at depths 2 and 3. Filled marks are runs that reached the goal, open marks runs that did not, drawn at the checks they spent; bars are medians over all five seeds, a run that did not reach the goal counting as its whole budget, and hatched bars are configurations in which some seed did not reach it; the fraction above each bar is the number of seeds that reached it; the dashed line is the budget. (d) On G_2 , the checks from the lemma, the first gadget with the profile 3, 3, 4, to the goal, for the ladder (blue) and the mesh (red) on the same seed and depth, joined by a line; the one ladder run that never reached the goal is drawn on the dotted line.

and is either everything or meagre, and it builds for each statement the object that would decide it next and proves at which step the construction stops. Those stopping points are the five target points of Table XVII. The question for the network is whether, with every result of [17], of the companion paper on Weil classes [30] and of the literature in its library, and with the fourteen external sources of Section XIX D, it produces one object that passes one target point. The measured quantity is the gap of Section XVIII E: for each target point the number of open steps between the best checked composite and the point, together with the counts of composites checked, rejected and stopped. Everything in this section was fixed before the runs except the numbers.

A. Library, agents and rounds

The library for this round has 76 typed constructions: the 19 results of [17] that bear on the target points; 9 of [30], namely Markman’s theorem for trivial discriminant, the Weil tori, the convolutions on E_0^6 and on E_0^8 , the quartic rank, the abelian covers, Schoen’s product, the Vandermonde intersections and the escape theorem; 34 results of the literature from Deligne to Markman’s recent work, among them one unrefereed claim, the family of preprints in [125] that asserts the Hodge conjecture for abelian varieties of CM type and for products of K3 surfaces, which enters as a hypothesis and never as a result; two counts computed for this round (Section XIX B); six moves that change the variety (blow-up at a CM point, degeneration to a CM fibre, Kuga–Satake passage, fibration, Hilbert scheme, isogeny cover); and six open conjectures from the OpenConjecture collection (Section XIX D),

TABLE XVII. The five target points of the Hodge round, from [17]. Each is the object that paper needs next and could not build; “open steps” is the depth of the deepest open step in the library’s route to the point, the gap of Fig. 24(c). None is reached.

target	object that passes it	where [17] stops	open steps
F2.lift	for one polarised abelian scheme $A \rightarrow S$ with a flat Hodge class ξ algebraic at a CM point c , a semiregular object at c with Chern character ξ_c ; then ξ is algebraic on all of S	the CM point exists in every family; the semiregular object at it does not	3
F2.E06	a convolution E of the pieces $O_{Z_\zeta} \otimes L^k[s], L^l[t]$ on $E_0^6, E_0 = \mathbf{C}/\mathbf{Z}[i]$, with $\text{ch}(E)$ of polarised shape and $\dim \text{Ext}^2(E, E) = 57$	the 35 pieces of its Proposition on separation give $\dim \text{Ext}^2 \geq 2605$; four configurations are left open	2
F2.link	one algebraic class $\xi \in \text{Hom}_{\text{Hdg}}(T_X(-1), T_Z)$ on $X \times X \times Z_K$ for a Mumford fourfold X whose cubic field F contains no imaginary quadratic field	$\text{Hom}(X, Z_K) = 0$; the class is a Hodge class on a CM sixfold and nothing produces it	1
F3.cubic	an algebraic correspondence of S_λ (Picard number 13, transcendental rank 9) acting on $T(S_\lambda)$ through the cubic field F , with (F, D) not $(\mathbf{Q}(\zeta_7)^+, D_\infty)$ or $(\mathbf{Q}(\zeta_9)^+, D_\infty)$	finite groups reach only $\mathbf{Q}(\zeta_7)^+$ and $\mathbf{Q}(\zeta_9)^+$; a self-map f with $f^2 = n$ in F forces $f \in \mathbf{Q}$	1
F3.sqrt	a family of K3 surfaces with real multiplication by $\mathbf{Q}(\sqrt{10})$ or $\mathbf{Q}(\sqrt{14})$ on which a chain of symplectic quotients or an isogeny of elliptic surfaces starts	the chain theorem gives $\sqrt{6}$ and the Fricke cases; no family starts the chain for $\sqrt{10}, \sqrt{14}$	1

which the refuter treats as hypotheses, so that a composite resting on one is recorded as conditional and stopped. One agent sits on each target point. In a *steady* round each agent runs six generations of ten pairings, with the refuter and the memory of Section XVII F; in a *burst* each agent runs ten generations of forty pairings and the round then checks three hundred random chains of three to six constructions, pruned by the same refuter. Every checked composite is recorded with its verdict and its reason, so that no later round checks it again.

Rounds 1 to 11 were steady and round 12 was a burst. Fig. 24 shows the run. The memory records 78 checked composites in the first round, 29 in the second, and between 7 and 16 in each of the next eight, as the combinations of the library are used up; round 11, run after the two counts and the moves had entered the library, checked 35. In all 230 composites were checked in the eleven steady rounds, 22 survived, and every survivor is a known result: the reduction to (F2) and (F3’), the propagation theorem, the density of CM points, Markman’s theorem and the cases it covers. No composite reaches a target point, and the gap of every target point is the same after round 11 as after round 1: three open steps for F2.lift, two for F2.E06 and one for each of the other three.

Round 12 had a time limit of 7000 seconds. It committed 29 checked composites to the memory in its first 98 minutes and was stopped at the limit before it wrote its summary and before the long chains began. None of the 29 survived. Twenty-four were stopped because they rest on an open conjecture used as a hypothesis, the derived form of the Hodge conjecture or a statement on generic Hodge classes from the OpenConjecture entries; three stopped at the count on E_0^6 , one at the semiregular object of F2.lift, and one was rejected. Most of the time of the round went into the channel, which brought 58,872 records of the memory to the proposers, against 11,558 in round 11 and between 1,546 and 4,108 in each earlier round. The burst therefore multiplies the cost of a round and moves the search toward composites that assume what they need; it does not change the gap. A wider search of the same library does not move a flat line at the depth of the first open step, because that step is not a combination of what the library holds.

B. Counting on E_0^6

The target F2.E06 is the one that a finite computation can decide, and [17] leaves four configurations open after its 35 pieces: (a) four or more line bundles, (b) one sheaf at two shifts, (c) one threefold at several twists, (d) two threefolds at the same shift meeting along an abelian surface. For a convolution E of pieces F_a the complex $\text{Ext}^\bullet(E, E)$ is filtered by the blocks $\text{Ext}^k(F_a, F_b)$ with total degree $k + s_b - s_a$, and $\dim \text{Ext}^2(E, E)$ is at least the number of degree-two blocks that no differential reaches or leaves: a block $\text{Ext}^2(F_a, F_b)$ is *isolated* when no degree-one block maps to it along a chain of pieces and it maps to no degree-three block, and a bound that counts only isolated blocks is conservative, since a non-exact block can only add edges. The Ext groups of the pieces are the tables of [17] for $K = \mathbf{Q}(i)$, with $\chi_{Z_\zeta} = 8$ and the groups between threefolds meeting along an abelian variety of dimension d in $\prod_j |\zeta_j - \zeta'_j|^2$ components. We computed, for every polarised configuration of each family in a window of twists 1 to 4 and shifts -3 to 3 , the sums C^1, C^2, C^3 of the block dimensions by total degree, the Euler characteristic $C^2 - C^1 - C^3$, the isolated blocks, and a restricted Euler bound that removes the blocks that cannot interact. Fig. 25 shows the pieces of the least-bound configuration of family (a) and its block matrices; Fig. 26 shows the bounds over the families.

The results are these. The 35 pieces of [17] with the three line bundles at shifts $(0, 0, 0)$, $(2, 3, 4)$ or $(-3, -2, -1)$ have at least 7213, 2605 and 2605 isolated degree-two blocks, so $\dim \text{Ext}^2 \geq 2605 > 57$, in agreement with the bound 2560 of [30] for its line-bundle model. Family (d), two threefolds at the same shift meeting along a surface, has six polarised configurations in the window and the least bound is 5325: it is excluded. Family (b) has no polarised configuration in the window. Family (a) has 2401 polarised configurations, and for the four line bundles $L^1[-2], L^2[-1], L^3[0], L^4[1]$ the isolated blocks are 0 and the Euler characteristic is -17665 ; family (c) has 432, and for one threefold at the twists 1 and 2 at the shifts $-1, 0, 1, 2$ the isolated blocks are again 0. Counting decides nothing for (a) and (c). The closed chains of pieces

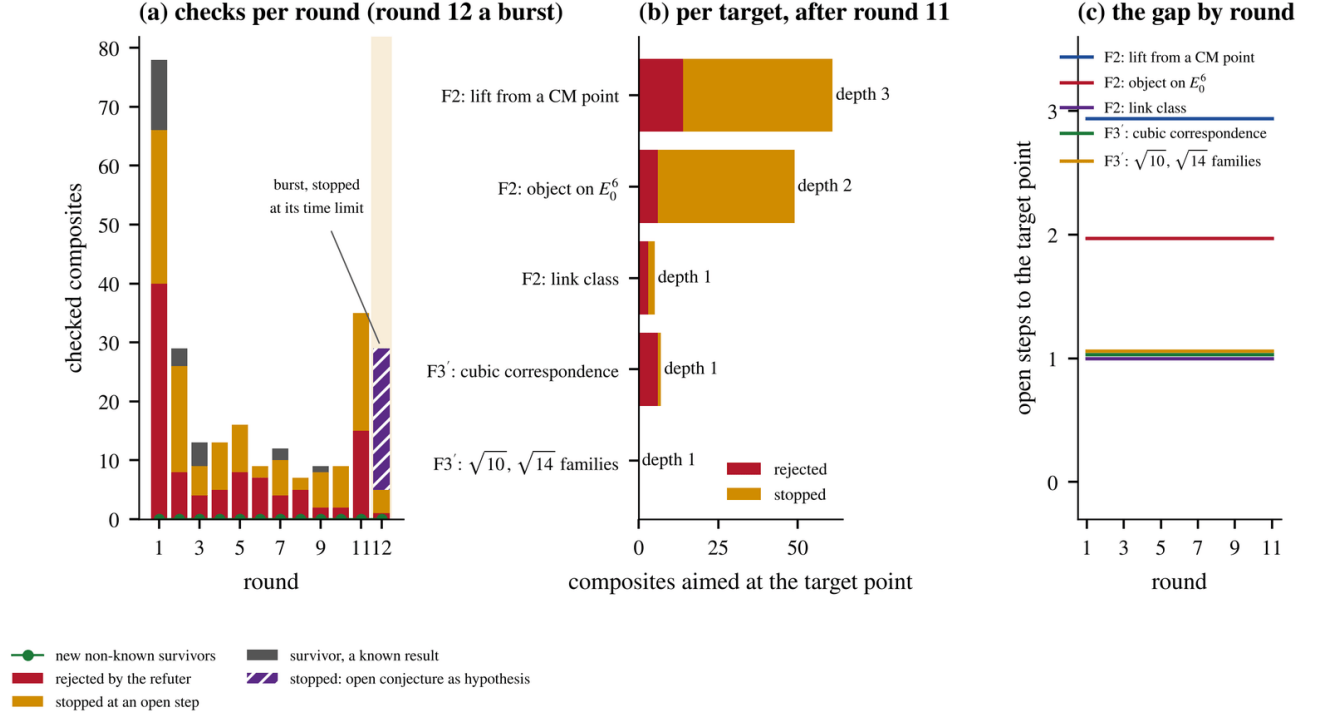


FIG. 24. The Hodge-only run. (a) Checked composites per round, from the memory: rejected by the refuter, stopped at an open step, and survivors, all of them known results; the green line, the survivors that are more than a known result, is zero in every round. Round 12 (shaded) is the burst, stopped at its time limit; its hatched part rests on an open conjecture used as a hypothesis. (b) The composites aimed at each target point over rounds 1 to 11, with the depth of the gap after round 11. (c) The gap of each target point by round: the number of open steps between the best checked composite and the point. It does not move.

along which a term of d_E can return to a diagonal block decide which product acts: for (a) there is no closed chain with every step of total degree one up to length eight, and the one chain of [17], through two threefolds and four line bundles with Ext degrees 0, 3, 3, 0, 0, 0, has length six, so the deciding product is the sixfold Massey product m_6 ; for (c) the chains have lengths two to five. The two open configurations are therefore a rank computation, not a search, and that computation is the step the network cannot take by combination.

C. Candidates

Exclusion alone leaves a target point where it was, so the agents also construct. Table XVIII lists the seven explicit candidates written down in this round, each with its invariants, what it achieves, where it falls short and what decides it; Fig. 27 places them under their target points. Three die here: the two threefolds at one shift (C3), by the count above; the link class as a construction (C5), since a Hodge class on the CM sixfold $X \times X \times Z_K$ is exactly the statement to prove and $\text{Hom}(X, Z_K) = 0$ leaves no map to transport it, so C5 survives only as the reduction it already is in [17]; and the correspondences of S_λ from Mukai vectors, Buskin's isometries and self-maps (C6), since a self-map f of the transcendental lattice with $f^2 = n$ in a cubic field is rational, and the moduli-

of-sheaves correspondences act through the Hodge isometries that [17] classifies. Four are open and each names the computation that decides it: C1 and the Weil family through it (C4) by the rank of m_6 on the chain of length six, C2 by the ranks of m_3 to m_5 along its chains, in the theta-function model of [30]; and C7, a chain for $\sqrt{10}$ through a symplectic involution and a symplectic automorphism of order five, multiplier $2/5 \equiv 10$ modulo squares, by one lattice question, whether a K3 surface with a symplectic automorphism of order five and transcendental rank six carries real multiplication by $\sqrt{10}$. Kummer surfaces of abelian surfaces with real multiplication are excluded from C7 because their Asai representation is irreducible. None of the seven is a proof of anything, and the table says so for each.

D. Sources

Fourteen sources were read during the round, and each is recorded in the memory with what was taken and what was left: a vault memory with embeddings and a knowledge graph, from which the community structure of the construction graph is kept and the server is not; a security reasoning stack with nothing for Hodge; an orchestration policy whose rule we adopt, that a problem is closed only after a kernel-checked Lean endpoint with an audited statement and the three stan-

TABLE XVIII. The explicit candidates of the Hodge round. The status names the computation or the result that settles the candidate; “dies” means excluded with the reason given, “open” means that a stated computation decides it and has not been done. No candidate reaches its target point.

	target	object	invariants and where it falls short	status
C1	F2.E06	the 32 threefolds $O_{Z_\zeta}[s]$ and $L^1[-2], L^2[-1], L^3[0], L^4[1]$ on E_0^6	ch polarised with $N = 32$, Hankel rank 3, $r = 57$; $C^1, C^2, C^3 = 2523, 20316, 35458$; isolated 0; one closed chain, length six	open: rank of m_6
C2	F2.E06	one threefold Z_{111} at the twists 1, 1, 2, 2 and shifts $-1, 0, 1, 2$ with the 35 pieces	polarised, isolated 0, Euler -11423 ; chains of lengths 2 to 5; the identity between the two copies contracts the pair, the zero map leaves the chains	open: ranks of m_3 to m_5
C3	F2.E06	two threefolds meeting along an abelian surface at the same shift, with a cancelling pair	six polarised configurations; isolated $\geq 5325 > 57$	dies
C4	F2.lift	the Weil family of dimension 9 through $A_0 = E_0^6$ with the product object of [17] at the CM point	the object at A_0 is C1; at trivial discriminant the result would be Markman’s theorem again, and the non-trivial discriminant needs C1 first	open: C1, then one discriminant
C5	F2.link	the class ξ as a Hodge class on the CM sixfold $X \times X \times Z_K$ at a CM point	it is the Hodge conjecture for that sixfold; $\text{Hom}(X, Z_K) = 0$ gives no transport	dies as a construction
C6	F3.cubic	correspondences of S_λ from Mukai vectors, Buskin isometries and self-maps	$f^2 = n$ in a cubic field forces $f \in \mathbf{Q}$; the isometries are those classified in [17]	dies
C7	F3.sqrt	$S \rightarrow Y \leftarrow S'$ through a symplectic involution and a symplectic automorphism of order 5, multiplier $2/5 \equiv 10$	coinvariant rank of order five is 16, transcendental rank ≤ 6 ; Kummer surfaces of RM abelian surfaces excluded	open: one lattice question

dard axioms, never `native_decide`, with evidence graded as lead, candidate, reproduced, formally verified and accepted; a Lean coding agent whose split of a theorem store from an axiom store the engine’s Lean check follows; a hypothesis tournament whose Elo ranking orders the survivors of a burst; the output of a research agent on equivariant traces of the Hodge bundle over the moduli of curves, which enters as one construction; vault hygiene rules; four repositories with no mathematics; and the `OpenConjecture` collection of 3531 open conjectures from the arXiv, in which a search finds 55 on Hodge classes, abelian varieties, K3 surfaces and motives, six of which enter the library as conditional nodes. The memory shows that no source contains a construction for any target point; the ones that help change how the network keeps and ranks what it finds, not what it can find.

E. What this round shows

Nothing closes. (F2) and (F3’) are where [17] left them, the gap of every target point is unchanged after the burst, and the only decisions of the round are negative: the configuration (d) and every configuration of the 35 pieces on E_0^6 are excluded by counting, two candidates die by arguments of [17] applied to them, and the sources add method and no construction. What the round isolates is positive in a narrower sense: two of the four open configurations of [17] are reduced to the ranks of finitely many Massey products in an explicit model, and the companion paper [30] has already carried out such a computation for the fourfold products at distance two or more and for E_0^8 within six shifts, with the answer that they fail. By the measured gap the search has reached its asymptote, and the next step is not a wider search but the rank of m_6 on the chain of candidate C1, which decides C1 and C4 either way. That rank is a finite computation in the theta-function model of [30]; it

is not carried out here.

XX. WHAT IS PROVED AND WHAT IS OPEN

A. Proved

Inside the model of Section IV, the paper determines the least cost of soundness. For finitely many candidates and exponential checks it is the entropy law, exact up to an additive term that is sharp (Theorem A) and exact without remainder when the soundness is the reciprocal of an integer and the budget a power of it, where the optimal schedule is a Huffman code (Theorem B). For checks of polynomial order the least cost is given in closed form by a Rényi entropy (Theorem C), and for every cost function an infinite sequence of candidates admits a sound schedule of finite cost exactly when one series built from the concave conjugate of the cost converges (Theorem VII.3). The same price, rescaled by the error rate of the generator, governs chains of improvements (Theorem F), and checking only the outcome of a chain of n steps is exponentially more expensive than checking every step (Theorem VIII.2). An infinite chain reaches a singularity soundly exactly when the costs of its checks have finite entropy in addition to a finite sum, which fails for a nonempty range of cost laws (Theorem D, Corollary IX.2); a branching takeoff with a fixed false acceptance rate keeps infinitely many errors on survival, and a sound one needs a number of runs that grows linearly with the generation (Theorem E). In the Osgood model of growth every singular takeoff can be made sound with exponential checks, at a price of order $1/(\alpha - 1)^2$ near the critical exponent, while polynomial checks move the threshold of a sound takeoff from $\alpha = 1$ to $\alpha = 1 + s$ (Proposition F.1).

None of this needs foresight or a single agent. A rule that sees only the past costs is sound for every future and pays over

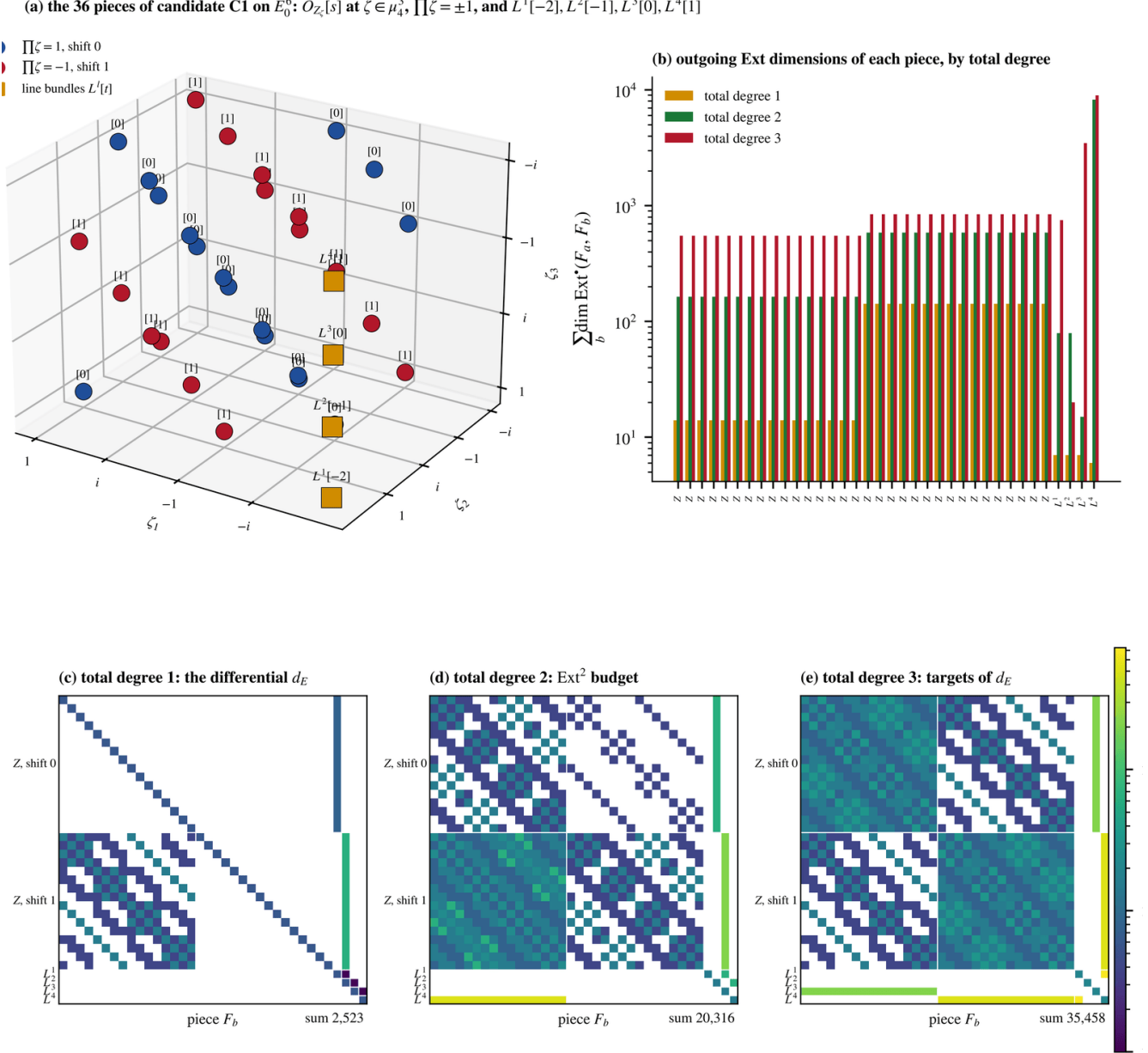


FIG. 25. The pieces of candidate C1 on E_0^6 and their Ext blocks. (a) The 32 threefold pieces $O_{Z_\zeta}[s]$ at the characters $\zeta \in \mu_4^3$ with $\prod \zeta = \pm 1$, the sign fixing the shift, and the four line bundles $L^1[-2], L^2[-1], L^3[0], L^4[1]$ of the least-bound configuration of family (a). (b) For each piece, the sum of the dimensions of the Ext groups to all pieces, by total degree. (c)–(e) The block matrices by total degree: the degree-one blocks carry the differential d_E , the degree-two blocks are the budget from which $\text{Ext}^2(E, E)$ is cut, and the degree-three blocks are the targets of d_E on degree two; the white lines separate the two parity classes of threefolds and the line bundles. Every degree-two block of this configuration is reached or leaves, so the counting bound is 0 and the rank of the sixfold product decides.

