hexagon:2611.00149v1

WHAT THE ETH REFUTATIONS DO NOT GIVE: BARRIERS FOR THE COUNTING ENGINES, WITH MACHINE-CHECKED LEMMAS

VAIBHAV TULSYAN

Abstract. In October 2026 OpenAI and Anthropic each claimed a deterministic subexponential algorithm for counting the solutions of a k-CNF; both counting engines sum products of local tables. For SAT itself, Majdoddin bounds the rate of Anthropic's grid recursions with product terms, and Dias and Lima bound OpenAI's clause-product scheme. However, Majdoddin notes that no barrier is known for decompositions whose terms are not products, and neither paper treats the counting problems beyond SAT. We introduce the *engine barrier problem*: bound from below the shared bits U of every network of a stated class that computes a given count, and the rate of every recursion of a stated class. Our approach puts a rank on every cut of a local network, and on one contraction inside both flattenings of a term. Machine-checked in Lean are: correlated (non-product) decompositions of two rectangles, of the 3×2 grid and of the full 3×3 grid level have flattening-rank sums at least N^2, N^3 and N^8, while N^9 is open; the row cut of the permanent has rank $\binom{n}{k}$, which on paper gives $U \geq n \log_2(n/s)$ for row-local networks; algebras in which the permanent is a functional of a product have dimension at least $\binom{n}{n/2}$; pieces have saving $\rho \geq 1/2$ (rank form; Anthropic's observation); and exact colouring filters for Hamiltonicity need $q \geq n - 4$ colours. On paper we prove $U \geq (1 - o(1))n \log_2 n$ for vertex-local networks for the Tutte polynomial off the integer hyperbolas, and $\rho \geq 3/8$ for correlated decompositions of full grid levels. We cite 45 Lean theorems, with standard axioms only. Every barrier is about a named model: in Anthropic's general model no unconditional bound on U exists without separating P from $\mathrm{P}^{\#\mathrm{P}}$, and an unconditional cap on oblivious algebraic algorithms would separate VP from VNP.

1. Introduction

On 26 October 2026 OpenAI [47] and Anthropic [21] (below [O] and [A]) each posted a claimed refutation of the Exponential Time Hypothesis: a deterministic algorithm that counts the satisfying assignments of a k-CNF on n variables in time $2^{O_k(n/\sqrt{\log n})}$ up to lower-order factors, with a Lean 4 development. Both claims are under review. Both engines are counting algorithms for sums of products of local tables, and they reach every problem whose witnesses have $O(n)$ bits and are checked locally. Our earlier work used them, with machine-checked proofs, for exact counting of Hamiltonian cycles and TSP tours on sparse graphs [59], Eulerian circuits [58], amplitudes of sparse quantum circuits [61], factoring [62], class numbers [57] and the parity of $\pi(x)$ [63].

For #SAT itself, two papers that appeared while this one was in preparation give negative accounts. Majdoddin [43] defines a rewrite-and-merge class that contains [A]'s recursion and proves that every grid algorithm in it has exponent at least $1/\log_2 K$ per edge bit; Dias and Lima [29] prove that the clause-product form of [O]'s interpolation algorithm costs $2^{\Omega(\sqrt{\mathrm{tw}})}$ at treewidth tw, in a cost model that charges what the preprint pays. The labs' own remarks are shorter. [A] writes that its theorem "does not refute the Orthogonal Vectors hypothesis" and that the result "is purely asymptotic" (Section 2). [O] notes that a faster algorithm for a target problem needs "a reduction in the other direction, from that problem to fixed-width SAT, that controls the number of variables", and that "runtime bounds of the form $2^{o(n)}$ do not transfer under general polynomial-time reductions". However, Majdoddin's class requires every term of a recursion to be a product of one piece per configuration; he calls non-product groupings of terms the "open door" of his class and writes that "for the full model of Signed Network Sums no barrier is known" [43, §4.8]. Neither paper treats a counting problem other than (Max-)SAT. We are unaware of a barrier for non-product decompositions, or

Date: November 2026.

2020 *Mathematics Subject Classification.* 68Q17, 68Q25, 68W30, 15A69, 05C31.

Key words and phrases. Exponential Time Hypothesis, counting algorithms, tensor networks, cut rank, permanent, Tutte polynomial, lower bounds for restricted models, Lean 4, formal verification.

of an account of which counting problems beyond SAT the engines cannot be expected to reach, by which mechanism, and what would have to change. This matters because the engines are new, and the obvious targets after sparse #HC (the permanent, dense #HC and TSP, the chromatic number, the Tutte polynomial, Subset Sum, exact $\pi(x)$) are exactly the ones they seem to miss.

1.1. **The problem and the running example.** We consider the *engine barrier problem*. For a counting problem P and a stated class of networks, bound from below the number U of shared bits of every network in the class whose value is P; and for a stated class of recursions, bound from below the rate e_K at which the recursion evaluates the dense (K, λ)-clique. The two numbers combine: the engine's own guarantee on a network is at least $2^{e_K U}$ (Theorem 2.4), and $e_K = \Theta(1/\sqrt{\log K})$ in [A]. So a dense problem is out of reach of the engine's guarantee when U is of order $n \log n$ and e_K is not $o(1/\log K)$. For grid recursions with product terms, $e_K = \Omega(1/\log K)$ is a theorem of Majdoddin [43, Theorem 4.13]; the question we add is what terms that are not products can do.

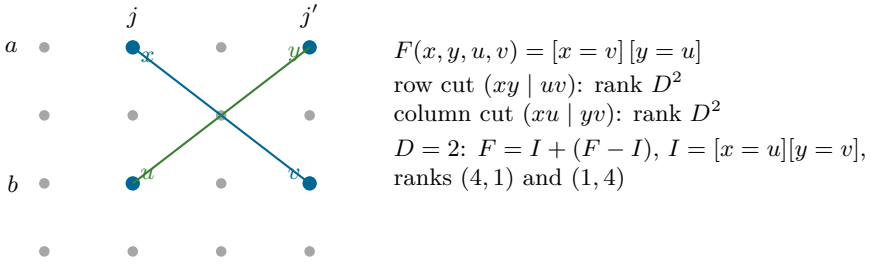


FIGURE 1. The running example. One level of [A]'s recursion evaluates a clique on a $g \times g$ grid of nodes; each pair of rows $a < b$ and columns $j < j'$ carries a *rectangle*, two crossing equalities of λ bits each. Its tensor F has both flattenings of full rank D^2, $D = 2^\lambda$. At $D = 2$ the split $F = I + (F - I)$ gives two terms, each cheap in one grouping: this is the profile of saving $\rho = 1/2$.

Figure 1 is our running example. One level of [A]'s recursion views a clique on g^2 nodes as a $g \times g$ grid. Each pair of rows and pair of columns spans a rectangle with corner legs $x, y, u, v \in [D]$ and tensor $F = [x = v][y = u]$. Evaluated row by row, a term costs the rank of its row flattening $(xy \mid uv)$; evaluated column by column, the rank of its column flattening $(xu \mid yv)$. For F both are permutation matrices of rank D^2. The recursion therefore splits F into *pieces*, terms that are cheap in one of the two groupings, and the saving ρ per level decides its rate. At $D = 2$ the split $F = I + (F - I)$ has rank profiles $(4, 1)$ and $(1, 4)$: each piece pays 2 bits in one grouping and none in the other, half of the 4 bits of F. We come back to this example in Section 2 (what a shared bit buys) and in Section 4: on two rectangles and on the 3×2 grid no decomposition, correlated or not, does better asymptotically, and on the 3×3 level the question is open.

1.2. **The naive route and the principle.** At first one might try to prove that the dense permanent needs $2^{\Omega(n)}$ steps in the engine, for example by bounding U from below for every network whose value is $\mathrm{per}(A)$. However, in [A]'s Definition 2 the node tables are arbitrary polynomial-time functions of the whole input; Ryser's formula is then a one-node network with $U = 0$, and a network with $U = 0$ and one setting exists if and only if $\mathrm{per} \in \mathrm{FP}$ (Theorem 3.13). Instead, we rely on a different principle: *every known network is local, and locality lets a cut carry a rank.* If the nodes on one side of a cut read one part of the input and the nodes on the other side read the rest, then the value, as a matrix indexed by the two parts, has rank at most 2^w, where w is the number of shared bits across the cut. The same rank argument bounds the cost of a term of a recursion through its two flattenings, as in Fig. 1.

In Anthropic's general network model a node table may be any polynomial-time function of the whole input, so no unconditional lower bound on shared bits is possible without separating P from $\mathrm{P}^{\#\mathrm{P}}$ (Theorem 3.13); every theorem of this paper is about a named class of local networks or recursions.