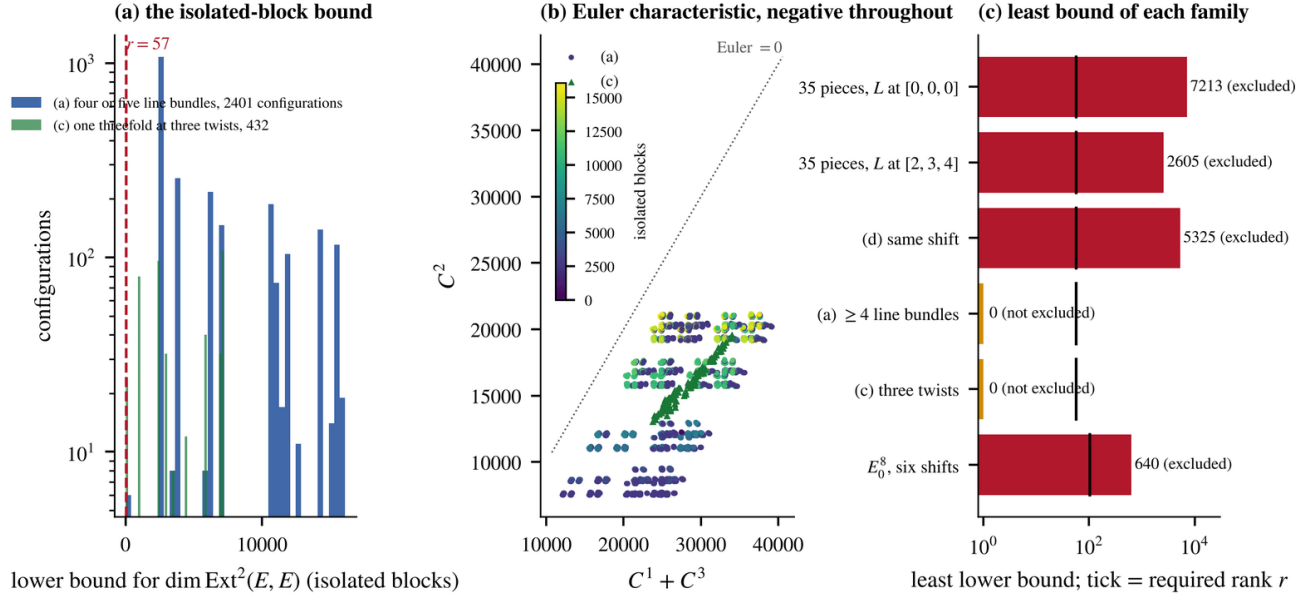


FIG. 26. Counting on E_0^6 . (a) The isolated-block lower bound for $\dim \text{Ext}^2(E, E)$ over the 2401 polarised configurations of family (a) and the 432 of family (c), against the required rank $r = 57$. (b) C^2 against $C^1 + C^3$ for the same configurations: the Euler characteristic is negative everywhere, so no configuration is excluded by it. (c) The least bound of each family against the rank it must meet: the 35 pieces and family (d) are excluded, families (a) and (c) are not; the row for E_0^8 is the bound of [30] within six shifts.

the offline optimum an excess of order $\frac{A}{L} \log \log \frac{A}{a_1}$, which no online rule avoids (Theorem G), so sound singularities are reached online whenever they are reached at all (Corollary X.1). Splitting the work among subagents is free when the budget is split in proportion to cost, and costs exactly a Kullback–Leibler divergence otherwise (Theorem H); cheap checks before one exact check reduce the cost of the exact check to the correct candidates, up to a logarithmic premium (Theorem XI.1). Depth has a price of the same kind: n verified layers that share one budget cost at least $n \log(n/\theta)$ nats of soundness, each added layer at least $\log((n+1)/\theta)$ more, and past a finite depth every tower costs more than any fixed amount (Theorem XIV.1).

Outside the model, two results come from the experiments. Theorem XVII.4 proves a lattice mass gap $m \geq -\log(3\beta_W)$ for four-dimensional $\text{SU}(2)$ on $\beta_W < \frac{1}{3}$ by a gauge fixing with one fixed link per plaquette, and Proposition XVII.5 shows that no forest gauge fixing goes further; the theorem is proved here and has not been refereed. Lemma XVII.1 reduces monochromatic sublattices of a level colouring to weighted posets, and with it Theorem XVII.2 determines $f(n)$ for $n \leq 20$.

The measurements of Section XVII are these: typing the constructions is what lets the network reach a step that random pairing does not; Hebbian boosters do not improve on the unboosted network; memory halves the checks needed to reach the plateau and does not raise it; agents reach the plateau sooner and no further; and no transfer of a pattern in the matrix of Section XVII H closes a blocking step. Construction mode rebuilds from the rules of Steinberg’s problem the two gadgets of the counterexample of [43] and the counterexam-

ple itself, where random combination of library pieces finds nothing, and finds nothing on Erdős’s question for cycles of length 4 to 6 (Section XVIII A). On three targets at equal budgets of exact checks, depth paid once: a second layer whose curriculum turns an easier gadget into a lemma reaches a gadget that the builder alone does not, a third layer adds nothing, and sharing one library and one set of weights shortens the step after the lemma (Section XVIII B). Pointed at the Hodge conjecture alone, the network leaves the gap of every target point where it was after twelve rounds, including a burst that multiplied the cost of a round; counting excludes two of the configurations on E_0^6 left open in [17] and reduces two others to explicit Massey products (Section XIX).

The finite inequalities behind these statements are proved in Lean 4 (Appendix B). The probabilistic limit theorems, Wald’s identity, the Kesten–Stigum theorem, Lévy’s conditional Borel–Cantelli lemma and the extinction theorem of Galton–Watson processes, are classical and are proved or cited in Appendix A; the asymptotics of ζ near 1 are proved in the text.

B. Open

Five questions are left open, and each is a precise mathematical question.

Adaptive protocols. Remark V.10 shows that a protocol which looks at the outcomes of the checks can avoid the entropy term by halting at the first rejection, at the price of losing the correct work after it. Let a protocol be *conservative* if, in every world, it keeps every correct candidate with probability

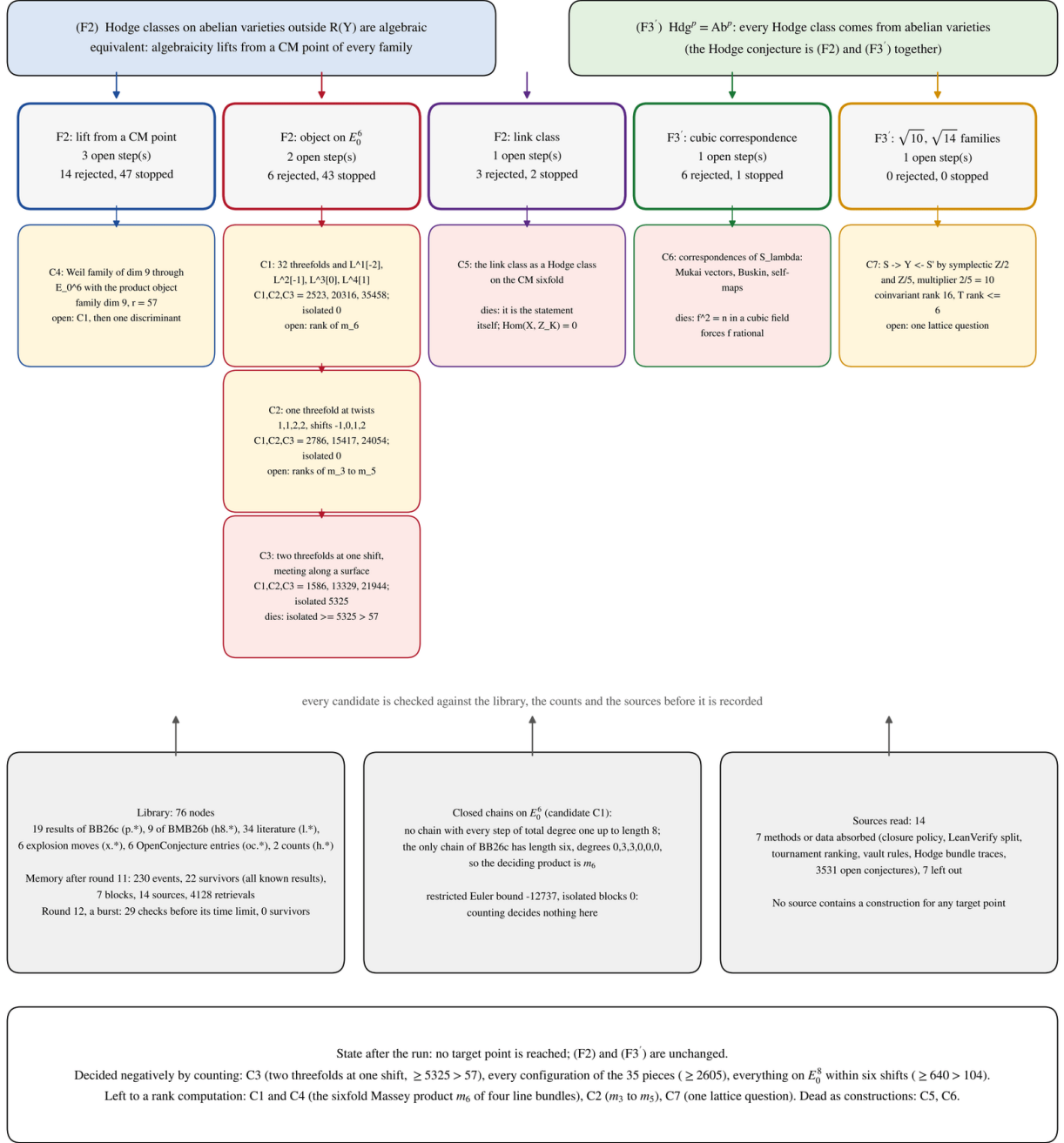


FIG. 27. The map of the Hodge round. The two statements (F2) and (F3') at the top, the five target points with their gap and their counts under them, the seven candidates under their targets (yellow open, red dead), and below them the three things every candidate is checked against before it is recorded: the library and the memory, the counting and chain results on E_0^6 , and the fourteen sources. The bottom line is the state after the run.

one. Is the least worst-case expected cost of a conservative θ -sound protocol equal to $V^*(\theta)$ up to an additive $O(A)$? An adaptive protocol may let the number of runs of later checks depend on the results of earlier ones; since a correct candidate never fails, a conservative protocol learns nothing from the runs of a candidate that passes, and the question is whether the information it gains from rejections can be converted into a saving that is not $O(A)$.

Two-sided checks. For checks that make errors on both sides Proposition V.11 gives the entropy law with the rate $D = \text{KL}(\text{Ber}(\frac{1}{2}) \parallel \text{Ber}(q))$ in the upper bound and $\log(1/q)$ in the lower bound. The exact rate of a sequential test with two error probabilities is Wald's [180], and the question is whether the optimal schedule for many two-sided checks attains the entropy law with the Chernoff rate of the pair of hypotheses, in both directions.

Correlated runs. The model assumes that the runs of a check are independent. A real verifier, a learned reward model or a test suite run on the same machine, has correlated errors, and repeated runs of a deterministic check gain nothing. The entropy law holds for every family of checks whose false acceptance probability after r runs is at most ε^r , whatever the mechanism; the open question is which checks used in practice have this property, with which ε , and what replaces the law for those that do not.

Monochromatic sublattices. Is $f(n) = O(n)$? For every $n \leq 20$ some level colouring is optimal (Theorem XVII.2); if level colourings are optimal for every n , Lemma XVII.1 reduces the question to words in two letters.

The online constant. The upper and lower bounds of Theorem G differ by a factor of four in the term $\frac{AN}{L} \log \log \frac{AN}{a_1}$. The exact constant is the least κ such that a θ -sound online rule has excess $(\kappa + o(1)) \frac{AN}{L} \log \log \frac{AN}{a_1}$ on every sequence; the bounds give $\frac{1}{2} \leq \kappa \leq 2$.

C. What is not claimed

The paper does not claim that any existing system is sound or that an intelligence explosion will or will not occur. It does not claim progress on the five problems of Section XVI beyond the two results above: no pass line is crossed, the Yang–Mills step lies below the crossover where the open problem begins, and the values of $f(n)$ do not decide whether $f(n) = O(n)$. The Steinberg gadgets are a rediscovery of a published construction, and the exclusions on E_0^6 leave (F2) and (F3') open. Passing all five lines is a definition we use to score a proposer, not a property any system has shown. The theorems are statements about a model with independent runs, perfect completeness and schedules fixed in advance, and their value for a real system depends on how well it satisfies them. What is claimed is that inside the model the cost of soundness is computable, that it is an entropy, and that it divides the singular processes into those that can be made sound and those that cannot; and that on five open problems a typed crossing network with memory reaches what its libraries can reach at half the cost, and stops where the libraries stop, while construction mode writes down objects that no combination of

the library reaches and has so far written down only known ones.

Appendix A: Classical facts used in the proofs

This appendix proves, or states with a precise reference, the classical results that the main text uses. Nothing here is new; the proofs are given so that the paper can be read without the references.

1. Kraft's inequality and McMillan's theorem

Fix an integer $D \geq 2$. A word of length ℓ over the alphabet $\{0, \dots, D-1\}$ is identified with the interval

$$J(w) = \left[\sum_{j=1}^{\ell} w_j D^{-j}, \sum_{j=1}^{\ell} w_j D^{-j} + D^{-\ell} \right) \subset [0, 1),$$

the set of reals in $[0, 1)$ whose D -ary expansion, taken without a tail of digits $D-1$, begins with w . Two intervals $J(w)$ and $J(w')$ meet if and only if one of the words is a prefix of the other: if w is a prefix of w' then $J(w') \subset J(w)$, and otherwise the two expansions differ at some position not beyond the shorter length, so no real begins with both.

Proposition A.1 (Kraft). *Lengths $\ell_1, \dots, \ell_n \in \mathbb{N}$ are those of a D -ary prefix code if and only if $\sum_i D^{-\ell_i} \leq 1$.*

Proof. If the words form a prefix code, the intervals $J(w_i)$ are disjoint subsets of $[0, 1)$ of lengths $D^{-\ell_i}$, so their lengths add up to at most 1. Conversely, order the lengths so that $\ell_1 \leq \dots \leq \ell_n$ and put $c_i = \sum_{j < i} D^{-\ell_j}$. Each $D^{-\ell_j}$ with $j < i$ is a multiple of $D^{-\ell_i}$, so $c_i D^{\ell_i}$ is an integer, and $c_i + D^{-\ell_i} \leq \sum_j D^{-\ell_j} \leq 1$, so $0 \leq c_i D^{\ell_i} \leq D^{\ell_i} - 1$. Let w_i be the expansion of this integer in base D with exactly ℓ_i digits. Then $J(w_i) = [c_i, c_{i+1})$, these intervals are disjoint, and the words form a prefix code. $\square$

Proposition A.2 (McMillan). *If a code with lengths $\ell_1, \dots, \ell_n$ is uniquely decodable, that is, distinct finite sequences of codewords have distinct concatenations, then $\sum_i D^{-\ell_i} \leq 1$.*

Proof. Let $M = \max_i \ell_i$ and $k \geq 1$. Expanding the k -th power,

$$\left(\sum_i D^{-\ell_i} \right)^k = \sum_{i_1, \dots, i_k} D^{-(\ell_{i_1} + \dots + \ell_{i_k})} = \sum_{m=k}^{kM} N_m D^{-m},$$

where N_m is the number of k -tuples of codewords whose concatenation has length m . By unique decodability these concatenations are distinct words of length m , so $N_m \leq D^m$ and the sum is at most kM . Hence $\sum_i D^{-\ell_i} \leq (kM)^{1/k}$ for every k , and $(kM)^{1/k} \rightarrow 1$. $\square$

2. Huffman's algorithm

Let $p_1 \geq \dots \geq p_n \geq 0$ with $\sum p_i = 1$, and let $L_D(p)$ be the least value of $\sum_i p_i \ell_i$ over $\ell \in \mathbb{N}^n$ with $\sum_i D^{-\ell_i} \leq 1$. The minimum exists: let c_0 be the cost of the code in which every word has length $\lceil \log_D n \rceil$; a vector of cost at most c_0 has $\ell_i \leq c_0/p_i$ for every i with $p_i > 0$, and the cost does not depend on the other lengths, so it takes only finitely many values at most c_0 . Huffman's algorithm [93] is the following recursion. If $n = 1$ the code is the empty word. Otherwise add zero weights until $n \equiv 1 \pmod{D-1}$, merge the D smallest weights into one symbol whose weight is their sum, solve the smaller problem, and give the D merged symbols the codeword of their merger followed by each of the D letters.

Proposition A.3 (Huffman). *The algorithm returns a prefix code of expected length $L_D(p)$.*

Proof. Step 1: padding is free. Adding a symbol of weight 0 cannot lower the optimum, since removing a codeword from a code keeps Kraft's inequality and the cost. It does not raise it either, by the following argument. Let ℓ be optimal for p with $n \not\equiv 1 \pmod{D-1}$, and let $M = \max \ell_i$. Since $D^{M-\ell_i} \equiv 1 \pmod{D-1}$, the integer $\sum_i D^{M-\ell_i}$ is congruent to n , and D^M to 1, modulo $D-1$; so $\sum_i D^{-\ell_i} < 1$, the difference being a positive multiple of D^{-M} . Zero weights at lengths $M+1, \dots, M+k$ add $\sum_{j=1}^k D^{-M-j} < D^{-M}$ to the Kraft sum and nothing to the cost. So we may assume $n \equiv 1 \pmod{D-1}$.

Step 2: an optimal code with equality. Among the optimal ℓ choose one with $\sum_i \ell_i$ least, and let $M = \max \ell_i$; if $M = 0$ then $n = 1$. By the congruence of Step 1, now with $n \equiv 1$, the deficit $\Delta = 1 - \sum_i D^{-\ell_i}$ satisfies $\Delta D^M \equiv 0 \pmod{D-1}$. If $\Delta > 0$, then $\Delta \geq (D-1)D^{-M}$, and shortening one word of length M by one letter adds $(D-1)D^{-M}$ to the Kraft sum, keeps it at most 1, does not raise the cost and lowers $\sum \ell_i$, a contradiction. So $\sum_i D^{-\ell_i} = 1$.

Step 3: the smallest weights are deepest. If $p_i > p_j$ and $\ell_i > \ell_j$, exchanging the two lengths lowers the cost by $(p_i - p_j)(\ell_i - \ell_j) > 0$; so in an optimal code $p_i > p_j$ implies $\ell_i \leq \ell_j$, and among equal weights the lengths may be permuted freely. Multiplying the equality of Step 2 by D^M , every word shorter than M contributes a multiple of D , so the number of words of length M is a positive multiple of D , hence at least D . By the exchange we may assume that the D smallest weights $p_{n-D+1}, \dots, p_n$ have length M .