1.3. **What we do.** In this paper we define the models (row-local and vertex-local networks, pieces, grouping recursions, correlated decompositions, [O]'s M_0-template, tree verifiers), prove rank lower bounds in each, and assemble them into a map of nine problems. The key observation is that the two numbers U and e_K

are each controlled by one rank: U by the rank of the value across a cut of the input, and e_K by the rank of a contraction that is a submatrix of both flattenings of a term at once. Many ingredients are classical: Nisan's rank characterisation of noncommutative branching programs [46], Lindström's factorisation of meet matrices [41], and rank subadditivity. The product-term caps of Sections 4.1 and 4.3 are parallel to, and weaker than, Majdoddin's [43]. What is new is the correlated decompositions of Section 4.4, the walls for the problems beyond SAT (Sections 3, 6 and 7), and a Lean development that checks every rank statement we cite.

1.4. **Contributions.** Our contributions are:

- We introduce the engine barrier problem and its two numbers, and define the network and recursion models in which we bound them (Section 2).
- We prove that dense counting costs $\Omega(n \log n)$ shared bits in the local networks we can analyse: $U \geq n \log_2(n/s)$ for row-local networks for the permanent (rank and counting steps in Lean), $\binom{n}{n/2}$ for algebras (Lean), and, on paper, $(1 - o(1))\, n \log_2 n$ for vertex-local networks for the Tutte polynomial off the integer hyperbolas (Section 3).
- We prove, to our knowledge, the first barriers for correlated (non-product) decompositions, the region that Majdoddin's class leaves open: rank sums at least N^2, N^3 and N^8 on two rectangles, the 3×2 grid and the 3×3 level (Lean, rank form), $\rho \geq 3/8$ on full levels (on paper), and a proof that flattening certificates stop at 8 on the 3×3 level (Section 4.4).
- We machine-check the rank form of $\rho \geq 1/2$ for pieces ([A]'s Appendix H observation, generalised in [43]), and prove on paper $\rho \geq 1/\sqrt{2} - o(1)$ for graded two-order pieces and $\alpha' \leq 1$ for every family of groupings of the grid rectangles (Sections 4.1 to 4.3); and we prove that [O]'s M_0-template is tight at $R \approx \sqrt{Q}$ (Section 5).
- We prove floors for three of our own linear encodings of Hamiltonicity: $q \geq n - 4$ colours, $k^*(H_t) \geq (\log_2 t - 10)/20$ turn bits, and a support threshold $t_p \leq \log_2 \#S$ below which every residue query vanishes (Lean, Section 6).
- We map nine problems out of reach (Table 2, Section 7), state what would have to change and why proving that it will not is itself out of reach (Section 8), and describe the Lean development: 45 cited theorems in 16 modules (Section 9).

1.5. **Status of each claim.** We separate four kinds of claim, and every theorem environment carries its label. *Machine-checked in Lean* (Theorem, "machine-checked"): the 45 statements named in Section 9, with the axioms `propext`, `Classical.choice` and `Quot.sound` only. Two of them have hypotheses that stand for steps proved on paper (**??** 3.3**??** 3.4), and the bracket says so; the hypothesis of Theorem 6.2 is discharged in Lean by Theorem 6.1. The Lean statements about correlated decompositions bound ranks of flattenings; the step "a network of cost c has flattening rank at most 2^c" is on paper. *Proved on paper only* (Proposition, "on paper", with the research note and its own number): the engine exponent (Theorem 2.4), the general-model statement (Theorem 3.13), Model V and the Tutte barrier (**??** 3.5**??** 3.10), the entropy bounds for #SAT encodings (Theorem 3.9), the graded two-order bound and the grouping caps (**??** 4.4**??** 4.6–4.8), the full-level bound $\rho \geq 3/8$ and its limit (**??** 4.12**??** 4.13), the template bounds (Theorem 5.2), the black-box and adversary statements, the accounting for Subset Sum and products (**??** 7.3**??** 7.5), and the meta-barrier and conditional cap (**??** 8.1**??** 8.2). *Assumed*: the statements of [A] and [O] that we quote as Imported Theorems, pinned to commits and sections. They enter only as the object our barriers are about; no barrier of this paper depends on their correctness. *Checked numerically* (Remark, "checked numerically"): exact rank tables, the $\lambda = 1$ searches and the integer programs of Section 4.4. They never support an asymptotic claim. No theorem in this paper is about all algorithms.