Step 4: the recursion. Let p' be the vector of $n - D + 1 \equiv 1 \pmod{D-1}$ weights obtained by merging the D smallest into $\sigma = p_{n-D+1} + \dots + p_n$. The lengths of Step 3, with the merged symbol at length $M-1$, satisfy Kraft's inequality for p' , since $D \cdot D^{-M} = D^{-(M-1)}$, and have cost $\sum_i p_i \ell_i - \sigma$; so $L_D(p') \leq L_D(p) - \sigma$. Conversely, a code for p' in which the merged symbol has length ℓ' gives a code for p by giving the D merged symbols length $\ell' + 1$, with the same Kraft sum and cost increased by σ ; so $L_D(p) \leq L_D(p') + \sigma$. Hence $L_D(p) = L_D(p') + \sigma$, which is the recursion of the algorithm, and induction on n concludes. $\square$

3. Wald's identity and the geometric number of retries

Let $c_1, c_2, \dots$ be independent and identically distributed costs of successive candidates, with finite mean, and let K be the index of the first candidate kept, where whether candidate j is kept depends only on candidate j , with probability $s > 0$. Then K is geometric, $\mathbb{E}K = 1/s$, and $\mathbb{E}\sum_{j \leq K} c_j = \mathbb{E}K \cdot \mathbb{E}c_1$ [180]. Indeed $\{K \geq j\}$ is determined by the candidates $1, \dots, j-1$ and is therefore independent of c_j , so by monotone convergence $\mathbb{E}\sum_{j \leq K} c_j = \sum_j \mathbb{E}[c_j] \mathbb{P}(K \geq j) = \mathbb{E}c_1 \sum_j \mathbb{P}(K \geq j) = \mathbb{E}c_1 \mathbb{E}K$. This is the identity used in Lemma VIII.1.

4. Products and the Borel–Cantelli lemmas

Let $0 \leq \pi_n < 1$. If $\sum \pi_n < \infty$ then $\prod_n (1 - \pi_n) > 0$: for n large $\pi_n \leq \frac{1}{2}$, and $\log(1-x) \geq -2x$ on $[0, \frac{1}{2}]$, so the logarithm of the tail of the product is at least $-2\sum_{n \geq N} \pi_n > -\infty$, and the first factors are positive. If $\sum \pi_n = \infty$ then $\prod_{n \geq N} (1 - \pi_n) \leq \exp(-\sum_{n \geq N} \pi_n) = 0$ for every N . For independent events E_n with $\mathbb{P}(E_n) = \pi_n$ the second statement is the second Borel–Cantelli lemma, since $\mathbb{P}(\text{no } E_n \text{ with } n \geq N) = \prod_{n \geq N} (1 - \pi_n) = 0$ for every N ; the first Borel–Cantelli lemma, that $\sum \mathbb{P}(E_n) < \infty$ implies that only finitely many E_n occur, needs no independence, because $\mathbb{E}\sum \mathbf{1}_{E_n} = \sum \mathbb{P}(E_n) < \infty$ [32, 64].

5. Lévy's conditional Borel–Cantelli lemma

Let $(\mathcal{F}_k)$ be a filtration, $B_k \in \mathcal{F}_k$ and $p_k = \mathbb{P}(B_k \mid \mathcal{F}_{k-1})$. Then, up to a null set, $\{B_k \text{ infinitely often}\} = \{\sum_k p_k = \infty\}$ [64, Section 4.3].

Proof. $M_n = \sum_{k \leq n} (\mathbf{1}_{B_k} - p_k)$ is a martingale with increments bounded by 1. For such a martingale, almost surely either M_n converges to a finite limit or $\limsup M_n = +\infty$ and $\liminf M_n = -\infty$ [64, Section 4.3]. Write $\sum_{k \leq n} \mathbf{1}_{B_k} = M_n + \sum_{k \leq n} p_k$; the left side is nonnegative and nondecreasing. On $\{\sum p_k < \infty\}$, $\liminf M_n = -\infty$ would make the left side negative, so M_n converges and $\sum \mathbf{1}_{B_k} < \infty$. On $\{\sum p_k = \infty\}$, if $\sum \mathbf{1}_{B_k} < \infty$ then $M_n \rightarrow -\infty$, which is neither alternative; so $\sum \mathbf{1}_{B_k} = \infty$. $\square$

6. Cauchy's condensation test

If $b_n \geq 0$ is nonincreasing for $n \geq n_0$, then $\sum b_n < \infty$ if and only if $\sum_k 2^k b_{2^k} < \infty$. Indeed, for $2^k \geq n_0$ the block $2^k \leq n < 2^{k+1}$ has 2^k terms, each between $b_{2^{k+1}}$ and b_{2^k} , so its sum lies between $\frac{1}{2} 2^{k+1} b_{2^{k+1}}$ and $2^k b_{2^k}$. For $b_n = 1/(n \log^c n)$, which is eventually nonincreasing for every real c , the condensed terms are $(k \log 2)^{-c}$, and their series converges if and only if $c > 1$.

7. Sums and integrals of unimodal functions

Let $f \geq 0$ be integrable on $[1, \infty)$, nondecreasing on $[1, x_0]$ and nonincreasing on $[x_0, \infty)$, with maximum $M_f = f(x_0)$. Then

$$\left| \sum_{n \geq 1} f(n) - \int_1^\infty f(x) dx \right| \leq M_f.$$

Let $N = \lfloor x_0 \rfloor$. For $1 \leq n \leq N-1$, f is nondecreasing on $[n, n+1]$, so $f(n) \leq \int_n^{n+1} f \leq f(n+1)$; for $n \geq N+1$, f is nonincreasing on $[n, n+1]$, so $f(n+1) \leq \int_n^{n+1} f \leq f(n)$. Summing,

$$\begin{aligned} \sum_{n \geq 1} f(n) &\leq \int_1^N f + f(N) + f(N+1) + \int_{N+1}^\infty f, \\ \sum_{n \geq 1} f(n) &\geq f(1) + \int_1^N f + \int_{N+1}^\infty f. \end{aligned}$$

On $[N, N+1]$ the function is at least $\min(f(N), f(N+1))$, because it rises and then falls there; so $\int_N^{N+1} f \geq \min(f(N), f(N+1))$, and the first inequality gives $\sum f(n) \leq \int_1^\infty f + \max(f(N), f(N+1))$. The second gives $\sum f(n) \geq \int_1^\infty f - \int_N^{N+1} f \geq \int_1^\infty f - M_f$. For $f(x) = x^{-p} \log x$ with $p > 1$ one has $x_0 = e^{1/p}$, $M_f = 1/(pe)$ and $\int_1^\infty f = 1/(p-1)^2$, which is the estimate used in Proposition IX.3; the same argument, with $x_0 = 1$, gives $1/(p-1) \leq \zeta(p) \leq 1 + 1/(p-1)$.

8. Extinction of a Galton–Watson process

Let $N \geq 0$ be an integer random variable with $f(z) = \mathbb{E}z^N$, mean $m = f'(1^-) \in [0, \infty]$ and $\mathbb{P}(N=1) < 1$, and let (Z_k) be the Galton–Watson process with $Z_0 = 1$ and offspring law N . Then the extinction probability $e = \mathbb{P}(Z_k = 0 \text{ for some } k)$ is the least root of $f(z) = z$ in $[0, 1]$, and $e = 1$ if and only if $m \leq 1$ [4, 183].

Proof. Conditioning on the first generation, the generating function of Z_k is the k -fold composition $f_k = f \circ \dots \circ f$, so $\mathbb{P}(Z_k = 0) = f_k(0)$. These numbers increase with k to e , and $f(e) = e$ by continuity of f on $[0, 1]$. If $z^* \in [0, 1]$ is a root, then $f_k(0) \leq f_k(z^*) = z^*$ because f is nondecreasing, so $e \leq z^*$: e is the least root. If $\mathbb{P}(N \geq 2) = 0$, then $f(z) = \mathbb{P}(N=0) + \mathbb{P}(N=1)z$ with $m = \mathbb{P}(N=1) < 1$, and $f(z) > z$ on $[0, 1)$, so $e = 1$. Otherwise f' is strictly increasing on $[0, 1)$. If $m \leq 1$, then $f'(z) < 1$ on $[0, 1)$, so $f(z) - z$ is strictly decreasing there with limit 0 at 1, hence positive, and the only root is 1. If $m > 1$, then $f(z) < z$ for $z < 1$ close to 1, while $f(0) \geq 0$, so there is a root in $[0, 1)$, and $e < 1$. $\square$

The theorem of Kesten and Stigum [103], used in the proof of Theorem E, is not proved here: if $1 < m < \infty$, the martingale Z_k/m^k converges almost surely to a limit W , and $W > 0$ almost surely on survival if and only if $\mathbb{E}N \log N < \infty$; the proof is in [4, Chapter I].

Appendix B: Formal proofs and numerical checks

1. The Lean development

The finite inequalities of the paper are proved in Lean 4 with Mathlib, in fifteen files and sixty-five theorems, listed in Tables XIX and XX. Every theorem depends only on the three standard axioms of Lean, propositional extensionality, the axiom of choice and the soundness of quotients; the command `#print axioms` is run on each of them, and no theorem uses `sorry`. The statements are over the real numbers, with finite sums indexed by a `Finset` and infinite ones expressed by `Summable`; in them σ is the polynomial order written s in the text, and $(1/D)^r$ is ε^r with $\varepsilon = 1/D$.

The formal statements cover the deterministic content of the main theorems: Gibbs' inequality in the form used for the entropy law, the lower bound and the two properties of the ceiling schedule (Theorem A), the correspondence between schedules and Kraft vectors and the shifted cost (Theorem B), Hölder's lower bound and the optimal allocation of the Rényi law (Theorem C) with the summability criterion for polynomial checks, both halves of the duality criterion (Theorem VII.3), the inequalities for clean runs and random schedules (Propositions V.8 and V.9), the precision and cost identities of the gating theorem (Theorem VIII.2), the summability criterion for sound singular chains with both directions (Theorem D(iii)), the two Borel–Cantelli statements for false acceptances (Theorem D(i)), the divergence and amplification statements of the branching theorem (Theorem E), the online, subagent, cascade and gap laws (Theorems G, H, XI.1 and I), and the four parts of the tower law (Theorem XIV.1). The limit theorems of probability (Wald's identity, the extinction theorem, the Kesten–Stigum theorem and Lévy's lemma), the asymptotics of ζ and of the entropies in Propositions IX.3 and F.1, and the estimates of Proposition V.11 are proved in the text and are not formalised.

2. Numerical checks

Independent programs test the closed forms of the paper against exhaustive searches, exact optima and simulation, on instances drawn with fixed seeds. They are listed in Table XXI; the shell script that runs them also builds the Lean development and prints the axioms of every theorem; the programs run 55,271 checks in all, with no failure. The checks are not part of any proof. They guard against errors in the statements: a wrong constant, a reversed inequality or a misprinted formula would make one of them fail. The numerical values quoted in the text, in Tables IV and VI and in the figures are produced by the same programs.

Appendix C: Hardware and reproduction

Every figure below was measured on one machine with four virtual processors of an Intel Xeon at 2.10 GHz, 16 GB of memory, no graphics processor and about 40 GB of writable

TABLE XIX. The sixty-five theorems of the Lean development, by file. The third column gives the statement in the notation of the paper and the fourth the result in which it is used. Sums over i are finite; sums over n or k are infinite and “ $\sum < \infty$ ” means summability.

File	Theorem	Statement	Used in
Basic	<code>gibbs_term</code>	$x \log \frac{1}{x} \leq x \log \frac{1}{y} + y - x$ for $x, y > 0$	Lemma V.1
	<code>entropy_term</code>	$a \log \frac{A}{a} + a \log \frac{1}{\theta} \leq a \log \frac{1}{b} + \frac{Ab}{\theta} - a$	Lemma V.2
	<code>log_inv_pow</code>	$\log(1/\varepsilon^r) = r \log(1/\varepsilon)$	Theorem A
	<code>log_inv_pos</code>	$\log(1/\varepsilon) > 0$ for $0 < \varepsilon < 1$	Theorem A
EntropyLaw	<code>entropy_lower</code>	$\sum \beta_i \leq \theta \Rightarrow \sum a_i \log \frac{A}{a_i} + A \log \frac{1}{\theta} \leq \sum a_i \log \frac{1}{\beta_i}$	Proposition V.3
	<code>verification_lower</code>	$\sum \varepsilon^{r_i} \leq \theta \Rightarrow \frac{A}{L} (H + \log \frac{1}{\theta}) \leq \sum a_i r_i$	Theorem A
	<code>ceilSchedule_sound</code>	the ceiling schedule has $\sum \varepsilon^{r_i} \leq \theta$	Theorem A
	<code>ceilSchedule_cost</code>	the ceiling schedule costs at most $\frac{A}{L} (H + \log \frac{1}{\theta}) + A$ for $\theta \leq 1$	Theorem A
Kraft	<code>kraft_of_schedule</code>	$\sum D^{-r_i} \leq D^{-k} \Rightarrow r_i \geq k$ and $\sum D^{-(r_i-k)} \leq 1$	Lemma VI.1(i)
	<code>schedule_of_kraft</code>	$\sum D^{-\ell_i} \leq 1 \Rightarrow \sum D^{-(\ell_i+k)} \leq D^{-k}$	Lemma VI.1(ii)
	<code>shifted_cost</code>	$\sum a_i(\ell_i + k) = kA + \sum a_i \ell_i$	Lemma VI.1(iii)
Renyi	<code>renyi_lower</code>	$\sum \delta_i \leq \theta \Rightarrow (\sum a_i^{1/(1+s)})^{1+s} \leq \theta^s \sum a_i \delta_i^{-s}$	Theorem C
	<code>renyi_attain</code>	$\delta_i = \theta a_i^{1/(1+s)}/S$ has $\sum \delta_i = \theta$ and cost $\theta^{-s} S^{1+s}$	Theorem C
Duality	<code>poly_schedule_iff</code>	a positive δ with $\sum \delta_n < \infty$, $\sum a_n \delta_n^{-s} < \infty$ exists $\iff \sum a_n^{1/(1+s)} < \infty$	Corollary VII.5(ii)
	<code>dualCost_nonneg</code>	$\psi(u) \geq 0$ for $u \geq 0$	Theorem VII.3
	<code>dualCost_le</code>	$\psi(u) \leq \kappa(\delta) + u\delta$ for $\delta \in (0, 1]$	Theorem VII.3
	<code>dual_summable_of_schedule</code>	a sound allocation of finite cost gives $\sum a_n \psi(\lambda/a_n) < \infty$ for every $\lambda > 0$	Theorem VII.3
Clean	<code>schedule_of_dual_summable</code>	$\sum a_n \psi(\lambda/a_n) < \infty$ for one $\lambda > 0$ gives a sound allocation of finite cost	Theorem VII.3
	<code>clean_le_exp</code>	$\prod (1 - q_i) \leq e^{-\sum q_i}$ for $q_i \leq 1$	Proposition V.8
	<code>clean_ge</code>	$1 - \sum q_i \leq \prod (1 - q_i)$ for $0 \leq q_i \leq 1$	Proposition V.8
	<code>clean_budget</code>	$\prod (1 - q_i) \geq 1 - \theta \Rightarrow \sum q_i \leq \log \frac{1}{1-\theta}$	Proposition V.8
Adaptive	<code>jensen_repetition</code>	$\varepsilon^{\sum w_j r_j} \leq \sum w_j \varepsilon^{r_j}$ for weights w	Proposition V.9
	<code>randomised_budget</code>	a random schedule within budget in mean has its mean within budget	Proposition V.9
Gating	<code>step_precision</code>	$\frac{(1-p)(1-\eta)}{(1-p)(1-\eta)+p\delta} = \frac{1}{1+x}$	Theorem VIII.2(i)
	<code>precision_bounds</code>	$e^{-\sum x_i} \leq \prod (1+x_i)^{-1} \leq e^{-\sum x_i/(1+x_i)}$	Theorem VIII.2(ii)
	<code>outcome_precision</code>	$\frac{c(1-\eta_0)}{c(1-\eta_0)+(1-c)\delta_0} = \left(1 + \frac{(1-c)\delta_0}{c(1-\eta_0)}\right)^{-1}$	Theorem VIII.2(ii)
	<code>separation</code>	cost ratio $(1-p)^{-n}$ and $1+n\rho \leq (1-p)^{-n}$	Theorem VIII.2(iii)

disk, running Ubuntu 24.04.5 with Linux 6.18, Python 3.13.16 with NumPy 2.5.3, NetworkX 3.7 and PySAT 1.9 (whose CaDiCaL, Glucose and MiniSat solvers decide every satisfiability instance), Lean 4.34.1 for the package and Lean 4.33.0 with Mathlib at the tag v4.33.0 for the paper; an entry marked as an estimate was not measured. The package checks a machine against the minimum column before a run, and one command runs the hardware check, the unit tests, the memory study, the channel study and the agent loop in that order, writing every output to a fresh directory so that the recorded data are never overwritten.

Two runs did not fit the machine. Rebuilding a corpus of 121,734 Lean files with Mathlib needs, by our estimate, about 60 GB of free disk, against 7.8 GB available after cleaning; the interval-arithmetic check of a box of side about 90, the size that leaf 5 of Table XIII needs, was not attempted, and we estimate that it needs a cluster. Neither result is used in the paper.

The command `scripts/run_all.sh` runs the five steps; with `-quick` it uses two seeds and two rounds per study and took 46 s here, the largest process reaching 126 MB of resident memory. The full run uses 60 seeds, and its memory study took 330 s; the channel study reached 1.2 GB of resident memory. Every random choice in the package is drawn from a generator seeded explicitly, Python’s `random.Random` (the Mersenne Twister) or, in one Yang–Mills script, the default generator of NumPy. The seed of a run is its run number, or in the Hodge rounds a fixed seed plus the round number,

and the scripts that produced the tables and figures fix them, so a rerun with the same library versions repeats every draw; a run stopped by a time limit, such as the burst round of Section XIX, depends in addition on the speed of the machine.

Appendix D: What the engine draws on

The list at the end of this appendix is the registry of Section XVI, one entry for each system, method or collection that the engine takes something from: what it takes, the place in the package where it is used, and the part it does not take. The list is written from the registry file of the package, so that the paper and the code cannot drift apart, and the command `anaxia sources` prints the same entries with their links. A source is cited where it has a paper; otherwise its address is given. The status after each name is the status of the source, not of the engine: a claim that is unrefereed in its source stays unrefereed when the engine indexes it, and an index entry is used only to check whether a target is still open or to supply a construction that the gate of Section XV G then checks again.

The three indexes cover 6,539 items: 5,035 from the public repositories of Google DeepMind, of which 3,908 are the statements of Formal Conjectures (1,471 open and 2,437 solved) [70]; 782 problem dossiers of OpenRefute, of which 8 are open, 296 proved, 235 refuted and 243 otherwise settled [127]; and the 722 manuscripts of the OpenAI collection, ev-

TABLE XX. The Lean development, continued.

File	Theorem	Statement	Used in
Singularity	entropy_summable_of_schedule	$\sum \varepsilon^{r_n} < \infty, \sum a_n r_n < \infty, a_n \leq 1 \Rightarrow \sum a_n \log \frac{1}{a_n} < \infty$	Theorem D(iii)
	costSchedule_pow_le	$r_n = \lceil \log(1/a_n)/L \rceil$ has $\varepsilon^{r_n} \leq a_n$	Proposition VII.6
	schedule_of_entropy_summable	$\sum a_n < \infty, \sum a_n \log \frac{1}{a_n} < \infty$ make that schedule sound and of finite cost	Theorem D(iii)
	sound_schedule_iff	for $\sum a_n < \infty, a_n \leq 1$: a sound schedule of finite cost exists $\iff \sum a_n \log \frac{1}{a_n} < \infty$	Proposition VII.6
	shift_schedule	$\varepsilon^{r_{n+k}} = \varepsilon^k \varepsilon^{r_n}$	Theorem D(iii)
Soundness	false_accepts_finite	$\mathbb{P}(E_n) \leq C q_n, \sum q_n < \infty \Rightarrow \mathbb{P}(E_n \text{ i.o.}) = 0$	Theorem D(i)
	false_accepts_infinite	independent $E_n, \mathbb{P}(E_n) \geq b q_n, b > 0, \sum q_n = \infty \Rightarrow \mathbb{P}(E_n \text{ i.o.}) = 1$	Theorem D(i)
	any_false_accept_le	$\mathbb{P}(\bigcup E_n) \leq C \sum q_n$	Theorem D(i)
Branching	contamination_diverges	$\sum m^k \delta = \infty$ for $m \geq 1, \delta > 0$	Theorem E
	amplified_term	$k \log(2m) \leq r_k L \Rightarrow m^k \varepsilon^{r_k} \leq 2^{-k}$	Theorem E
	amplified_summable	under the same hypothesis $\sum m^k \varepsilon^{r_k} < \infty$	Theorem E
	amplified_schedule_ok	$r_k = \lceil k \log(2m)/L \rceil$ satisfies that hypothesis	Theorem E
Online	ratio_le_log	$1 - s/t \leq \log(t/s)$ for $s, t > 0$	Theorem G
	telescope_step	$\frac{y-x}{(1+y)^2} \leq \frac{1}{1+x} - \frac{1}{1+y}$ for $0 \leq x \leq y$	Theorem G
	telescope_sum	$\sum_n \frac{u_{n+1} - u_n}{(1+u_{n+1})^2} \leq \frac{1}{1+u_0} - \frac{1}{1+u_N}$ for monotone $u \geq 0$	Theorem G
	onlineWeight_sum_le	$\sum_{n < N} w_n \leq 1$ for every sequence of positive costs	Theorem G
	onlineWeight_pos	$w_n > 0$	Theorem G
	onlineSchedule_sound	the online rule (5) has $\sum_{n < N} \varepsilon^{r_n} \leq \theta$ for every N	Theorem G
	small_weight_often	$w \geq 0, \sum_{n < N} w_n \leq 1$ for all $N \Rightarrow w_N \leq \frac{1}{N+1}$ infinitely often	Theorem G
	online_lower	a θ -sound schedule has $r_N L \geq \log \frac{N+1}{\theta}$ infinitely often	Theorem G
	doubling_sum	$\sum_{k < N} \frac{k+1}{2^{k+1}} = 2 - \frac{N+2}{2^N}$	Theorem G
	groupCost_pos	$A_k > 0$ for a nonempty group	Theorem H
Agents	groupCost_sum	$\sum_k A_k = A$	Theorem H
	chain_rule	$\sum a_i \log \frac{A}{a_i} = \sum_k A_k \log \frac{A}{A_k} + \sum_k \sum_{i \in G_k} a_i \log \frac{A_k}{a_i}$	Theorem H(i)
	split_lower	$\sum \theta_k \leq \theta \Rightarrow$ the group lower bounds add up to at least the flat one	Theorem H(ii)
	proportional_split	the ceiling schedule of G_k at budget $\theta A_k/A$ is the flat one on G_k	Theorem H(iii)
	cascade_lower	$br + cpe^r \geq \frac{b}{L} (1 + \log \frac{cpe^r}{b})$ for every r	Theorem XI.1
	cascade_upper	$b \leq cpe^r \Rightarrow r = \lceil \frac{1}{L} \log \frac{cpe^r}{b} \rceil$ costs at most that plus b	Theorem XI.1
	search_lower	$\sum q = 1 \Rightarrow (\sum a_j)^2 \leq \sum a_j^2 / q_j$, with $a_j = \sqrt{\pi_j}$	Theorem I
	search_optimal	equality at $q_j = a_j / \sum a$	Theorem I
Gap	search_chi2	$\sum a_j^2 / q_j = (\sum a)^2 (1 + \chi^2(r q))$ with $r_j = a_j / \sum a$	Theorem I
	search_mix	$q \geq \lambda r \Rightarrow \sum a_j^2 / q_j \leq (\sum a)^2 / \lambda$	Theorem I
Tower	tower_cost_lower	$\beta_k > 0, \sum_{k < n} \beta_k \leq \theta \Rightarrow n \log \frac{n}{\theta} \leq \sum_{k < n} \log \frac{1}{\beta_k}$	Theorem XIV.1(i)
	tower_checks_lower	$\sum_{k < n} \varepsilon^{r_k} \leq \theta \Rightarrow n \log \frac{n}{\theta} / \log \frac{1}{\varepsilon} \leq \sum_{k < n} r_k$	Theorem XIV.1(i)
	tower_increment	$\log \frac{n+1}{\theta} \leq (n+1) \log \frac{n+1}{\theta} - n \log \frac{n}{\theta}$	Theorem XIV.1(ii)
	tower_increment_unbounded	$\log \frac{n+1}{\theta} \rightarrow \infty$	Theorem XIV.1(ii)
	tower_saturates	$S_n \geq B, V_n \geq cn \log \frac{n}{\theta}, c > 0, \theta \leq 1 \Rightarrow S_n + V_n > T_1$ for all large n	Theorem XIV.1(iii)
	stage_cost_diverges	$\beta_k > 0, \sum \beta_k < \infty \Rightarrow \log \frac{1}{\beta_k} \rightarrow \infty$	Theorem XIV.1(iv)

ery one marked unrefereed [125]. No weights of any model are bundled with the package or the image.

Google DeepMind

FunSearch [143], refereed. *Takes*: Evolve and evaluate: a population of strategies, each scored by an exact evaluator; the better ones are kept and mutated. *In*: src/ebllm/construct/tower.py (Layer2); the funsearch constructions are indexed by src/ebllm/blm/deepmind.py. *Leaves*: Program mutation by a language model and the island model: no model runs in construction mode.

AlphaEvolve [123], preprint. *Takes*: The evaluation cascade: cheap tests discard most candidates before the expensive test, and the evaluator, not the proposer, decides. *In*: src/ebllm/construct/proof_builder.py (try_add: forbidden cycles, contact distances, planarity, then the SAT forcing check, then minimise) and tower.meets (the gate). *Leaves*: The Gemini ensemble and the evolution of whole programs; the 67 construction problems and AlphaEvolve's results enter only as indexed targets and library nodes.

AlphaProof [90], refereed. *Takes*: A curriculum: easier variants of the target are solved first and what they yield is kept for the target. *In*: src/ebllm/construct/tower.py (Layer2.profile: the target's weaker contact profiles first; each gadget found joins the library as a lemma). *Leaves*: Reinforcement learning, the formaliser network and the Lean proof

search; the AlphaProof Nexus proofs are read by src/ebllm/blm/corpus.py (read_alphaproof).

AlphaGeometry [174], refereed. *Takes*: Deduction run to a fixed point, with a construction proposed only when deduction stalls. *In*: src/ebllm/construct/proof_builder.py (status: the shared colour spreads by forward rules; Builder.moves: attach, link and lemma moves when no closing move applies). *Leaves*: The language model that proposes auxiliary points, and the geometry engine DDAR itself.

AlphaGeometry2 [54], preprint. *Takes*: Nothing beyond AlphaGeometry's division of labour; pinned and fetchable. *In*: anaxia fetch deepmind alphageometry2. *Leaves*: DDAR2 (geometry only).

Formal Conjectures [70], software. *Takes*: The statements with their open or solved status: the target catalogue and the check that a target is still open. *In*: src/ebllm/blm/deepmind.py (index, Search); anaxia intake deepmind --search. *Leaves*: The Lean statements are not compiled by Anaxia.

Mathematics dataset [155], refereed. *Takes*: Pinned as a generator of warm-up questions for a local proposer. *In*: anaxia fetch deepmind mathematics_dataset; experiments/ fine-tuning script. *Leaves*: Not used in any construction-mode run or measurement.

AlphaTensor [69], refereed. *Takes*: The decompositions, indexed as library constructions. *In*: src/ebllm/blm/deepmind.py. *Leaves*: The

TABLE XXI. The numerical checks.

Language	What is checked	Checks
Python	Theorems A, B: bounds and Huffman cost against exact optima	649
Python	Theorem VIII.2: costs and precisions against simulation	533
Python	Theorems C, VII.3: Rényi optimum, dual costs	20,850
Python	Theorem D, Propositions IX.3, F.1	241
Python	Theorem G: online soundness, bounds, doubling lower bound	1,581
Python	Theorems H, XI.1: chain rule, split price, cascades	8,104
Python	Theorem E: takeoff, kept errors, amplified checks	2,432
Python	Theorem I, Propositions XII.1, XII.2: gap law, square-root sampling, spiking selector	12,081
C	Theorem B: exhaustive search over schedules	800
C++	Theorems C, VIII.2 at random feasible points	8,000
total		55,271

reinforcement-learning game.

AlphaDev [118], refereed. *Takes*: The sorting programs, indexed. *In*: `src/ebllm/blm/deepmind.py`. *Leaves*: The method.

Mathematics conjectures (Davies et al.) [65], refereed. *Takes*: The notebooks, indexed. *In*: `src/ebllm/blm/deepmind.py`. *Leaves*: The attribution method, which needs data of a kind the targets do not supply.

Aletheia and LEAP (superhuman),

`github.com/google-deepmind/superhuman`, software; outputs unrefereed. *Takes*: The generate, verify and revise loop with a verifier independent of the generator, which is the refuter pass; one Aletheia output as a library node of the Hodge attack. *In*: `src/ebllm/blm/superbrain.py` and `hodge.py` (the refuter pass); `experiments/hodge/absorb_sources.py`. *Leaves*: The Erdős, Kirby and FirstProof outputs.

Methods

Successive halving [98], refereed. *Takes*: Keep the better half of the arms after each round. *In*: `src/ebllm/construct/tower.py` (LayerK). *Leaves*: —

Hebbian learning [87], book. *Takes*: A connection used in a success is strengthened and every connection decays. *In*: `src/ebllm/construct/tower.py` (Mesh.reinforce, Mesh.bias). *Leaves*: —

Postnikov towers [84], book. *Takes*: The framing of the layer tower: stage n is the engine cut at depth n and each stage is the one below with one layer added; an analogy, not a construction of a Postnikov tower. *In*: paper; Lean BLLM.tower_increment states the precise part (each added layer raises the cost bound). *Leaves*: —

OpenAI

OpenAI mathematics manuscript collection [125], unrefereed claims (the collection says that some unformalised results could have issues). *Takes*: The move statistics of the crossing network and an index of the manuscripts for the check that a target is still open, every item marked unrefereed. *In*: `src/ebllm/blm/corpus.py` (read_openai); `src/ebllm/blm/sources.py` (index_openai). *Leaves*: The claims themselves.

OpenAI Navier-Stokes Lean proof [124], Lean build checked; paper unrefereed. *Takes*: A literature row with its status; its build record applies the rule of Anaxia's Lean check (kernel check, only propext, Classical.choice and Quot.sound). *In*: `src/ebllm/blm/sources.py` (registry). *Leaves*: Nothing of the mathematics enters the engine.

The Omega Institute

trureturing (OpenRefute) [127], Lean results kernel-checked; the reading

of each paper's statement as a Lean claim is the repository's own and unrefereed. *Takes*: An index of the 782 problem dossiers (statement, resolution, Lean declaration, link): refuter data, since it records which published conjectures fell to an explicit instance, and a source for the check that a target is still open. *In*: `src/ebllm/blm/sources.py` (index_openrefute); `anaxia` sources check. *Leaves*: The Lean library (not compiled here) and the judging tools.

Memory

obsidian-brain [181], software. *Takes*: Memory as a vault of linked notes, and retrieval that returns each hit with its link neighbours. *In*: `src/ebllm/blm/vault.py` (Vault.write; Vault.search: TF-IDF over the notes plus the notes a hit was built from or that were built from it). *Leaves*: The Node server, the embedding model and the community detection.

claude-obsidian (lab009dm), software. *Takes*: Notes captured with their provenance and linked to what they were built from. *In*: `src/ebllm/blm/vault.py`. *Leaves*: The plugin and its health checks.

RealMythos, `github.com/tszdanger/RealMythos`, software. *Takes*: Nothing: its content is vulnerability analysis; its staged layout (data, model, environment, traces) is already the layout of Anaxia's benchmark. *Leaves*: All of it, as off-topic.

Advanced Reasoning MCP,

`github.com/creelie/advanced-reasoning-mcp`, software. *Takes*: The TF-IDF ranking of its memory search. *In*: `src/ebllm/blm/deepmind.py` (Search). *Leaves*: The sequential-thinking tool with self-reported confidence.

Agents

ultra-mathematician, `github.com/txmy/ultra-mathematician`, software. *Takes*: The closure policy: closed only after a kernel-checked Lean endpoint with an audited statement and allowed axioms, never native_decide; evidence levels lead, candidate, reproduced, formally verified, accepted. *In*: the labels of anaxia controls and the release rule. *Leaves*: The host-specific skills.

mathcode, `github.com/math-ai-org/mathcode`, software. *Takes*: Only a strict check of one fully qualified declaration certifies completion, and stored assumptions are never injected silently. *In*: the Lean verifier and the axiom audit (anaxia axioms). *Leaves*: The CLI, web interface and backend.

open co-scientist (lab007dm), software. *Takes*: Nothing in code: the tournament of hypotheses was considered and not built, since the checks and the refuter already rank survivors. *Leaves*: The tournament and the research-agent stack.

Corpora

OpenConjecture (lab010dm), conjectures as stated in arXiv papers; solver attempts unverified. *Takes*: The candidate pool for conjecture attacks; six Hodge-related conjectures entered the Hodge library. *In*: `experiments/hodge/absorb_sources.py`. *Leaves*: The model-reported solver attempts.

Provers and models

DeepSeek-Prover-V2 [149], preprint. *Takes*: Can serve as the proposer through an OpenAI-compatible server; its subgoal decomposition is the split of an obligation graph into steps. *In*: `src/ebllm/blm/proposers.py` (provider local). *Leaves*: Not run in the paper's measurements (no GPU). *Weights*: not bundled; served by the user through the local connector.

Kimina-Prover [186], preprint. *Takes*: Can serve as the proposer through the local connector. *In*: `src/ebllm/blm/proposers.py` (provider local). *Leaves*: Not run in the paper's measurements. *Weights*: not bundled; served by the user through the local connector.

Goedel-Prover [110], preprint. *Takes*: Can serve as the proposer through the local connector. *In*: `src/ebllm/blm/proposers.py` (provider local). *Leaves*: Not run in the paper's measurements. *Weights*: not bundled; served by the user through the local connector.

Qwen2.5-Math [189], preprint. *Takes*: Can serve as the proposer through the local connector; `experiments/finetune/finetune.py` fine-tunes a Qwen2.5 model (default Qwen/Qwen2.5-0.5B-Instruct) with LoRA on the choices the checks confirmed. *In*: `src/ebllm/blm/proposers.py` (provider

TABLE XXII. Hardware for each part of the package. “Measured” refers to the machine described in the text.

Part	Minimum	Recommended	Measured
crossing network, boosters, unit tests	1 CPU, 1 GB, 200 MB disk	4 CPU, 4 GB	99 tests in 35 s; the booster study of 200 seeds in minutes
memory store (SQLite)	50 MB disk	1 GB disk	156 kB after five rounds of one seed; 372 MB after the twelve rounds of the Hodge run, most of it the records of the channel
memory study, 60 seeds, five rounds, five arms	2 CPU, 2 GB	4 CPU, 4 GB	330 s, sharing the machine with three annealing runs
agent loop	2 CPU, 2 GB	as above, with the local model	seconds per round without the model
Hodge run, steady round	1 CPU, 1 GB	4 CPU, 2 GB	30 to 155 s per round
Hodge run, burst round	1 CPU, 2 GB	4 CPU, 4 GB	29 checks in 98 min; stopped at its limit of 7000 s, the channel bringing 58,872 records
construction mode	1 CPU, 0.5 GB, 10 MB disk	one search per core	G_1 rebuilt in 10.8 s, G_2 from lemmas in 2.1 s, the 166-vertex check in 0.3 s; peak 39 MB
layer tower, depth 1 to 3	1 CPU, 0.25 GB	one tower per core	depth-3 mesh on the G_2 target at 3000 checks: 36.9 s, peak 39 MB; time linear in the budget
tower bench, 120 runs	1 CPU and 0.25 GB per job	4 CPU, three jobs	5.6 processor-hours in all, 1.9 h on three cores; 4 to 400 s per run
local model, 0.5×10^9 parameters, 4-bit, CPU	2 CPU, 1.5 GB, 0.5 GB disk	4 CPU, 2 GB	file 469 MB, peak resident memory 784 MB, 41 tokens per second
local model, 1.5×10^9 parameters, 4-bit, CPU	4 CPU, 3 GB, 1.2 GB disk	8 CPU or a 4 GB GPU	file 1.1 GB, peak 2.1 GB, 16 tokens per second
fine-tuning the local model on its checked successes	not run	estimate: a 16 GB GPU, 32 GB memory	not measured
Lean development of the package, without Mathlib	1 GB, 3 GB disk	2 GB	cold build 1.8 s, peak 523 MB; toolchain 3.0 GB
Lean development of the paper, with Mathlib	8 GB, 12 GB disk	16 GB, 20 GB disk	Mathlib 7.2 GB, toolchain 2.9 GB; cached check of 8721 jobs in 6.1 s, peak 921 MB
satisfiability certificates for Erdős problem 1183	1 CPU, 2 GB	4 CPU, 8 GB	$n = 12$, sublattices of 8 members: 20 min 31 s on one core
grid search for Erdős problem 1183 (C)	1 CPU, 50 MB	one core per annealing run	one colouring of $2^{[20]}$ in 0.01 to 8 s; 4000 annealing steps in about 3.5 min
interval-arithmetic mixing check on a gauge-fixed box	not run	estimate: a cluster	not attempted

local). *Leaves*: Not run in the paper’s measurements. Weights: not bundled; served by the user through the local connector.

Mathstral, mistral.ai/news/mathstral, model card. *Takes*: Can serve as the proposer through the local connector or the mistral provider. *In*: `src/ebllm/bllm/proposers.py` (providers local, mistral). *Leaves*: Not run in the paper’s measurements. Weights: not bundled; served by the user through the local connector.

HyperTree Proof Search (Meta AI) [108], refereed. *Takes*: Search over an AND-OR tree of goals, which is the shape of Anaxia’s obligation graph. *In*: `src/ebllm/obligations.py`. *Leaves*: The trained models. Weights: not released for Lean 4.

LeanDojo [188], refereed. *Takes*: Premise retrieval: proposals are prompted with retrieved library nodes. *In*: `src/ebllm/bllm/memory.py` (retrieval). *Leaves*: The extraction tool and the trained retriever. Weights: not bundled.

Seed-Prover [41], preprint. *Takes*: Lemma-style proving, in which proved lemmas are kept and reused: the same shape as the lemma library of construction mode. *In*: `src/ebllm/construct/tower.py` (Library). *Leaves*: The model (not released). Weights: not released.

Aristotle (Harmonic) [1], preprint. *Takes*: Informal reasoning checked step by step in Lean, which is the contract of Anaxia’s verifier. *Leaves*: The system (not released). Weights: not released.

Language-model APIs, services. *Takes*: Proposers behind one interface; keys come from the environment. *In*: `src/ebllm/bllm/proposers.py`. *Leaves*: No key is stored in the package or the image. Weights: none: remote services.

Appendix E: Three nervous systems and one plan

The selector of Section [XIID](#) borrows three mechanisms from nervous systems: a race to threshold, a reward that potentiates the winner, and homeostatic scaling. This section shows where the rest of the architecture of a BLLM sits in the wiring of real brains. We use three complete reconstructions: the nervous system of the nematode *Caenorhabditis elegans* [45], the whole brain of the adult fruit fly *Drosophila melanogaster* [61], and a cubic millimetre of human temporal cortex [160].¹⁰ In each of them the same plan appears: signals enter at one end, are integrated in a middle stage that is mostly

¹⁰ The worm is the hermaphrodite of S. J. Cook et al., Nature 571 (2019) 63–71 [45], in the edge list distributed by the OpenWorm project, with the classes of sensory and motor neurons of L. R. Varshney et al. [177]. The fly is FlyWire, S. Dorkenwald et al., Nature 634 (2024) 124–138 [61], release 783 of the proofread connections (Zenodo, doi:10.5281/zenodo.10676866, CC BY 4.0), with the transmitters predicted from electron micrographs by N. Eckstein et al. [67]. The cortex is H01, A. Shapson-Coe et al., Science 384 (2024) eadk4858 [160], in its table of somas. The files are public and are not redistributed; the BLLM package reads them from the paths a user gives, and the summaries behind Figs. 29 to 31 are computed by a program in the repository of the paper.

recurrent, and leave at the other end, with feedback from the output back to the input (Fig. 28(a)). A BLLM has the same plan with a different meaning of each stage (Fig. 28(b)): the proposers sense, the selector integrates, the checks act, and the memory feeds back.

1. The worm: depth of processing

The hermaphrodite connectome has 448 nodes, 300 neurons and 148 muscles and end organs, with 27,019 chemical synapses in 4,681 connections and 12,683 gap-junction contacts in 2,698. Varshney and coauthors gave its neurons a *processing depth* [177]: a coordinate that increases along the direction in which signals travel. Let W be the matrix of synaptic weights, $\text{in}_k = \sum_i W_{ik}$ and $\text{out}_k = \sum_j W_{kj}$, and let L be the Laplacian of the symmetric matrix $(W + W^\top)/2$.

Lemma E.1 (Processing depth). *The system $Lz = \text{in} - \text{out}$ has a solution, unique up to an additive constant on each connected component, and it is the minimiser of $\sum_{i,j} W_{ij}(z_j - z_i - 2)^2$.*

Proof. The kernel of L is spanned by the indicator vectors of the connected components, and L is symmetric, so its range is the set of vectors that sum to zero on every component. Each synapse $i \rightarrow j$ adds W_{ij} to in_j and to out_i , both in the same component, so $\text{in} - \text{out}$ sums to zero there. For the variational form, $z^\top Lz = \frac{1}{2} \sum_{i,j} W_{ij}(z_j - z_i)^2$ and $(\text{in} - \text{out})^\top z = \sum_{i,j} W_{ij}(z_j - z_i)$, so $\frac{1}{2} z^\top Lz - (\text{in} - \text{out})^\top z = \frac{1}{4} \sum_{i,j} W_{ij}(z_j - z_i - 2)^2 - \sum_{i,j} W_{ij}$, a convex function whose critical points are the solutions. $\square$

So z is the arrangement in which every synapse comes as close as possible to pointing two units deeper; with another normalisation of the Laplacian the unit changes and the order does not. Computed on the chemical and electrical synapses together, the mean depth is -1.92 for the 86 sensory neurons, -0.68 for the 100 interneurons, -0.22 for the 114 motor neurons and $+1.75$ for the muscles. Four of the five shallowest neurons are amphid sensory neurons, AWAL, AWAR, ASIL and ASEL, and the deepest nodes are pharyngeal muscles and marginal cells. The order sensory, inter, motor, muscle is not imposed; it is computed from the wiring alone, and it is the order of the three stages of Fig. 28(a). Figure 29 draws every node at its depth, against a spectral coordinate: the rank of the node in the lowest eigenvector of the normalised Laplacian that is spread over at least a quarter of the neurons, the third, with eigenvalue 0.123.