Outline. Section 2 recalls the engines and defines the models. Section 3 proves the $n \log n$ wall for shared bits, Section 4 the caps on the rate and Section 5 the tightness of [O]'s template. Section 6 treats our own encodings, Section 7 the nine problems and Section 8 what would have to change. Section 9 describes the Lean development, and Sections 10 to 12 cover related work, limitations and conclusions.

2. The engines and their two numbers

In this section we recall the statements of [A] and [O] that our barriers are about, define the network models, and prove on paper that the engine's guarantee on a network is at least $2^{e_K U}$. We use [A]'s development at commit `16a5ce4` (`anthropics/formal-math`, folder `eth`) and [O]'s at commit `7ef5c0c` (`openai/math`, preprint revised 25 October 2026). Logarithms are to base 2.

The labs' own limits. [A], Introduction: "Theorem 1 does not refute the Orthogonal Vectors hypothesis, which SETH implies and on which many lower bounds in fine-grained complexity rest", and "Our result is purely asymptotic". [O], §1.1.3 ("Implications for fine-grained complexity"): lower bounds that "follow from the hypothesis that unrestricted CNF-SAT admits no $2^{(1-\delta)n}\mathrm{poly}(n,m)$-time algorithm" survive, since its theorem "gives a running-time guarantee for each fixed clause width k, while this hypothesis allows the width to grow with n"; and "to obtain a faster algorithm for a target problem from our SAT algorithm, we need a reduction in the other direction". In the remark before its corollary on graph counting, [O] adds that "runtime bounds of the form $2^{o(n)}$ do not transfer under general polynomial-time reductions". For SAT itself Majdoddin [43] and Dias and Lima [29] give negative accounts; this paper gives one for the problems beyond SAT and for non-product decompositions.

2.1. Networks and shared bits.

Definition 2.1. (Networks; [A], Definition 2) A *network* on $[g]$ has, for every node a, a set S_a of settings of at most P bits and a weight table W_a, and for every pair $a \neq b$ an interaction $\Psi_{a,b}$ that factors through $c_{a,b}$ shared bits: $\Psi_{a,b}(z,z') = \sum_{\theta\in\{0,1\}^{c_{a,b}}} f_{a,b}(z,\theta) f_{b,a}(z',\theta)$. Its value is $\sum_z \prod_a W_a(z_a) \prod_{a<b} \Psi_{a,b}(z_a,z_b)$, and its total is $U = \sum_a \sum_{b\neq a} c_{a,b}$, so every shared bit is counted at both ends. The *skeleton* is $(g,(c_{a,b}))$. A *network family* for a counting problem is a map from inputs to networks whose value is the count.

In [A] the tables are given by polynomial-time procedures that see the whole input. The known networks are local: in [A]'s 3-SAT network a node reads the clauses of one block, and in our Hamiltonian-cycle network a node reads the edges at one vertex cluster. Two locality classes carry our theorems.

Definition 2.2. (Model R(s) and Model V(s); `barrier.md` Definitions 5.1, 5.2) Let the input be a matrix $A \in \{0,1\}^{n\times n}$ (or a graph on $[n]$). In *Model R(s)* the rows are partitioned into clusters of size at most s, every node is assigned to one cluster or is a routing node that reads nothing, the skeleton does not depend on A, and the data of a node depend on A only through the rows of its cluster. In *Model V(s)* the $2n$ rows and columns (or the n vertices) are partitioned into clusters of size at most s, and the data of a node depend only on the entries in the rows and columns of its cluster (the edges with an end in it); otherwise as before.

2.2. The engine of [A].

Imported Theorem 2.3. ([A], Lemma 8(i), Theorem A and Theorem B; imported, not used as an assumption) *Let $K \geq 2$, $g = 3K^8$. A network on $[g]$ with parameters P, B, τ and total U is evaluated within $(K^g + T\cdot 2^{2gP/K}\tau)(g(P+B))^{C_1}$ operations, where T bounds the cost of a (K,L)-clique with $L = \lceil (U+gP/K)/K^2 \rceil$ (Lemma 8(i), Lean `ExplicitCost.groupingExplicit`). If pieces with parameters (ρ, c) exist, one algorithm evaluates the (K_ℓ,λ)-clique within $2^{e_\ell K_\ell^2 \lambda}(K_\ell(\lambda+B))^{C_0 K_\ell^2}$ operations, where $\log K_\ell = (3x_0+6)2^\ell - 6\ell - 2x_0 - 6$ and $\frac{1}{2}\rho^\ell \leq e_\ell \leq 0.85\rho^\ell$, $x_0 \geq 18$ (Theorem A, Lean `ExplicitCost.recursionSolves`). Pieces with $\rho = 1/\sqrt{2}$ exist (Theorem B, Lean `PiecesCounted.halfPieces`).*