The package can also run this wiring. It builds an echo-state network whose recurrent matrix is the connectome itself, scaled to spectral radius 0.9, with one sign per presynaptic neuron, as Dale's principle requires (excitatory with probability 0.8, since the transmitter map of the worm is not part of the wiring data), and fits a linear readout [96]. Its short-term memory, the squared correlation between the input k steps back and its best linear readout from sixty units, is 0.77, 0.37, 0.27 and 0.14 for $k = 1, 2, 4, 8$. This makes no claim that a worm-shaped reservoir computes better than another; it lets the memory and the selector of a BLLM be wired as a real circuit,

which is what the experiments of Lappalainen and coauthors show to be informative in the fly, where networks constrained by the connectome predict the activity of the visual system neuron by neuron [111].

2. The fly: 79 neuropils

At a threshold of five synapses per connection the Fly-Wire brain has 134,013 neurons, 2,710,038 connections and 31,578,726 synapses in 79 neuropils. Give each neuron the neuropil where it receives most of its input, and count every synapse from the neuropil of its presynaptic neuron to the neuropil where it lies. The resulting matrix (Fig. 30(a)) has 59.0% of its weight on the diagonal: most synapses are made where the presynaptic neuron itself listens, a local and recurrent integration. The off-diagonal weight connects the stages of Fig. 28(a): the optic lobes and antennal lobes, which receive the senses, project to the mushroom bodies, the lateral horn and the central complex, which integrate and learn, and these to the premotor neuropils and the gnathal ganglion, where the descending neurons leave for the body. Acetylcholine, excitatory in the fly, is the predicted transmitter of 57.1% of the synapses, GABA of 24.7% and glutamate, mostly inhibitory, of 16.4%; serotonin, dopamine and octopamine together of 1.8%, concentrated in the mushroom bodies where dopamine carries the reward signal of associative learning. The numbers of partners are heavy-tailed: the median neuron has 11 output partners and 10 input partners, the mean about 19, and the largest 6,398 and 5,066 (Fig. 30(c)).

3. The human cortex: six layers

H01 is a fragment of human temporal cortex of about a cubic millimetre, reconstructed by electron microscopy at nanometre resolution [160]. Its table of somas lists 49,379 cells: 20,139 oligodendrocytes, 8,803 pyramidal neurons, 6,536 microglia and precursor cells, 5,474 astrocytes and 4,688 interneurons, among others, with 16,253 neurons in all. The layers are its stages (Fig. 31). Layer 1 holds almost no neurons (2% of its cells are pyramidal); pyramidal neurons are 40% of the cells of layer 2, 25% of layer 3, 33% of layer 4, 18% of layer 5 and 16% of layer 6, interneurons are densest in layer 2 (24%), and the white matter below is 77% oligodendrocytes, the insulation of the axons that leave. In the canonical circuit of the neocortex [59], input from the thalamus arrives in layer 4, is integrated in the recurrent networks of layers 2 and 3, and leaves from the large pyramidal neurons of layer 5, with feedback from layer 6 to the thalamus: the plan of Fig. 28(a) again.

4. From wiring to a BLLM

The three stages carry over with a precise meaning. The *sensing* stage is the set of proposers, whose profile q is what Theorem I measures. The *integrating* stage is the selector,

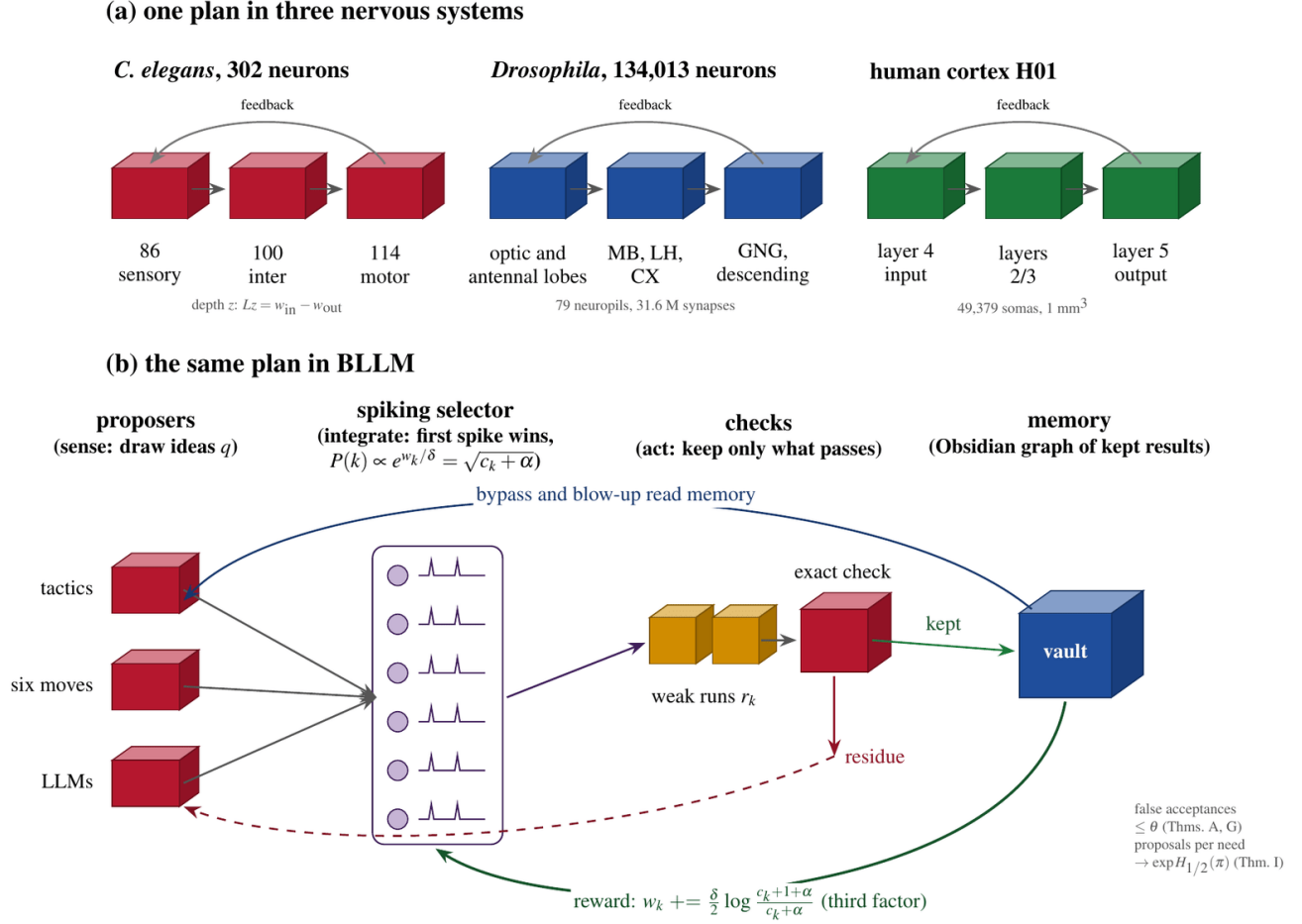


FIG. 28. One plan in three nervous systems and in a BLLM. (a) Sensory, integrating and output stages, with feedback, in the worm [45], the fly [61] and the human cortex [160]; the numbers are those of the reconstructions. (b) The same plan in a BLLM: proposers (language models, the six moves of Section XIC, tactic search) draw ideas from a profile q ; the spiking selector of Section XIID chooses who proposes next by a race to the first spike; weak runs and one exact check keep only what passes (Theorem XI.1); kept results enter the memory, which bypass and blow-up read, and send the reward (13) back to the selector; rejected candidates leave residues for the proposers. The checks bound the expected number of false acceptances by θ (Theorems A and G), and the selector brings the number of proposals per need to $e^{H_{1/2}(\pi)}$ (Theorem I).

whose job, like that of the basal ganglia in the account of Redgrave, Prescott and Gurney [146], is to choose one action among competing ones and to learn from a reward signal which choices pay, the role that dopamine plays in the account of Schultz, Dayan and Montague [154]; Proposition XII.2 shows that with a race to the first spike this choice is square-root sampling, the optimum of the gap law. The *acting* stage is the cascade of checks, which lets through only what passes, and the *feedback* is the memory and the residues. One part of the plan of a brain has no counterpart: no stage of a nervous system certifies what it keeps, and that is the part this paper adds. A BLLM is therefore not a model of a brain. It is a system that keeps the organisation which brains share, replaces the stage that decides by a check whose cost is priced by Theorems A to H, and uses at the stage of choice the mechanism that the gap law shows to be optimal. Because the selector is a circuit of leaky integrate-and-fire neurons with a local learn-

ing rule, it can run on neuromorphic hardware [63, 117]; we have run it only in simulation.

Appendix F: Growth laws and the price of a sound takeoff

1. Osgood takeoffs

Theorems D and E take the costs a_n as given. Growth models of an intelligence explosion produce them. The simplest such model, and the one behind most informal arguments, is the equation

$$\frac{dI}{dt} = cI^\alpha, \quad c > 0, \alpha > 0,$$

for a capability I that speeds up its own improvement. Its solution from $I(0) = 1$ reaches infinity at the finite time

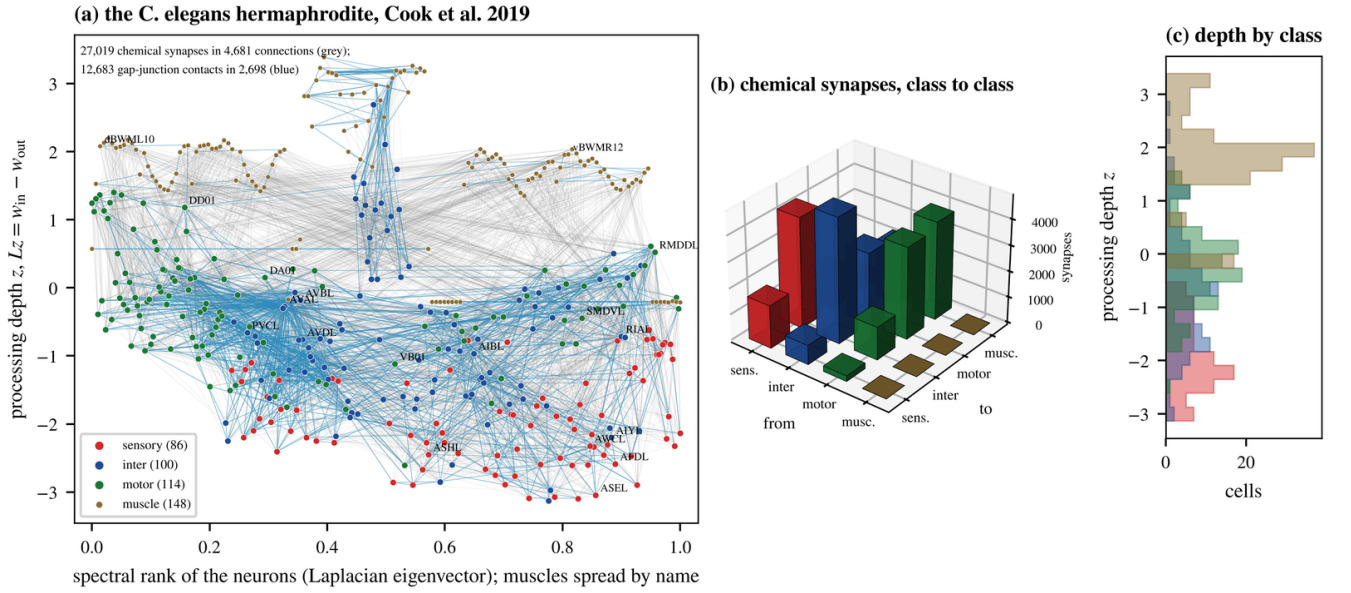


FIG. 29. The *C. elegans* hermaphrodite [45]. (a) Every neuron and muscle at its processing depth z (Lemma E.1) against its rank in a spectral coordinate of the neurons, with chemical connections in grey and gap junctions in blue; muscles, which have no spectral coordinate, are spread by name. Sensory neurons lie lowest and muscles highest. (b) Chemical synapses from class to class: the largest flows are from sensory neurons to interneurons, from interneurons to interneurons and motor neurons, and from motor neurons to muscles. (c) The distribution of depths in each class.

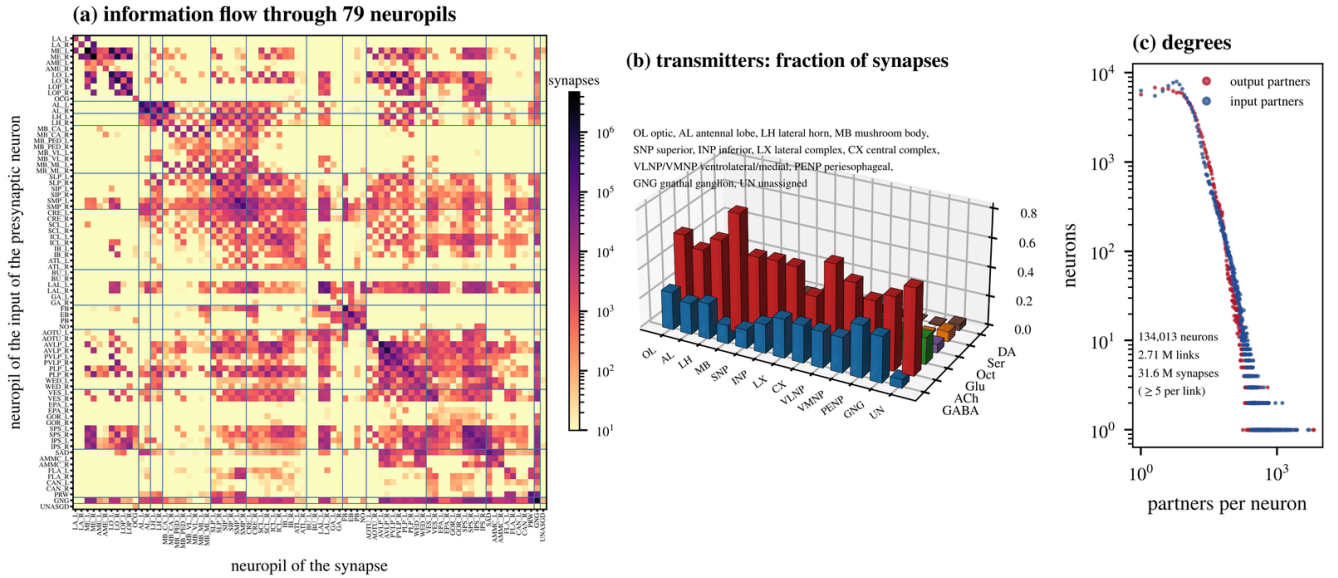


FIG. 30. The adult *Drosophila* brain, FlyWire release 783, connections of at least five synapses [61]. (a) Synapses from the neuropil where the presynaptic neuron receives most of its input (rows) to the neuropil of the synapse (columns), on a logarithmic scale, with the neuropils grouped into the super-groups of FlyWire; 59% of the weight is on the diagonal. (b) The predicted transmitter of the synapses [67] in each super-group. (c) The numbers of output and input partners per neuron.

$\int_1^\infty dI/(cI^\alpha) = 1/(c(\alpha - 1))$ when $\alpha > 1$ and grows without blowing up when $\alpha \leq 1$; Osgood's criterion states the same

for $dI/dt = f(I)$ with the integral of $1/f$ [131].¹¹ To put a check on the curve one has to say what one improvement is.

¹¹ W. F. Osgood, *Beweis der Existenz einer Lösung der Differentialgleichung $dy/dx = f(x, y)$ ohne Hinzunahme der Cauchy-Lipschitzschen Bedingung*,

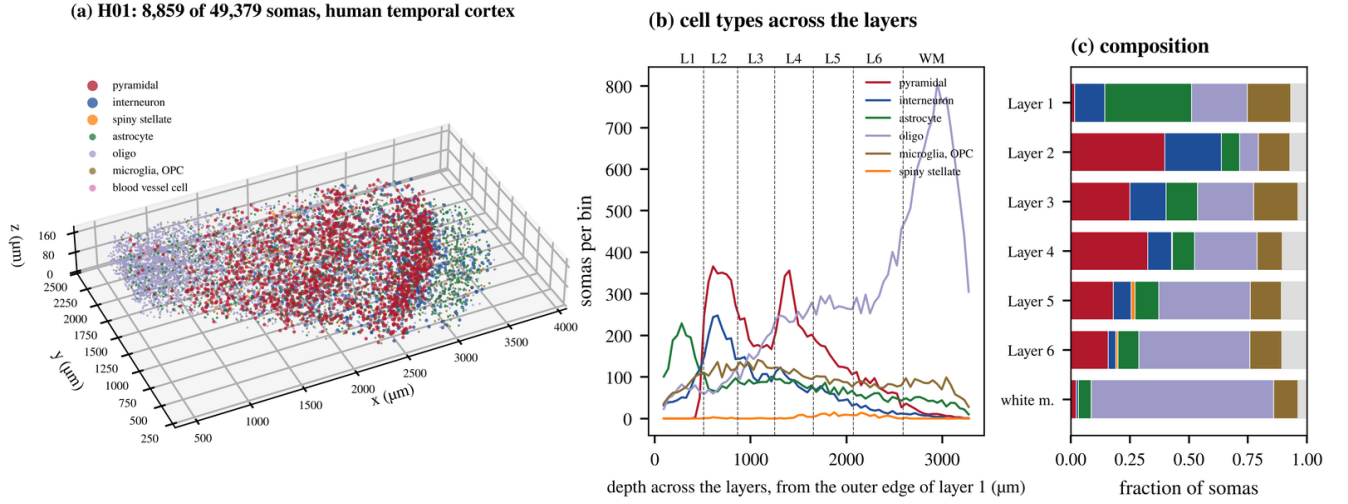


FIG. 31. The human cortical fragment H01 [160]. (a) A random sample of 8,859 of its 49,379 somas at their positions, coloured by cell type. (b) The number of somas of each type along the axis of the layers, measured from the outer edge of layer 1, with the boundaries between layers. (c) The composition of each layer and of the white matter.

Two choices are natural.

Additive steps. The n -th improvement raises the capability from n to $n+1$ at the rate reached at its start, so it takes the time $g_n = 1/(cn^\alpha)$.

Multiplicative steps. The n -th improvement multiplies the capability by a fixed factor $\lambda > 1$, from λ^{n-1} to λ^n , at the rate reached at its start, so it takes the time $g_n = (\lambda - 1)\lambda^{n-1}/(c\lambda^{(n-1)\alpha}) = \frac{\lambda-1}{c}q^{n-1}$ with $q = \lambda^{-(\alpha-1)}$.

In both cases we let the check of the n -th improvement cost a fixed fraction $\kappa > 0$ of the time it took to make, $a_n = \kappa g_n$ per run, so that faster improvements are also faster to check. Write $G = \sum_n g_n$ for the unchecked takeoff time and $T_\theta = G + V^*(\theta)$ for the least time of a takeoff with at most θ expected false acceptances, measured, as in Theorem A, in the run in which every candidate is correct.

Proposition F.1 (Osgood takeoffs). *Let $\alpha > 1$, $0 < \theta \leq 1$ and $0 < \varepsilon < 1$.*

(i) *In both models*

$$\begin{aligned} \frac{T_\theta}{G} &\geq 1 + \frac{\kappa}{L} \left(H + \log \frac{1}{\theta} \right), \\ \frac{T_\theta}{G} &\leq 1 + \frac{\kappa}{L} \left(H + \log \frac{1}{\theta} \right) + \kappa, \end{aligned}$$

where H is the entropy of the profile g/G .

(ii) *For additive steps $G = \zeta(\alpha)/c$, which lies between $1/(c(\alpha-1))$ and $\alpha/(c(\alpha-1))$, and, as $\alpha \rightarrow 1^+$,*

$$H = \frac{1}{\alpha-1} + \log \frac{1}{\alpha-1} + O(1).$$

Hence the least check time of a sound takeoff is

$$T_\theta - G = \frac{\kappa(1 + O((\alpha-1)\log \frac{1}{\alpha-1}))}{c(\alpha-1)^2 L}.$$

(iii) *For multiplicative steps $G = (\lambda-1)/(c(1-q))$ and*

$$\begin{aligned} H &= \log \frac{1}{1-q} + \frac{q}{1-q} \log \frac{1}{q} \\ &= \log \frac{1}{(\alpha-1)\log \lambda} + 1 + o(1) \quad (\alpha \rightarrow 1^+). \end{aligned}$$

Hence

$$T_\theta - G = \frac{\kappa}{c(\alpha-1)L} \cdot \frac{\lambda-1}{\log \lambda} \left(\log \frac{1}{\alpha-1} + O(1) \right).$$

(iv) *With checks of polynomial order s , a sound takeoff of finite cost exists for additive steps if and only if $\alpha > 1+s$, and then its least check cost is $\frac{\kappa}{c}\theta^{-s}\zeta(\frac{\alpha}{1+s})^{1+s}$; for multiplicative steps it exists for every $s > 0$, with least check cost $\kappa G\theta^{-s}(1-q)/(1-q^{1/(1+s)})^{1+s}$.*

Proof. (i) The profile of the checks is $a/A = g/G$ and $A = \kappa G$. Proposition V.7 gives $\frac{A}{L}(H + \log \frac{1}{\theta}) \leq V^*(\theta) \leq \frac{A}{L}(H + \log \frac{1}{\theta}) + A$; divide by G .

(ii) $G = \sum_n n^{-\alpha}/c = \zeta(\alpha)/c$, and $1/(\alpha-1) \leq \zeta(\alpha) \leq 1 + 1/(\alpha-1)$ by comparison with the integral, as in the proof of Proposition IX.3, which also gives the expansion

Monatshefte für Mathematik und Physik 9 (1898), 331–345 [131]. The paper is about uniqueness; the blow-up criterion that bears Osgood's name is the elementary fact, implicit there, that a solution of $y' = f(y)$ with $f > 0$ reaches infinity in the finite time $\int_{y_0}^{\infty} dy/f(y)$ exactly when the integral converges. For $f(y) = cy^\alpha$ the integral is $y_0^{1-\alpha}/(c(\alpha-1))$ for $\alpha > 1$ and diverges for $\alpha \leq 1$.

of H . Then $T_\theta - G = \kappa G(H + \log \frac{1}{\theta} + O(L))/L$ by (i), and $\kappa GH = \frac{\kappa}{c}(\frac{1}{\alpha-1} + O(1))(\frac{1}{\alpha-1} + O(\log \frac{1}{\alpha-1}))$.

(iii) $G = \frac{\lambda-1}{c} \sum_{n \geq 1} q^{n-1} = \frac{\lambda-1}{c(1-q)}$. The profile is geometric, $p_n = (1-q)q^{n-1}$, so $\log(1/p_n) = \log \frac{1}{1-q} + (n-1) \log \frac{1}{q}$ and $\sum_n (n-1)p_n = q/(1-q)$; this is the formula for H . As $\alpha \rightarrow 1^+$, $u = (\alpha-1) \log \lambda \rightarrow 0$ and $q = e^{-u}$, so $1-q = u + O(u^2)$ and $\frac{q}{1-q} \log \frac{1}{q} = \frac{ue^{-u}}{1-e^{-u}} \rightarrow 1$; and $G = \frac{\lambda-1}{cu}(1 + O(u))$. Combine with (i).

(iv) By Corollary VII.5(ii) a sound allocation of finite cost exists if and only if $\sum_n a_n^{1/(1+s)} < \infty$. For additive steps this is $\sum_n n^{-\alpha/(1+s)} < \infty$, that is $\alpha > 1+s$; for multiplicative steps it is a geometric series. For the value, let $\beta = 1/(1+s)$ and $S = \sum_n a_n^\beta < \infty$. For every allocation with $\sum_n \delta_n \leq \theta$ and every N , the lower bound in the proof of Theorem C, applied to the first N checks, gives $\sum_{n \leq N} a_n \delta_n^{-s} \geq \theta^{-s} (\sum_{n \leq N} a_n^\beta)^{1+s}$; letting $N \rightarrow \infty$, every sound allocation costs at least $\theta^{-s} S^{1+s}$, and the allocation $\delta_n = \theta a_n^\beta / S$ attains it, by the computation of the attainment step there. For additive steps $S = (\kappa/c)^\beta \zeta(\alpha\beta)$, and for multiplicative steps $S = A^\beta \sum_n p_n^\beta = A^\beta (1-q)^\beta / (1-q^\beta)$, with $A = \kappa G$. $\square$

So the takeoff can be made sound whenever it is singular, and the price is the factor $\frac{\kappa}{L}(H + \log \frac{1}{\theta})$ on the time. Near the critical exponent the two counts of improvements part ways (Fig. 32). If every unit of capability is a separate modification that must be checked, the entropy of the profile is of order $1/(\alpha-1)$, and the checks cost a factor $1/(\alpha-1)$ more than the improvements, $\kappa/(c(\alpha-1)^2 L)$ in all. If every doubling of capability is one modification, the entropy is only $\log \frac{1}{\alpha-1}$. At $\alpha = 1.01$, $\varepsilon = \frac{1}{2}$ and $\theta = 10^{-6}$ the lower bound of (i) gives a slowdown factor $1 + 171.5 \kappa$ for additive steps and $1 + 28.6 \kappa$ for doublings; at $\alpha = 2$ the two factors are $1 + 22.3 \kappa$ and $1 + 21.9 \kappa$, and the budget term $\log_2 10^6 \approx 19.9$ dominates. With Chebyshev checks, $s = 1$, additive steps need $\alpha > 2$, and at $\alpha = 3$ the least check cost is $\frac{\kappa}{c} \zeta(\frac{3}{2})^2 \cdot 10^6 \approx 6.8 \times 10^6 \frac{\kappa}{c}$, five orders of magnitude above the $25.1 \frac{\kappa}{c}$ of exponential checks.

2. What the growth exponent is, and what it is not

The exponent α of Proposition F.1 is the elasticity of the speed of research with respect to the capability of the system that does it. It is not one of the exponents of the empirical scaling laws of language models. Kaplan and coauthors fitted the test loss of a language model trained with the optimal allocation of a compute budget C to a power law $C^{-\alpha_C}$ with $\alpha_C \approx 0.050$ [100], and Hoffmann and coauthors fitted the loss as $E + A_0 N^{-0.34} + B_0 D^{-0.28}$ in the number N of parameters and the number D of training tokens [86].¹² These laws relate

loss to resources. To turn them into a value of α one needs two further maps, from loss to research ability and from research ability to the speed of the next improvement, and neither has been measured. Anbar Jafari, Ozczinar and Anbarjafari propose to estimate how fast improvement accelerates relative to its current level from series that engineers already record, which is an estimate of α [6].

Ord observes that a singular growth path needs the generation time, the time around the loop of improvement, to tend to zero [128]. In the present model this is the condition $g_n \rightarrow 0$, which a convergent $\sum g_n$ forces. Theorem D adds a second clock: the time spent checking the n -th improvement must also tend to zero, and faster than the generation time can afford to, because the sum of $a_n \log(1/a_n)$, and not only of a_n , must converge. Proposition F.1 shows that for power laws this second condition is free in kind and expensive in amount: every Osgood takeoff with $\alpha > 1$ can be made sound with exponential checks, but near the threshold the checks take longer than the improvements by a factor that diverges as $1/(\alpha-1)$ when every unit of capability is checked. With checks that are only polynomially reliable, the threshold for a sound takeoff moves from $\alpha = 1$ to $\alpha = 1+s$, and between the two lies a range of singular takeoffs that cannot be made sound at any cost.

Remark F.2 (Errors of the generator). Proposition F.1 prices the run in which every candidate is correct. If the n -th candidate is incorrect with probability in $[\rho_*, \bar{\rho}] \subset (0, 1)$ and incorrect candidates are regenerated, the proof of Theorem F, with Proposition V.7 in place of Theorem A, applies to the infinite chain: the lower bound holds with θ replaced by θ/ρ_* , and the upper bound with θ replaced by $\theta(1-\bar{\rho})/\bar{\rho}$ and the whole multiplied by $1/(1-\bar{\rho})$. The orders in $\alpha-1$ of parts (ii) and (iii) do not change.

Appendix G: Formal proofs of hard problems

Several of the hardest problems of mathematics now have proofs that a machine has checked line by line, and four of these formal proofs were written with the help of artificial intelligence [5, 85, 124, 190]. They are the natural test of the theory: each is a long chain of steps, each step has a check, and each records which steps were hard. This section reads six of them in the language of Sections VIII and XI, states what a BLLM adds and what it does not, and describes the package that implements the system.

Both fits measure the cross-entropy loss on held-out text. The exponent $\alpha_C \approx 0.050$ is that of the compute-efficient frontier in the first paper; the exponents 0.34 and 0.28, with fitted constants $E \approx 1.69$, $A_0 \approx 406.4$ and $B_0 \approx 410.7$, are those of the parametric fit of the second. The letters A_0 and B_0 are renamed here to avoid a clash with the total cost A .

¹² J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu and D. Amodei, *Scaling laws for neural language models*, arXiv:2001.08361 (2020) [100]; J. Hoffmann et al., *Training compute-optimal large language models*, arXiv:2203.15556 (2022) [86].

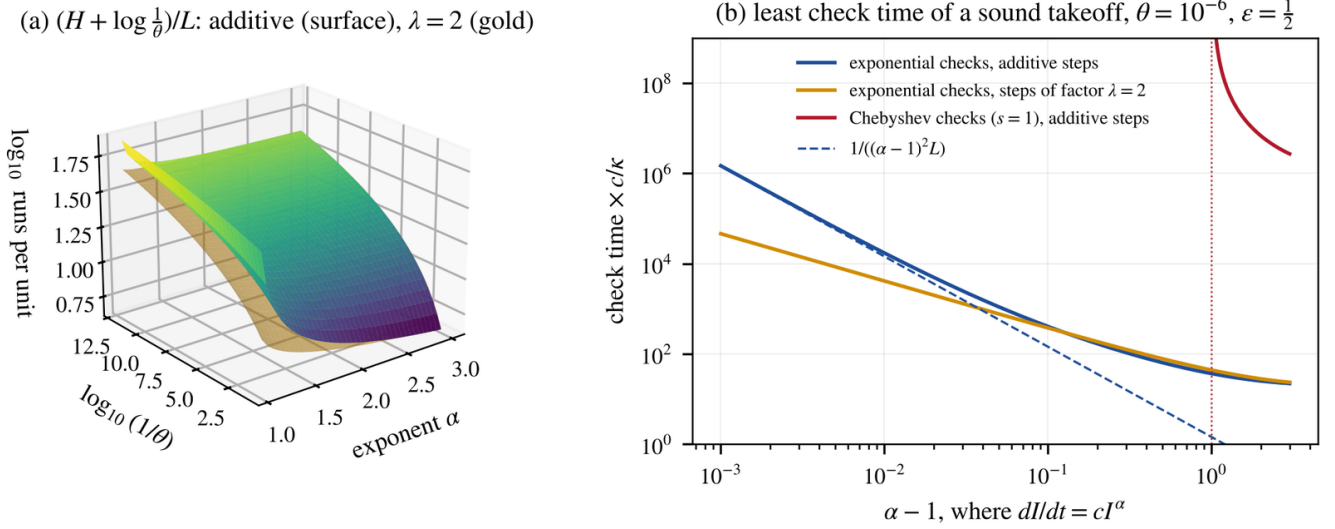


FIG. 32. Proposition F.1, with $\varepsilon = \frac{1}{2}$. (a) The lower bound $(H + \log \frac{1}{\theta})/L$ for the number of runs per unit of check cost, on a logarithmic scale, as a function of the growth exponent α and of $\log_{10}(1/\theta)$, for additive steps (coloured surface) and for doublings, $\lambda = 2$ (gold surface). The additive surface rises like $1/(\alpha - 1)$ as $\alpha \rightarrow 1$, the multiplicative one like $\log \frac{1}{\alpha - 1}$. (b) The lower bound for the least check time of a sound takeoff, in units of κ/c , at $\theta = 10^{-6}$, against $\alpha - 1$ on logarithmic axes: exponential checks with additive steps (blue), which approach $1/((\alpha - 1)^2 L)$ (dashed), exponential checks with doublings (gold), and Chebyshev checks with additive steps (red), which cost infinitely much for $\alpha \leq 2$ (dotted line).

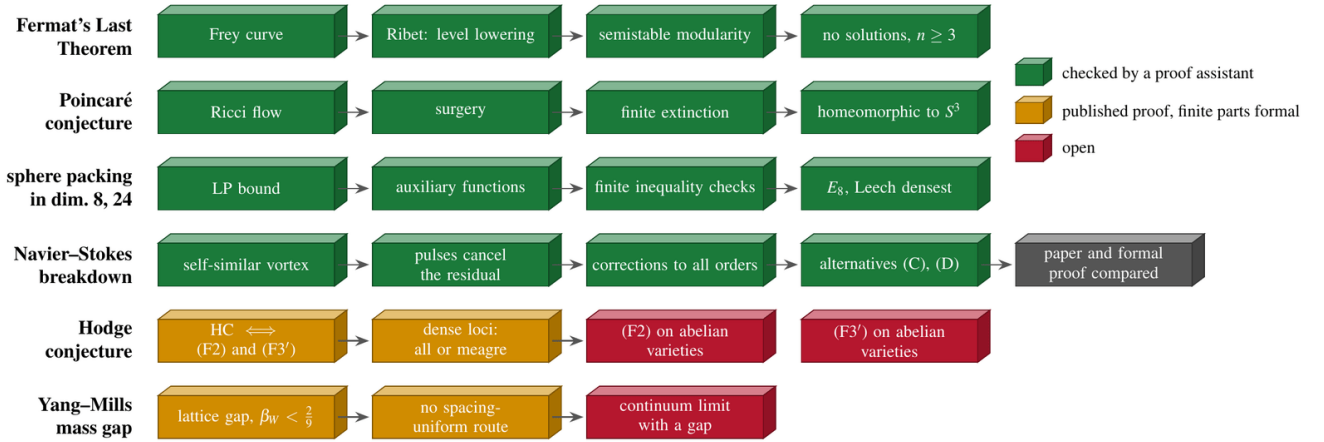


FIG. 33. The main steps of the six proofs of Table XXIII. Green steps are checked by a proof assistant, gold steps have published proofs whose formal checks cover only finite parts, red steps are open, and grey marks the comparison of the Navier–Stokes paper with its formal proof [20]. The two Hodge statements (F2) and (F3') are independent, and both are needed.

1. Six problems

Table XXIII and Fig. 33 list the problems, the formal results and the steps that remain. Four are settled: Fermat's Last Theorem by Wiles and by Taylor and Wiles [173, 184], formalised in Lean 4 in September 2026 by dozens of Claude agents in eleven days on the Prove2Me platform, along the simplified proof of Darmon, Diamond and Taylor, in about thirteen million lines with 29,500 theorems in the final proof and only the three standard axioms [5, 50, 57]; the Poincaré conjecture by Perelman, now formalised in Lean 4 with Mathlib follow-

ing the account of Morgan and Tian [190]; the sphere packing problem in dimensions 8 and 24 [47, 178], formalised by Harisharan, Birkbeck, Lee, Ma, Mehta, Poiroux and Viazovska and completed by Math, Inc. [85]; and finite-time breakdown of the three-dimensional Navier–Stokes equations with forcing, alternatives (C) and (D) of the official description of the Millennium problem, proved by OpenAI for every positive viscosity in a paper of 165 pages and in Lean [124]. Each formal development is archived with a record of an independent build [8–10]. Two are open: the Hodge conjecture, which the first two authors reduced to two statements (F2) and (F3') about abelian varieties [17], and the existence of Yang–Mills theory

with a mass gap, for which they proved a lattice mass gap for four-dimensional $SU(2)$ up to the coupling $\beta_W = \frac{2}{9}$, the limit of single-link Dobrushin arguments [18].¹³

The history of the problems contains the moves of Section XI C. Wiles announced a proof in 1993 in which one step failed; the repair with Taylor replaced it by a different argument [173], a residual move on a single step of a long chain. The Navier–Stokes construction of [124] builds a self-similar vortex, cancels the singular part of its momentum residual by oscillatory pulses and corrects what is left to every order: construction, residual and blow-up in one argument. The sphere packing proofs reduce optimality to the existence of auxiliary functions whose inequalities are checked by finite computations, a trivialization. The Hodge reduction replaces the conjecture by statements on abelian varieties through known theorems, a bypass, and weakens one of them to (F3'), a step down the ladder of strong then weak.

2. What a BLLM adds, and what it does not

Three of the theorems of this paper speak directly to these proofs.

First, Theorem VIII.2. A formal development checks every step of the formal argument, not the steps of the paper it formalises. Bastounis, Circelli and Hansen argue that the Lean proof of [124] does not follow the argument of the paper, and that a faithful translation of a natural-language proof is not computable in general [20]. Whatever the outcome of that discussion, a natural-language argument whose only certificate is a formal proof of its final statement is checked at its outcome, and Theorem VIII.2 shows that certification at the outcome is exponentially more expensive in the number of steps than certification step by step. A BLLM that keeps the paper and the formal proof aligned keeps a check at every step of both.

Second, Theorem XI.1. The developments above contain tens of thousands of formal lemmas, and the strong check, a full build, is expensive. A system that proposes most steps wrongly and filters them with cheap checks pays the build essentially only on its correct proposals, with a premium logarithmic in the ratio of costs; this is what makes very large numbers of machine proposals affordable without any loss of soundness.

Third, Theorem G and Corollary X.1. A system that keeps working on an open problem does not know how many steps

it will take or what they will cost. The online law lets it run indefinitely with a guaranteed bound on the expected number of false acceptances over its whole life, at a cost within a double logarithm of the best schedule chosen with hindsight.

What the theorems do not give is the step that no proposer produces. The open entries of Table XXIII, (F2), (F3') and the continuum limit with a gap, are statements for which no proof has been proposed, by people or by machines. A BLLM cannot make them easier to prove: its guarantee is that whatever it keeps has been checked, at the least possible cost, and that its record of what failed, the residues, is complete. That record is the part of the work a later proposer, human or artificial, can build on.

3. Five strategies of mathematicians as proposers

The six moves of Section XI C describe what a step does. The proofs above also show five larger strategies, each a way of choosing what to propose, and the package implements each of them as a proposer whose candidates meet the same checks.

- (i) *Certificate first* (Cohn and Elkies, Viazovska). Replace the statement by the existence of one explicit object whose properties imply it: Cohn and Elkies reduced the optimality of a sphere packing to an auxiliary function with sign conditions on it and on its Fourier transform [40], and Viazovska built that function in dimension 8 from quasimodular and weakly holomorphic modular forms, after which the sign conditions are finite inequalities [47, 178]; the Lean development checks them [10, 85]. The proposer searches a structured space for the certificate and hands the finite inequalities to the checks.
- (ii) *Flow with surgery* (Perelman). Evolve the object by a monotone flow, classify the limits at its singularities and cut them out [133–135]. On a lattice gauge theory the corresponding flow is the gradient flow of the Wilson action [113].
- (iii) *Lift from a residual case* (Wiles). Prove the statement in a small case and propagate it through a deformation argument, as modularity was lifted from residual representations [173, 184]; for Hodge classes the small case is a CM point, and the propagation is along the dense loci of [17].
- (iv) *Infinite descent* (Fermat). Show that a least counterexample yields a smaller one [182]; the proposer looks for the reduction map and the checks verify that it decreases a well-founded measure.
- (v) *Physics first* (Witten). Let a quantum field theory predict the answer and then prove it, as Chern–Simons theory predicted the Jones polynomial [185] and electric–magnetic duality the mechanism of confinement in a supersymmetric gauge theory [164].

None of the five changes a theorem of this paper: they change the profile q of the proposer, so their value is measured by

¹³ D. Bhattacharjee and U. Bhattacharya, *Dense algebraic loci of Hodge classes*, Hexagon preprint hexagon:2610.00315v1 (17 October 2026), with Lean 4 archive doi:10.5281/zenodo.23439578 [17]. The paper proves that the Hodge conjecture is equivalent to two statements about abelian varieties, that the locus on which either holds is dense in every family and is either the whole family or a meagre set, builds for each statement the object it needs next and proves at which step it stops. The second work, *Lattice Yang–Mills theory: spacing-uniform obstructions, the mass gap up to the Dobrushin ceiling, and Lean certificate*, Zenodo (2026), doi:10.5281/zenodo.23449466 [18], also proves that lattice inequalities uniform in the spacing cannot produce the continuum mass gap, and isolates the remaining step as one hypothesis on the ratio of the lattice gap to the spacing.

TABLE XXIII. Six problems with machine-checked or published proofs, in the language of the moves of Section XIC. “Formal” names the proof checker of the archived development; the last column names the moves on which the proof rests.

Problem	What is proved	Formal	What remains	Moves
Fermat’s Last Theorem	modularity of semistable elliptic curves, hence the theorem [5, 173, 184]	Lean 4	nothing	bypass (Frey, Ribet), residual (Taylor–Wiles)
Poincaré conjecture	Hamilton–Perelman Ricci flow with surgery [190]	Lean 4	nothing	construction, residual (surgery)
Sphere packing in dimensions 8, 24	E_8 and the Leech lattice are densest [47, 85, 178]	Lean 4	nothing in dimensions 8, 24	construction, trivialization
Navier–Stokes breakdown	alternatives (C), (D) for every viscosity [124]	Lean 4	the correspondence of paper and formal proof [20]	construction, residual, blow-up
Hodge conjecture	equivalence with (F2) and (F3’); dense loci [17]	Lean 4, finite parts	(F2), (F3’)	bypass, strong then weak
Yang–Mills mass gap	lattice gap for $\beta_W < \frac{2}{9}$; no spacing-uniform route [18]	Lean 4, finite parts	continuum limit with a gap	construction, strong then weak, residual

Theorem I, and whatever they propose is kept only after the checks.

ACKNOWLEDGEMENTS

The paper builds on Claude Shannon’s coding theorem, David Huffman’s algorithm, the inequalities of Leon Kraft and Brockway McMillan, Alfréd Rényi’s entropies and L. L. Campbell’s coding theorem for them, the branching processes of Watson, Galton, Otter, Kesten and Stigum, and the Borel–Cantelli lemmas. The recent analyses of verifiers by Jia, Rakhlin and Xie, by Rohatgi and coauthors and by Balcan and coauthors, and Toby Ord’s study of the dynamics of intelligence explosions, fixed the questions it answers. We thank the developers of Lean and Mathlib.