The (K,λ)-clique $C(K,\lambda)$ is the network on K nodes in which every pair shares one λ-bit label through $\Psi_{w,w'} = [z_w(w') = z_{w'}(w)]$, with arbitrary tables. Its cost is written $2^{e_K K^2 \lambda}$; brute force has $e_K = (K-1)/(2K)$. One level of Theorem A maps a (g^2,λ)-clique, viewed as the $g \times g$ grid of Fig. 1, to networks on g nodes grouped by rows or by columns. The straight edges inside a row or column cost $O(N/g)$ of the $N = g^4\lambda$ shared bits; the rest sits on the $\binom{g}{2}^2$ rectangles. A *family of pieces* writes $d\cdot F = \sum_{i\in I} A_i$ with $|I| \leq (2\lambda)^c$ and certified costs $c_i^{\mathrm{row}}, c_i^{\mathrm{col}}$ with $c_i^{\mathrm{row}} + c_i^{\mathrm{col}} \leq 4\rho\lambda + c\log(2\lambda)$ ([A], Definition 7). Every term $\iota \in I^R$ of the expansion of $\prod_q F_q$ is a row network and a column network, and the level pays the cheaper one, which saves the factor ρ. Since $K_{\ell+1} \geq K_\ell^2$ while e drops by ρ, the rate is $e_K \asymp (\log K)^{-\alpha'}$ with $\alpha' = \log_2(1/\rho)$; $\rho = 1/\sqrt{2}$ gives $\alpha' = 1/2$.

What a shared bit buys. In Fig. 1 a rectangle carries 4λ shared bits at both ends. A piece with $(c^{\mathrm{row}}, c^{\mathrm{col}}) = (2\lambda, 0)$ costs nothing in the column grouping, and at $D = 2$ the split $F = I + (F - I)$ realises the profile $\{(2,0),(0,2)\}$, i.e. $\rho = 1/2$. For larger D the same split does not keep this profile ($F - I$ has row rank $D(D-1)/2$, and [A]'s Theorem B pays $1/\sqrt{2}$ instead. The question of Section 4 is whether any decomposition beats $1/2$; the question of Section 3 is how many rectangles, that is how many shared bits, a dense problem forces.

2.3. **The engine of [O].** [O]'s Theorem 1.1 counts the solutions of a k-CNF with m clauses in time $(n+m)^{O_k(1)} 2^{O_k(n \log\log n/\sqrt{\log n})}$ and polynomial space. Its Engine I needs a *layout* ([O], Definition `def:layout`; Lean `LayoutSpec.Layout`): partitions of the variables at Q interpolation points, schedules for the factors, measurement matrices of small ℓ_1 height, and for every occurrence a row polynomial $P_{f,v}(T)$ of degree at most R with $P_{f,v}(0) = \delta_v$ that annihilates the measurement kernels of the other groups. The cost exponent is about $n[(R/Q)\log(2Q/R) + \tau \log(2D+1) + 2/r]$. [O]'s construction (its §4) is a template: a matrix M_0 acting on labels in $[r]^Q$, with support condition (T2), $M_0[u,w] \neq 0 \Rightarrow d_H(d_u, a_w) \leq R - 1$. Section 5 bounds every instance of this template.

2.4. **The engine exponent.**

Proposition 2.4. (on paper; `barrier.md` Lemmas 2.0, 2.1, 2.2) *Let N be a network with total $U \geq 1$ to which Lemma 8 is applied with any admissible K and the clique algorithm of Theorems A(b) and B. Then the guaranteed running time is at least $2^{E(N)}$ with*

$$E(N) \;\geq\; e_K U + 2gP/K \;\geq\; \frac{U}{2\sqrt{\log U + 2x_0}}.$$

So the guarantee is below 2^n only if $U < (2 + o(1))\, n\sqrt{\log n}$, and with pieces of saving $\rho \geq 1/2$ it is at least $2^{U/(2(\log U + 2x_0))}$. Any algorithm for the (K,λ)-clique that reads its tables has $e_K \geq (K-1)/K^2 - o(1)$, which alone gives only $E(N) \geq \sqrt{2U}(1 - O(1/K))$.