DATA AND CODE AVAILABILITY

Everything the paper uses is in the repository github.com/creelie/eBLLM, the package of the engine Anaxia (command `anaxia`), released as version 3.0.0 and archived

at Zenodo under [doi:10.5281/zenodo.23464390](https://doi.org/10.5281/zenodo.23464390) [24]. The folder `paper/bllm` holds the $\text{\LaTeX}$ source, the figure programs with their data, the Lean 4 development of Appendix B, and the Python, C, C++ and shell programs of the numerical checks; the compiled paper is the file `preprintBLLM/BLLM.pdf` of the repository. The package contains the crossing network and its controls, the benchmark of Section XVI with its trees and the logs of every run, construction mode with the Steinberg gadgets of Section XVIII A, the layer towers and the bench of Section XVIII B, the per-round records of the Hodge round of Section XIX, the exact and weighted-poset searches for Erdős problem 1183 and the colourings of Theorem XVII.2, and the registry of Appendix D. A container image of the engine is published as `ghcr.io/creelie/bllm-engine`; it contains no model weights. Earlier versions of the paper are in the folder BLLM of github.com/creelie/D7. Deep Bhattacharjee carried out the computations and the Lean, C, C++, Python and shell verifications with the help of AI assistance (Claude, Anthropic). Both authors approved the final version.

ETHICAL DECLARATIONS

Funding: none. Competing interests: none.

-
- [1] T. Achim, A. Best, A. Bietti, K. Der et al., *Aristotle: IMO-level automated theorem proving*, preprint (2025). [arXiv:2510.01346](https://arxiv.org/abs/2510.01346).
 - [2] G. Ancona, M. Cavicchi, R. Laterveer and G. Saccà, *Relative and absolute Lefschetz standard conjectures for some Lagrangian fibrations*, J. Lond. Math. Soc. (2) **111** (2025), e70133. [doi:10.1112/jlms.70133](https://doi.org/10.1112/jlms.70133).
 - [3] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, *Proof verification and the hardness of approximation problems*, J. ACM **45** (1998), no. 3, 501–555. [doi:10.1145/278298.278306](https://doi.org/10.1145/278298.278306).
 - [4] K. B. Athreya and P. E. Ney, *Branching Processes*, Grundlehren der mathematischen Wissenschaften 196, Springer, Berlin, 1972. [doi:10.1007/978-3-642-65371-1](https://doi.org/10.1007/978-3-642-65371-1).
 - [5] Anthropic, *Formalizing Fermat’s Last Theorem*, research report (4 September 2026). <https://www.anthropic.com/research/formalizing-fermats-last-theorem>.
 - [6] A. Anbar Jafari, C. Ozcinar and G. Anbarjafari, *A mathematical framework for AI singularity: conditions, bounds, and control of recursive improvement*, preprint (2025). [arXiv:2511.10668](https://arxiv.org/abs/2511.10668).
 - [7] D. Arapura, *Motivation for Hodge cycles*, Adv. Math. **207** (2006), 762–781. [doi:10.1016/j.aim.2006.01.005](https://doi.org/10.1016/j.aim.2006.01.005).
 - [8] OpenAI, *Finite-time blowup for Navier–Stokes: OpenAI’s Lean proof, built and checked*, Zenodo archive, version 1.0.0 (2026). [doi:10.5281/zenodo.23286556](https://doi.org/10.5281/zenodo.23286556).
 - [9] Z. Zhang, A. Delaval, L. Chen, J. Chen, J. Xu, Y. Liao, J. Jiang, C. Liu and B. Dong, *The Poincaré conjecture in Lean 4: the frenzmath formalization, built and checked*, Zenodo archive, version 1.0.0 (2026). [doi:10.5281/zenodo.23288534](https://doi.org/10.5281/zenodo.23288534).

- [10] S. Hariharan, C. Birkbeck, S. Lee, H. K. G. Ma, B. Mehta, A. Poiroux, M. Viazovska and Math, Inc., *Sphere packing in dimensions 8, 24 and 3: an independent check of the formal proofs*, Zenodo archive, version 1.0.0 (2026). [doi:10.5281/zenodo.23289869](https://doi.org/10.5281/zenodo.23289869).
- [11] S. Arora and S. Safra, *Probabilistic checking of proofs: a new characterization of NP*, J. ACM **45** (1998), no. 1, 70–122. [doi:10.1145/273865.273901](https://doi.org/10.1145/273865.273901).
- [12] S. Aaronson and A. Wigderson, *Algebrization: a new barrier in complexity theory*, ACM Trans. Comput. Theory **1** (2009), 2:1–2:54. [doi:10.1145/1490270.1490272](https://doi.org/10.1145/1490270.1490272).
- [13] L. Babai, *Trading group theory for randomness*, in Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing (STOC '85), ACM, New York, 1985, 421–429. [doi:10.1145/22145.22192](https://doi.org/10.1145/22145.22192).
- [14] T. Balaban, *Renormalization group approach to lattice gauge field theories. I. Generation of effective actions in a small field approximation and a coupling constant renormalization in four dimensions*, Comm. Math. Phys. **109** (1987), 249–301. [doi:10.1007/BF01215223](https://doi.org/10.1007/BF01215223).
- [15] G. G. Batrouni, *Plaquette formulation and the Bianchi identity for lattice gauge theories*, Nuclear Phys. B **208** (1982), 467–483. [doi:10.1016/0550-3213\(82\)90231-0](https://doi.org/10.1016/0550-3213(82)90231-0).
- [16] R. Bauerschmidt and T. Bodineau, *Log-Sobolev inequality for the continuum sine-Gordon model*, Comm. Pure Appl. Math. **74** (2021), 2064–2113. [doi:10.1002/cpa.21926](https://doi.org/10.1002/cpa.21926).
- [17] D. Bhattacharjee and U. Bhattacharya, *Dense algebraic loci of Hodge classes*, Hexagon preprint (2026), hexagon:2610.00315v1, hexagonmath.org/2610.00315v1; source and Lean 4 archive, version 2.0.0, [doi:10.5281/zenodo.23461752](https://doi.org/10.5281/zenodo.23461752).
- [18] D. Bhattacharjee and U. Bhattacharya, *Lattice Yang–Mills theory: spacing-uniform obstructions, the mass gap up to the Dobrushin ceiling, and Lean certificate*, Zenodo, version 2.0.0 (2026). [doi:10.5281/zenodo.23449466](https://doi.org/10.5281/zenodo.23449466).
- [19] M.-F. Balcan, A. Blum, K. Fragkia, Z. Li and D. Sharma, *Verify to amplify: improving reasoning via learned chain-of-thought verification*, preprint (2026). [arXiv:2603.03538](https://arxiv.org/abs/2603.03538).
- [20] A. Bastounis, F. Circelli and A. C. Hansen, *Navier–Stokes lost in translation: why Lean verification of AI autoformalisation does not guarantee correct natural language proofs*, preprint (2026). [arXiv:2610.08144](https://arxiv.org/abs/2610.08144).
- [21] O. V. Borodin, A. N. Glebov, A. Raspaud and M. R. Salavatipour, *Planar graphs without cycles of length from 4 to 7 are 3-colorable*, J. Combin. Theory Ser. B **93** (2005), 303–311. [doi:10.1016/j.jctb.2004.11.001](https://doi.org/10.1016/j.jctb.2004.11.001).
- [22] T. Baker, J. Gill and R. Solovay, *Relativizations of the $\mathcal{P} = ? \mathcal{N}$ question*, SIAM J. Comput. **4** (1975), 431–442. [doi:10.1137/0204037](https://doi.org/10.1137/0204037).
- [23] D. Bhattacharjee, *Synthetic neuromorphic superintelligence for verifier-gated conjecture closure: sand-tile automata, K-dot circuits, and phenomenological proof learning*, PhilArchive (2026). philarchive.org/rec/BHASNS-2.
- [24] D. Bhattacharjee, *Anaxia, the BLLM engine*, Python package eBLLM with the command `anaxia`, Lean 4 library and container image `ghcr.io/creelie/bllm-engine`, Zenodo, release 3.0.0 (2026), [doi:10.5281/zenodo.23464390](https://doi.org/10.5281/zenodo.23464390); all versions [doi:10.5281/zenodo.23286641](https://doi.org/10.5281/zenodo.23286641). github.com/creelie/eBLLM.
- [25] B. Brown, J. Juravsky, R. Ehrlich, R. Clark, Q. V. Le, C. Ré and A. Mirhoseini, *Large language monkeys: scaling inference compute with repeated sampling*, preprint (2024). [arXiv:2407.21787](https://arxiv.org/abs/2407.21787).
- [26] Y. Bai, S. Kadavath, S. Kundu, A. Askill, J. Kernion, A. Jones et al., *Constitutional AI: harmlessness from AI feedback*, preprint (2022). [arXiv:2212.08073](https://arxiv.org/abs/2212.08073).
- [27] E. Botta, Y. Li, A. Mehta, J. T. Ash, C. Zhang and A. Risteski, *On the query complexity of verifier-assisted language generation*, preprint (2025). [arXiv:2502.12123](https://arxiv.org/abs/2502.12123).
- [28] T. F. Bloom, *Erdős problem #1183*, erdosproblems.com/1183, accessed 18 October 2026.
- [29] D. Bhattacharjee, P. Mandal and U. Bhattacharya, *Resolving Erdős–Ulam monochromatic union-closed family conjectures*, preprint (2026), [arXiv:2610.02833](https://arxiv.org/abs/2610.02833); Lean 4 archive, version 5, [doi:10.5281/zenodo.23374973](https://doi.org/10.5281/zenodo.23374973).
- [30] D. Bhattacharjee, P. Mandal and U. Bhattacharya, *Algebraic loci of Weil classes on abelian varieties*, with verification code, Zenodo, version 5.0.1 (2026). [doi:10.5281/zenodo.23225483](https://doi.org/10.5281/zenodo.23225483).
- [31] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal et al., *Language models are few-shot learners*, Advances in Neural Information Processing Systems **33** (2020), 1877–1901. [arXiv:2005.14165](https://arxiv.org/abs/2005.14165).
- [32] É. Borel, *Les probabilités dénombrables et leurs applications arithmétiques*, Rend. Circ. Mat. Palermo **27** (1909), 247–271. [doi:10.1007/BF03019651](https://doi.org/10.1007/BF03019651).
- [33] N. Bostrom, *The superintelligent will: motivation and instrumental rationality in advanced artificial agents*, Minds and Machines **22** (2012), 71–85. [doi:10.1007/s11023-012-9281-3](https://doi.org/10.1007/s11023-012-9281-3).
- [34] N. Bostrom, *Superintelligence: Paths, Dangers, Strategies*, Oxford University Press, Oxford, 2014.
- [35] N. Bostrom, *How long before superintelligence?*, Int. J. Futures Studies **2** (1998); reprinted in Linguistic and Philosophical Investigations **5** (2006), no. 1, 11–30.
- [36] M. Bhargava and A. Shankar, *Binary quartic forms having bounded invariants, and the boundedness of the average rank of elliptic curves*, Ann. of Math. (2) **181** (2015), 191–242. [doi:10.4007/annals.2015.181.1.3](https://doi.org/10.4007/annals.2015.181.1.3).
- [37] M. Bhargava and A. Shankar, *Ternary cubic forms having bounded invariants, and the existence of a positive proportion of elliptic curves having rank 0*, Ann. of Math. (2) **181** (2015), 587–621. [doi:10.4007/annals.2015.181.2.4](https://doi.org/10.4007/annals.2015.181.2.4).
- [38] N. Buskin, *Every rational Hodge isometry between two K3 surfaces is algebraic*, J. Reine Angew. Math. **755** (2019), 127–150. [doi:10.1515/crelle-2017-0027](https://doi.org/10.1515/crelle-2017-0027).
- [39] L. L. Campbell, *A coding theorem and Rényi’s entropy*, Information and Control **8** (1965), no. 4, 423–429. [doi:10.1016/S0019-9958\(65\)90332-3](https://doi.org/10.1016/S0019-9958(65)90332-3).
- [40] H. Cohn and N. Elkies, *New upper bounds on sphere packings I*, Ann. of Math. **157** (2003), 689–714. [doi:10.4007/annals.2003.157.689](https://doi.org/10.4007/annals.2003.157.689).
- [41] L. Chen, J. Gu, L. Huang, W. Huang et al., *Seed-Prover: deep and broad reasoning for automated theorem proving*, preprint (2025). [arXiv:2507.23726](https://arxiv.org/abs/2507.23726).
- [42] H. Chernoff, *A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations*, Ann. Math. Statist. **23** (1952), no. 4, 493–507. [doi:10.1214/aoms/1177729330](https://doi.org/10.1214/aoms/1177729330).
- [43] V. Cohen-Addad, M. Hebdige, D. Král’, Z. Li and E. Salgado, *Steinberg’s conjecture is false*, J. Combin. Theory Ser. B **122** (2017), 452–456. [doi:10.1016/j.jctb.2016.07.006](https://doi.org/10.1016/j.jctb.2016.07.006).
- [44] P. Chojceki, *Monochromatic union-closed and lattice subfamilies of the Boolean lattice*, preprint (18 March 2026). <https://www.ulam.ai/research/erdos1183.pdf>.
- [45] S. J. Cook, T. A. Jarrell, C. A. Brittin, Y. Wang et al., *Whole-animal connectomes of both Caenorhabditis elegans sexes*, Nature **571** (2019), 63–71. [doi:10.1038/s41586-019-1352-7](https://doi.org/10.1038/s41586-019-1352-7).
- [46] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser et al., *Training verifiers to solve math word problems*, preprint (2021). [arXiv:2110.14168](https://arxiv.org/abs/2110.14168).
- [47] H. Cohn, A. Kumar, S. D. Miller, D. Radchenko and M. Viazovska, *The sphere packing problem in dimension 24*, Ann. of Math. (2) **185**

- (2017), no. 3, 1017–1033. doi:10.4007/annals.2017.185.3.8.
- [48] Clay Mathematics Institute, *The Millennium Prize Problems*, <https://www.claymath.org/millennium-problems/>, accessed 18 October 2026.
- [49] F. Charles and E. Markman, *The standard conjectures for holomorphic symplectic varieties deformation equivalent to Hilbert schemes of K3 surfaces*, *Compos. Math.* **149** (2013), 481–494. doi:10.1112/S0010437X12000607.
- [50] S. Chen, K. Marwaha, X. Lu, H. Yuen and T. Peng, *Prove2Me: an open collaborative platform for scaling math formalization*, preprint (2026). arXiv:2608.28433.
- [51] M. Creutz, *Gauge fixing, the transfer matrix, and confinement on a lattice*, *Phys. Rev. D* **15** (1977), 1128–1136. doi:10.1103/PhysRevD.15.1128.
- [52] M. Creutz, *Monte Carlo study of quantized SU(2) gauge theory*, *Phys. Rev. D* **21** (1980), 2308–2315. doi:10.1103/PhysRevD.21.2308.
- [53] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed., Wiley, Hoboken, 2006. doi:10.1002/047174882X.
- [54] Yu. Chervonyi, T. H. Trinh, M. Olšák, X. Yang et al., *Gold-medalist performance in solving olympiad geometry with AlphaGeometry2*, preprint (2025). arXiv:2502.03544.
- [55] Z. Chen, Y. Zhao and A. Cohan, *Measuring the gap between human and LLM research ideas*, preprint (2026). arXiv:2607.01233.
- [56] N. G. de Bruijn, *The mathematical language AUTOMATH, its usage, and some of its extensions*, in *Symposium on Automatic Demonstration*, *Lecture Notes in Math.* 125, Springer, Berlin, 1970, 29–61. doi:10.1007/BFb0060623.
- [57] H. Darmon, F. Diamond and R. Taylor, *Fermat’s Last Theorem*, *Current Developments in Mathematics* **1995** (1995), 1–154. doi:10.4310/CDM.1995.v1995.n1.a1.
- [58] N. Dziri, X. Lu, M. Sclar, X. L. Li, L. Jiang, B. Y. Lin et al., *Faith and fate: limits of transformers on compositionality*, *Advances in Neural Information Processing Systems* **36** (2023). arXiv:2305.18654.
- [59] R. J. Douglas and K. A. C. Martin, *Neuronal circuits of the neocortex*, *Annu. Rev. Neurosci.* **27** (2004), 419–451. doi:10.1146/annurev.neuro.27.070203.144152.
- [60] S. Dorkenwald, A. Matsliah, A. R. Sterling, P. Schlegel, S.-C. Yu, C. E. McKellar et al., *Neuronal wiring diagram of an adult brain*, *Nature* **634** (2024), 124–138. doi:10.1038/s41586-024-07558-y.
- [61] S. Dorkenwald, A. Matsliah, A. R. Sterling, P. Schlegel et al., *Neuronal wiring diagram of an adult brain*, *Nature* **634** (2024), 124–138. doi:10.1038/s41586-024-07558-y.
- [62] R. L. Dobrushin, *The description of a random field by means of conditional probabilities and conditions of its regularity*, *Theory Probab. Appl.* **13** (1968), 197–224. doi:10.1137/1113026.
- [63] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya et al., *Loihi: a neuromorphic manycore processor with on-chip learning*, *IEEE Micro* **38** (2018), no. 1, 82–99. doi:10.1109/MM.2018.112130359.
- [64] R. Durrett, *Probability: Theory and Examples*, 5th ed., Cambridge University Press, Cambridge, 2019. doi:10.1017/9781108591034.
- [65] A. Davies, P. Veličković, L. Buesing, S. Blackwell, D. Zheng, N. Tomašev et al., *Advancing mathematics by guiding human intuition with AI*, *Nature* **600** (2021), no. 7887, 70–74. doi:10.1038/s41586-021-04086-x.
- [66] M. Dwass, *The total progeny in a branching process and a related random walk*, *J. Appl. Probab.* **6** (1969), no. 3, 682–686. doi:10.2307/3212112.
- [67] N. Eckstein, A. S. Bates, A. Champion, M. Du et al., *Neurotransmitter classification from electron microscopy images at synaptic sites in Drosophila melanogaster*, *Cell* **187** (2024), 2574–2594. doi:10.1016/j.cell.2024.03.016.
- [68] P. Elias, *Universal codeword sets and representations of the integers*, *IEEE Trans. Inform. Theory* **21** (1975), no. 2, 194–203. doi:10.1109/TIT.1975.1055349.
- [69] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatin et al., *Discovering faster matrix multiplication algorithms with reinforcement learning*, *Nature* **610** (2022), no. 7930, 47–53. doi:10.1038/s41586-022-05172-4.
- [70] The Formal Conjectures Authors, *Formal Conjectures*, Lean 4 statements of open and solved conjectures, Google DeepMind (2025). github.com/google-deepmind/formal-conjectures.
- [71] N. Frémaux and W. Gerstner, *Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules*, *Front. Neural Circuits* **9** (2016), 85. doi:10.3389/fncir.2015.00085.
- [72] S. Floccari, *On the motive of O’Grady’s six dimensional hyper-Kähler varieties*, *Épjournal Géom. Algébrique* **7** (2023). doi:10.46298/epiga.2022.9758.
- [73] J. Foster, *The Lefschetz standard conjectures for IHSMs of generalized Kummer deformation type in certain degrees*, preprint (2023). arXiv:2303.14327.
- [74] J. Foster, *The Lefschetz standard conjectures for 4d-dimensional Kummer-type hyper-Kähler varieties*, preprint (2025). arXiv:2512.04114.
- [75] R. G. Gallager, *Variations on a theme by Huffman*, *IEEE Trans. Inform. Theory* **24** (1978), no. 6, 668–674. doi:10.1109/TIT.1978.1055959.
- [76] B. Georgiev, J. Gómez-Serrano, T. Tao and A. Z. Wagner, *Mathematical exploration and discovery at scale*, preprint (2025). arXiv:2511.02864.
- [77] W. Gerstner, W. M. Kistler, R. Naud and L. Paninski, *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*, Cambridge University Press, Cambridge, 2014. doi:10.1017/CB09781107447615.
- [78] S. Goldwasser, S. Micali and C. Rackoff, *The knowledge complexity of interactive proof systems*, *SIAM J. Comput.* **18** (1989), no. 1, 186–208. doi:10.1137/0218012.
- [79] M. J. Gordon, A. J. Milner and C. P. Wadsworth, *Edinburgh LCF: A Mechanised Logic of Computation*, *Lecture Notes in Comput. Sci.* 78, Springer, Berlin, 1979. doi:10.1007/3-540-09724-4.
- [80] K. Gödel, *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*, *Monatsh. Math. Phys.* **38** (1931), 173–198. doi:10.1007/BF01700692.
- [81] I. J. Good, *Speculations concerning the first ultraintelligent machine*, *Advances in Computers* **6** (1966), 31–88. doi:10.1016/S0065-2458(08)60418-0.
- [82] D. Guo, *The deterministic horizon: impossibility results as design specifications for trustworthy AI systems*, preprint (2026). arXiv:2605.23024.
- [83] B. H. Gross and D. B. Zagier, *Heegner points and derivatives of L-series*, *Invent. Math.* **84** (1986), 225–320. doi:10.1007/BF01388809.
- [84] A. Hatcher, *Algebraic Topology*, Cambridge University Press, Cambridge, 2002.
- [85] S. Hariharan, C. Birkbeck, S. Lee, H. K. G. Ma, B. Mehta, A. Poiroux and M. Viazovska, *Progress in formalizing sphere packing in dimension 8*, preprint (2026). arXiv:2604.23468.
- [86] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford et al., *Training compute-optimal large language models*, preprint (2022). arXiv:2203.15556.
- [87] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, Wiley, New York, 1949.