The proof reads $e_K \geq 1/(2\sqrt{\log K + 2x_0})$ off Theorem A(a), notes $K^2 L \geq U$ and $2gP/K \geq 2K$, and splits on whether $2K$ exceeds the bound. The query bound $(K-1)/K^2$ is Majdoddin's input floor [43, Proposition 2.4(a)] in our normalisation (his exponent is per edge bit, of which the clique has $\binom{K}{2}\lambda$). The last sentence is the reason Section 4 matters: with a hypothetical optimal clique solver, no lower bound on U of order $n \log n$ would bar anything. This proposition is about [A]'s guarantee as a function of U, for all inputs; it is one of the three places (with Section 5 and Theorem 3.11) where a statement concerns the engines as published rather than a class of networks.

3. Dense counting costs $n \log n$ bits in every local network we can analyse

In this section we prove the lower bounds on U. We first state the cut lemma in rank form (Section 3.1), then compute the rank of the permanent across a row cut (Section 3.2), and sum the cuts in Model R (Section 3.3) and Model V (Section 3.4). Section 3.5 treats algebras and characteristic p, Section 3.6 the Tutte plane, and Section 3.7 says why no such bound exists in the general model.

3.1. **The cut lemma.** Let N be a network and (L, R) a partition of its nodes, with $w(L,R) = \sum_{a \in L, b \in R} c_{a,b}$ crossing bits. Replacing every crossing interaction by its factorisation and exchanging the sums gives $\mathrm{value}(N) = \sum_{\theta \in \{0,1\}^w} F_L(\theta) F_R(\theta)$, where F_L depends only on the data of the nodes in L (`barrier.md` Lemma 3.1, on paper). So if the nodes in L read only a part X of the input and those in R only the rest Y, the matrix $[\mathrm{value}(N(X,Y))]_{X,Y}$ factors through 2^w terms. The rank form is machine-checked.

Theorem 3.1. (`EthFollowup.Barriers.Rank.rank_le_card_of_eq_sum_mul`, `EthFollowup.Barriers.Rank.rank_le_two_pow_of_eq_sum_mul`, `EthFollowup.Barriers.Rank.rank_eq_card_of_linearIndependent`; machine-checked) *Let K be a field and X, Y, Θ types with Y and Θ finite.*[1] *If $M[x,y] = \sum_{\theta \in \Theta} F(x,\theta) G(\theta, y)$ for all x, y, then $\operatorname{rank} M \leq |\Theta|$; for $\Theta = \{0,1\}^w$ this is $\operatorname{rank} M \leq 2^w$. If moreover $M[x,y] = \sum_{t \in T} u_t(x) v_t(y)$ over a finite T with both families (u_t), (v_t) linearly independent (X, Y finite), then $\operatorname{rank} M = |T|$.*

[1] Module `lean/EthFollowup/Barriers/RankBounds.lean`.

Proof. Write $M = \mathcal{F}\mathcal{G}$ with $\mathcal{F} = [F(x,\theta)]$ and $\mathcal{G} = [G(\theta,y)]$; the rank of a product is at most its inner dimension $|\Theta|$. For the second statement, $M = UV^{\mathsf{T}}$ with U and V of full column rank $|T|$. □

3.2. **The permanent's row cut.**

Theorem 3.2. (`EthFollowup.Barriers.RowCut.rowCut_rank`, `EthFollowup.Barriers.RowCut.rowCut_rank_oneHot`, `EthFollowup.Barriers.RowCut.permanent_joinMat`; machine-checked) *Let K be a field, ι a finite type with n elements and $I \subseteq \iota$ with $|I| = k$.*[2] *Let M_I be the matrix whose rows are indexed by the 0/1 rows $X : I \to \iota \to \{0,1\}$, whose columns are indexed by $Y : I^c \to \iota \to \{0,1\}$, and whose entry is $\mathrm{per}(X;Y)$, the permanent of the matrix with rows I read from X and the other rows from Y. Then $\operatorname{rank} M_I = \binom{n}{k}$, and the same holds for the submatrix on one-1-per-row matrices. The proof is the Laplace expansion $\mathrm{per}(X;Y) = \sum_{|S|=k} U_S(X)V_S(Y)$ along the rows I.*

Proof. A permutation of ι maps I onto some S with $|S| = k$, which gives the Laplace expansion. Evaluated at the one-1-per-row matrix X that matches I onto S_0, $U_S(X) = [S = S_0]$, so the U_S are linearly independent, and likewise the V_S; Theorem 3.1 gives rank exactly $\binom{n}{k}$, already on the one-1-per-row family. □

An aside, not part of the statement above: on one-1-per-row matrices the permanent $\mathrm{per}(X;Y)$ is the indicator that $(X;Y)$ is a permutation matrix, so on that family the rank statement is about this indicator.