- [88] J. Huang, S. S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu and J. Han, *Large language models can self-improve*, preprint (2022). [arXiv:2210.11610](#).
- [89] X.-D. Hou, K. H. Leung and Q. Xiang, *A generalization of an addition theorem of Kneser*, J. Number Theory **97** (2002), no. 1, 1–9. [doi:10.1006/jnth.2002.2793](#).
- [90] T. Hubert, R. Mehta, L. Sartran, M. Z. Horváth, G. Žužić, E. Wieser et al., *Olympiad-level formal mathematical reasoning with reinforcement learning*, Nature **651** (2026), no. 8106, 607–613; published online 2025. [doi:10.1038/s41586-025-09833-y](#).
- [91] W. Hoeffding, *Probability inequalities for sums of bounded random variables*, J. Amer. Statist. Assoc. **58** (1963), no. 301, 13–30. [doi:10.1080/01621459.1963.10500830](#).
- [92] H. A. Helfgott and M. Radziwiłł, *Expansion, divisibility and parity*, preprint (2021). [arXiv:2103.06853](#).
- [93] D. A. Huffman, *A method for the construction of minimum-redundancy codes*, Proc. IRE **40** (1952), no. 9, 1098–1101. [doi:10.1109/JRPROC.1952.273898](#).
- [94] M. Hutter, *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*, Springer, Berlin, 2005. [doi:10.1007/b138233](#).
- [95] J. L. W. V. Jensen, *Sur les fonctions convexes et les inégalités entre les valeurs moyennes*, Acta Math. **30** (1906), 175–193. [doi:10.1007/BF02418571](#).
- [96] H. Jaeger and H. Haas, *Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication*, Science **304** (2004), 78–80. [doi:10.1126/science.1091277](#).
- [97] Z. Jia, A. Rakhlin and T. Xie, *Do we need to verify step by step? Rethinking process supervision from a theoretical perspective*, preprint (2025). [arXiv:2502.10581](#).
- [98] K. Jamieson and A. Talwalkar, *Non-stochastic best arm identification and hyperparameter optimization*, in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (2016). [arXiv:1502.07943](#).
- [99] S. L. Kleiman, *The standard conjectures*, in Motives (Seattle, 1991), Proc. Sympos. Pure Math. 55, Part 1, Amer. Math. Soc., Providence, 1994, 3–20. [doi:10.1090/pspum/055.1/1265519](#).
- [100] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu and D. Amodei, *Scaling laws for neural language models*, preprint (2020). [arXiv:2001.08361](#).
- [101] V. A. Kolyvagin, *Finiteness of $E(\mathbf{Q})$ and $Sha(E, \mathbf{Q})$ for a subclass of Weil curves*, Math. USSR-Izv. **32** (1989), 523–541. [doi:10.1070/IM1989v032n03ABEH000779](#).
- [102] L. G. Kraft, *A device for quantizing, grouping, and coding amplitude-modulated pulses*, M.S. thesis, Massachusetts Institute of Technology, 1949. [hdl:1721.1/12390](#).
- [103] H. Kesten and B. P. Stigum, *A limit theorem for multidimensional Galton–Watson processes*, Ann. Math. Statist. **37** (1966), no. 5, 1211–1223. [doi:10.1214/aoms/1177699266](#).
- [104] C. Liu, Y. Chen, T. Liu, J. Cheng, B. Han and K. Zhang, *On the thinking-language modeling gap in large language models*, preprint (2025). [arXiv:2505.12896](#).
- [105] T. Leelawat and L. D. Griffin, *Watson & Holmes: a naturalistic benchmark for comparing human and LLM reasoning*, preprint (2026). [arXiv:2602.19914](#).
- [106] D. I. Lieberman, *Numerical and homological equivalence of algebraic cycles on Hodge manifolds*, Amer. J. Math. **90** (1968), 366–374. [doi:10.2307/2373533](#).
- [107] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever and K. Cobbe, *Let’s verify step by step*, preprint (2023). [arXiv:2305.20050](#).
- [108] G. Lample, M.-A. Lachaux, T. Lavril, X. Martinet et al., *HyperTree proof search for neural theorem proving*, Advances in Neural Information Processing Systems **35** (2022). [arXiv:2205.11491](#).
- [109] M. H. Löb, *Solution of a problem of Leon Henkin*, J. Symbolic Logic **20** (1955), no. 2, 115–118. [doi:10.2307/2266895](#).
- [110] Y. Lin, S. Tang, B. Lyu, J. Wu et al., *Goedel-Prover: a frontier model for open-source automated theorem proving*, preprint (2025). [arXiv:2502.07640](#).
- [111] J. K. Lappalainen, F. D. Tschopp, S. Prakhya, M. McGill et al., *Connectome-constrained networks predict neural activity across the fly visual system*, Nature **634** (2024), 1132–1140. [doi:10.1038/s41586-024-07939-3](#).
- [112] C.-P. Lu, *Verification and self-improvement in agentic AI: foundations and limits*, preprint (2026). [arXiv:2610.10611](#).
- [113] M. Lüscher, *Properties and uses of the Wilson flow in lattice QCD*, J. High Energy Phys. **2010** (2010), no. 8, 071. [doi:10.1007/JHEP08\(2010\)071](#).
- [114] W. Maass, *Networks of spiking neurons: the third generation of neural network models*, Neural Networks **10** (1997), 1659–1671. [doi:10.1016/S0893-6080\(97\)00011-7](#).
- [115] E. Markman, *Cycles on abelian 2n-folds of Weil type from secant sheaves on abelian n-folds*, preprint (2025), [arXiv:2502.03415](#).
- [116] B. McMillan, *Two inequalities implied by unique decipherability*, IRE Trans. Inform. Theory **2** (1956), no. 4, 115–116. [doi:10.1109/TIT.1956.1056818](#).
- [117] C. Mead, *Neuromorphic electronic systems*, Proc. IEEE **78** (1990), 1629–1636. [doi:10.1109/5.58356](#).
- [118] D. J. Mankowitz, A. Michi, A. Zhernov, M. Gelmi, M. Selvi, C. Paduraru et al., *Faster sorting algorithms discovered using deep reinforcement learning*, Nature **618** (2023), no. 7964, 257–263. [doi:10.1038/s41586-023-06004-9](#).
- [119] K. Matomäki, M. Radziwiłł, T. Tao, J. Teräväinen and T. Ziegler, *Higher uniformity of bounded multiplicative functions in short intervals on average*, Ann. of Math. (2) **197** (2023), 739–857. [doi:10.4007/annals.2023.197.2.3](#).
- [120] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe et al., *Self-refine: iterative refinement with self-feedback*, Advances in Neural Information Processing Systems **36** (2023). [arXiv:2303.17651](#).
- [121] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in Automated Deduction – CADE 28, Lecture Notes in Comput. Sci. 12699, Springer, Cham, 2021, 625–635. [doi:10.1007/978-3-030-79876-5_37](#).
- [122] B. J. J. Moonen and Yu. G. Zarhin, *Weil classes on abelian varieties*, J. Reine Angew. Math. **496** (1998), 83–92. [doi:10.1515/crll.1998.034](#).
- [123] A. Novikov, N. Vü, M. Eisenberger, E. Dupont et al., *AlphaEvolve: a coding agent for scientific and algorithmic discovery*, preprint (2025). [arXiv:2506.13131](#).
- [124] OpenAI, *Finite time blowup for Navier–Stokes*, preprint (2026), with the Lean 4 formalization [openai/NavierStokesAndEuler.cdn.openai.com/pdf/32d9f210-8b73-45e0-91bc-82a30aef8a9a/navier-stokes.pdf](#).
- [125] OpenAI, *Mathematical manuscripts and proof artifacts*, repository [openai/math](#) (2026), 722 manuscripts in 372 families; mirror [github.com/creelie/math01-openAI](#).
- [126] OpenAI, *Ordinary two-point correlations of multiplicative functions*, preprint (24 September 2026), with a Lean 4 formalization, in [openai/math](#); mirror [github.com/creelie/math01-openAI](#).
- [127] The Omega Institute, *trureturning: a scientific method for AI to discover truth and find its next question*, repository (2026), with 782 problem dossiers and their Lean 4 resolutions; [github.com/the-omega-institute/trureturning](#); copy [github.com/creelie/trureturning-OpenRefute](#).
- [128] T. Ord, *The dynamics of intelligence explosions*, preprint (2026). [arXiv:2608.14426](#).

- [129] The Omega Institute, *trureturing*, Lean 4 theorem `odd_lower_bound` in `D5/S3/Combinatorics/ErdosUlam/SublatticeChainBound.lean`, repository `the-omega-institute/trureturing` (2026), commit `fa35d5b` github.com/creelie/trureturing-OpenRefute.
- [130] K. Osterwalder and E. Seiler, *Gauge field theories on a lattice*, Ann. Physics **110** (1978), 440–471. doi:10.1016/0003-4916(78)90039-8.
- [131] W. F. Osgood, *Beweis der Existenz einer Lösung der Differentialgleichung $dy/dx = f(x, y)$ ohne Hinzunahme der Cauchy–Lipschitz’schen Bedingung*, Monatsh. Math. Phys. **9** (1898), 331–345. doi:10.1007/BF01707876.
- [132] R. Otter, *The multiplicative process*, Ann. Math. Statist. **20** (1949), no. 2, 206–224. doi:10.1214/aoms/1177730031.
- [133] G. Perelman, *The entropy formula for the Ricci flow and its geometric applications*, preprint (2002). arXiv:math/0211159.
- [134] G. Perelman, *Ricci flow with surgery on three-manifolds*, preprint (2003). arXiv:math/0303109.
- [135] G. Perelman, *Finite extinction time for the solutions to the Ricci flow on certain three-manifolds*, preprint (2003). arXiv:math/0307245.
- [136] C. Pilatte, *Improved bounds for the two-point logarithmic Chowla conjecture*, preprint (2023). arXiv:2310.19357.
- [137] C. Pilatte, *Improved bounds for the Fourier uniformity conjecture*, preprint (2026). arXiv:2604.26564.
- [138] W. H. Press, *Strong profiling is not mathematically optimal for discovering rare malefactors*, Proc. Natl. Acad. Sci. USA **106** (2009), 1716–1719. doi:10.1073/pnas.0813202106.
- [139] K. Pratt, N. Robles, A. Zaharescu and D. Zeindler, *More than five-twelfths of the zeros of ζ are on the critical line*, Res. Math. Sci. **7** (2020), 2. doi:10.1007/s40687-019-0199-8.
- [140] S. Polu and I. Sutskever, *Generative language modeling for automated theorem proving*, preprint (2020). arXiv:2009.03393.
- [141] D. Platt and T. Trudgian, *The Riemann hypothesis is true up to $3 \cdot 10^{12}$* , Bull. Lond. Math. Soc. **53** (2021), 792–797. doi:10.1112/blms.12460.
- [142] Qwen Team (A. Yang et al.), *Qwen2.5 technical report*, preprint (2024). arXiv:2412.15115.
- [143] B. Romera-Paredes, M. Barekatin, A. Novikov, M. Balog, M. P. Kumar, E. Dupont et al., *Mathematical discoveries from program search with large language models*, Nature **625** (2024), no. 7995, 468–475. doi:10.1038/s41586-023-06924-6.
- [144] A. Rényi, *On measures of entropy and information*, in Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, University of California Press, Berkeley, 1961, 547–561.
- [145] J. Rissanen, *A universal prior for integers and estimation by minimum description length*, Ann. Statist. **11** (1983), no. 2, 416–431. doi:10.1214/aos/1176346150.
- [146] P. Redgrave, T. J. Prescott and K. Gurney, *The basal ganglia: a vertebrate solution to the selection problem?*, Neuroscience **89** (1999), 1009–1023. doi:10.1016/S0306-4522(98)00319-4.
- [147] A. A. Razborov and S. Rudich, *Natural proofs*, J. Comput. System Sci. **55** (1997), 24–35. doi:10.1006/jcss.1997.1494.
- [148] D. Rohatgi, A. Shetty, D. Salese, Y. Li, A. Moitra, A. Risteski and D. J. Foster, *Taming imperfect process verifiers: a sampling perspective on backtracking*, preprint (2025). arXiv:2510.03149.
- [149] Z. Z. Ren, Z. Shao, J. Song, H. Xin et al., *DeepSeek-Prover-V2: advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition*, preprint (2025). arXiv:2504.21801.
- [150] J. Schmidhuber, *Gödel machines: self-referential universal problem solvers making provably optimal self-improvements*, preprint (2003) arXiv:cs/0309048.
- [151] J. T. Schwartz, *Fast probabilistic algorithms for verification of polynomial identities*, J. ACM **27** (1980), no. 4, 701–717. doi:10.1145/322217.322225.
- [152] C. Schoen, *Addendum to: Hodge classes on self-products of a variety with an automorphism*, Compos. Math. **114** (1998), 321–328. doi:10.1023/A:1000566205021.
- [153] A. Scrivens, *Empirical validation of the classification-verification dichotomy for AI safety gates*, preprint (2026). arXiv:2604.00072.
- [154] W. Schultz, P. Dayan and P. R. Montague, *A neural substrate of prediction and reward*, Science **275** (1997), 1593–1599. doi:10.1126/science.275.5306.1593.
- [155] D. Saxton, E. Grefenstette, F. Hill and P. Kohli, *Analysing mathematical reasoning abilities of neural models*, International Conference on Learning Representations (2019). arXiv:1904.01557.
- [156] C. E. Shannon, *A mathematical theory of communication*, Bell System Tech. J. **27** (1948), no. 3, 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x.
- [157] T. Shioda, *The Hodge conjecture for Fermat varieties*, Math. Ann. **245** (1979), 175–184. doi:10.1007/BF01428804.
- [158] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche et al., *Mastering the game of Go with deep neural networks and tree search*, Nature **529** (2016), no. 7587, 484–489. doi:10.1038/nature16961.
- [159] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez et al., *A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play*, Science **362** (2018), no. 6419, 1140–1144. doi:10.1126/science.aar6404.
- [160] A. Shapson-Coe, M. Januszewski, D. R. Berger, A. Pope et al., *A petavoxel fragment of human cerebral cortex reconstructed at nanoscale resolution*, Science **384** (2024), eadk4858. doi:10.1126/science.adk4858.
- [161] C. Snell, J. Lee, K. Xu and A. Kumar, *Scaling LLM test-time compute optimally can be more effective than scaling model parameters*, preprint (2024). arXiv:2408.03314.
- [162] R. J. Solomonoff, *A formal theory of inductive inference. Part I*, Information and Control **7** (1964), no. 1, 1–22. doi:10.1016/S0019-9958(64)90223-2.
- [163] R. Steinberg, *The state of the three color problem*, in Quo Vadis, Graph Theory?, Ann. Discrete Math. **55** (1993), 211–248. doi:10.1016/S0167-5060(08)70391-1.
- [164] N. Seiberg and E. Witten, *Electric-magnetic duality, monopole condensation, and confinement in $N = 2$ supersymmetric Yang–Mills theory*, Nuclear Phys. B **426** (1994), 19–52. doi:10.1016/0550-3213(94)90124-4.
- [165] H. Shen, R. Zhu and X. Zhu, *A stochastic analysis approach to lattice Yang–Mills at strong coupling*, Comm. Math. Phys. **400** (2023), 805–851. doi:10.1007/s00220-022-04609-1.
- [166] S. G. Tankeev, *On the standard conjecture of Lefschetz type for complex projective threefolds*, Izv. Math. **74** (2010), 167–187. doi:10.1070/IM2010v074n01ABEH002484.
- [167] S. G. Tankeev, *On the standard conjecture of Lefschetz type for complex projective threefolds. II*, Izv. Math. **75** (2011), 1047–1062. doi:10.1070/IM2011v075n05ABEH002563.
- [168] T. Tao, *The logarithmically averaged Chowla and Elliott conjectures for two-point correlations*, Forum Math. Pi **4** (2016), e8. doi:10.1017/fmp.2016.6.
- [169] G. G. Turrigiano, K. R. Leslie, N. S. Desai, L. C. Rutherford and S. B. Nelson, *Activity-dependent scaling of quantal amplitude in neocortical neurons*, Nature **391** (1998), 892–896. doi:10.1038/36103.
- [170] T. Tao and J. Teräväinen, *The structure of logarithmically averaged correlations of multiplicative functions, with applications to the Chowla and Elliott conjectures*, Duke Math. J. **168** (2019),

- 1977–2027. [doi:10.1215/00127094-2019-0002](https://doi.org/10.1215/00127094-2019-0002).
- [171] T. Tao and J. Teräväinen, *Odd order cases of the logarithmically averaged Chowla conjecture*, J. Théor. Nombres Bordeaux **30** (2018), 997–1015. [doi:10.5802/jtnb.1062](https://doi.org/10.5802/jtnb.1062).
- [172] A. M. Turing, *Computing machinery and intelligence*, Mind **59** (1950), no. 236, 433–460. [doi:10.1093/mind/LIX.236.433](https://doi.org/10.1093/mind/LIX.236.433).
- [173] R. Taylor and A. Wiles, *Ring-theoretic properties of certain Hecke algebras*, Ann. of Math. (2) **141** (1995), no. 3, 553–572. [doi:10.2307/2118560](https://doi.org/10.2307/2118560).
- [174] T. H. Trinh, Y. Wu, Q. V. Le, H. He and T. Luong, *Solving olympiad geometry without human demonstrations*, Nature **625** (2024), no. 7995, 476–482. [doi:10.1038/s41586-023-06747-5](https://doi.org/10.1038/s41586-023-06747-5).
- [175] J. Uesato, N. Kushman, R. Kumar, F. Song, N. Siegel, L. Wang, A. Creswell, G. Irving and I. Higgins, *Solving math word problems with process- and outcome-based feedback*, preprint (2022). [arXiv:2211.14275](https://arxiv.org/abs/2211.14275).
- [176] M. Varesco, *Hodge similarities, algebraic classes, and Kuga–Satake varieties*, Math. Z. **305** (2023), 69. [doi:10.1007/s00209-023-03390-8](https://doi.org/10.1007/s00209-023-03390-8).
- [177] L. R. Varshney, B. L. Chen, E. Paniagua, D. H. Hall and D. B. Chklovskii, *Structural properties of the Caenorhabditis elegans neuronal network*, PLoS Comput. Biol. **7** (2011), e1001066. [doi:10.1371/journal.pcbi.1001066](https://doi.org/10.1371/journal.pcbi.1001066).
- [178] M. Viazovska, *The sphere packing problem in dimension 8*, Ann. of Math. (2) **185** (2017), no. 3, 991–1015. [doi:10.4007/annals.2017.185.3.7](https://doi.org/10.4007/annals.2017.185.3.7).
- [179] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, *Attention is all you need*, Advances in Neural Information Processing Systems **30** (2017). [arXiv:1706.03762](https://arxiv.org/abs/1706.03762).
- [180] A. Wald, *Sequential tests of statistical hypotheses*, Ann. Math. Statist. **16** (1945), no. 2, 117–186. [doi:10.1214/aoms/1177731118](https://doi.org/10.1214/aoms/1177731118).
- [181] S. Weir, *obsidian-brain*, server for semantic search and graph analysis of an Obsidian vault, npm package, version 1.7.24 (2026). sweir1.github.io/obsidian-brain.
- [182] A. Weil, *Number Theory: An Approach through History from Hammurapi to Legendre*, Birkhäuser, Boston, 1984. [doi:10.1007/978-0-8176-4571-7](https://doi.org/10.1007/978-0-8176-4571-7).
- [183] H. W. Watson and F. Galton, *On the probability of the extinction of families*, J. Anthropol. Inst. Great Britain and Ireland **4** (1875), 138–144. [doi:10.2307/2841222](https://doi.org/10.2307/2841222).
- [184] A. Wiles, *Modular elliptic curves and Fermat’s Last Theorem*, Ann. of Math. (2) **141** (1995), no. 3, 443–551. [doi:10.2307/2118559](https://doi.org/10.2307/2118559).
- [185] E. Witten, *Quantum field theory and the Jones polynomial*, Comm. Math. Phys. **121** (1989), 351–399. [doi:10.1007/BF01217730](https://doi.org/10.1007/BF01217730).
- [186] H. Wang, M. Unsal, X. Lin, M. Baksys et al., *Kimina-Prover Preview: towards large formal reasoning models with reinforcement learning*, preprint (2025). [arXiv:2504.11354](https://arxiv.org/abs/2504.11354).
- [187] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le and D. Zhou, *Chain-of-thought prompting elicits reasoning in large language models*, Advances in Neural Information Processing Systems **35** (2022). [arXiv:2201.11903](https://arxiv.org/abs/2201.11903).
- [188] K. Yang, A. M. Swope, A. Gu, R. Chalamala et al., *LeanDojo: theorem proving with retrieval-augmented language models*, Advances in Neural Information Processing Systems **36** (2023), Datasets and Benchmarks Track. [arXiv:2306.15626](https://arxiv.org/abs/2306.15626).
- [189] A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li et al., *Qwen2.5-Math technical report: toward mathematical expert model via self-improvement*, preprint (2024). [arXiv:2409.12122](https://arxiv.org/abs/2409.12122).
- [190] Z. Zhang, A. Delaval, L. Chen, J. Chen, J. Xu, Y. Liao, J. Jiang, C. Liu and B. Dong, *An AI-assisted formalization of the Poincaré conjecture*, preprint (2026). [arXiv:2610.08329](https://arxiv.org/abs/2610.08329).
- [191] R. Zippel, *Probabilistic algorithms for sparse polynomials*, in Symbolic and Algebraic Computation (EUROSAM ’79), Lecture Notes in Comput. Sci. 72, Springer, Berlin, 1979, 216–226. [doi:10.1007/3-540-09519-5_73](https://doi.org/10.1007/3-540-09519-5_73).
- [192] E. Zelikman, Y. Wu, J. Mu and N. D. Goodman, *STaR: bootstrapping reasoning with reasoning*, Advances in Neural Information Processing Systems **35** (2022). [arXiv:2203.14465](https://arxiv.org/abs/2203.14465).