3.3. **Model R: $U \geq n\log(n/s)$.** For a cluster j of Model R(s), cut the nodes into those of cluster j and the rest. The left side reads the rows V_j and the right side the other rows, so **??** 3.1**??** 3.2 give $\binom{n}{|V_j|} \leq 2^{w_j}$ with w_j the bits crossing that cut; this step, from a network to a factorisation, is on paper. Each ordered pair of nodes is counted in at most one w_j, which is machine-checked, and the rest is arithmetic.

Theorem 3.3. (`EthFollowup.Barriers.Rank.modelR_bound_nlogn`, `EthFollowup.Barriers.Rank.modelR_bound`, `EthFollowup.Barriers.Rank.sum_crossing_le_total`; machine-checked with hypotheses `hrank` and `hU` on paper) *Let J be a finite set of clusters, $n, s, U \in \mathbb{N}$, and $k, w : J \to \mathbb{N}$ with $1 \leq k_j \leq s$ and $\sum_j k_j = n$. Assume **`hrank`:** $\binom{n}{k_j} \leq 2^{w_j}$ for every j (the cut lemma with the row cut, on paper for a given network), and **`hU`:** $\sum_j w_j \leq U$. Then*

$$n\log_2(n/s) \;\leq\; \sum_j \log_2 \binom{n}{k_j} \;\leq\; U.$$

*The second inequality needs only **`hrank`** and **`hU`** (**`modelR_bound`**). For **`hU`**: for every assignment* cl *of nodes to clusters or routing and every c, $\sum_j \sum_{a,b} [\mathrm{cl}(a) = j \neq \mathrm{cl}(b)]\, c_{a,b} \leq \sum_{a \neq b} c_{a,b}$ (**`sum_crossing_le_total`**), so **`hU`** holds with w_j the crossing bits of cluster j.*

Assembled on paper (`barrier.md` Theorem 6.1): every network family for per in Model R(s) has $U \geq n\log_2(n/s)$, already if the value is correct only on one-1-per-row matrices; with Theorem 2.4, for $s \leq n^{1-\varepsilon}$ the engine's guarantee on any such network is $2^{\omega(n)}$ (`barrier.md` Corollary 6.3). This is the analogue of Nisan's bound for noncommutative branching programs [46], with row-locality in place of the variable order.

3.4. **Vertex-local networks: trees and the single-cut limit.** In Model V a node also reads the columns of its cluster, and the row-cut argument breaks: each side sees part of the other's input. A vertex cut with k rows and k columns on the small side has rank exactly $\binom{2k}{k}$ for a suitable fixing of the shared entries, by the Vandermonde identity.

Theorem 3.4. (`EthFollowup.Barriers.Rank.treeSkeleton_bound`, `EthFollowup.Barriers.Rank.sum_choose_mul_choose`; machine-checked with hypothesis `hrank` on paper) *Let E be a finite set of tree edges and $U \in \mathbb{N}$, $c, k, k' : E \to \mathbb{N}$, with $U = 2\sum_e c_e$. Assume **`hrank`:** $\binom{k_e + k'_e}{k_e} \leq 2^{c_e}$ for every edge e (the cut lemma with the vertex-cut rank, on paper). Then $2\sum_e \log_2 \binom{k_e+k'_e}{k_e} \leq U$. Also $\sum_{i=0}^{k} \binom{k}{i}\binom{k'}{i} = \binom{k+k'}{k}$ for all k, k'.*

Proposition 3.5. (on paper; `barrier.md` Theorem 6.2 and Proposition 3.5, `vertex-local-bound.md` Theorems 2.1, 3.1, Corollary 3.2, Propositions 4.1, 4.2) *Every network family for* per *in Model V(s) has $U \geq 2n - o(n)$ for $s \leq n^{1/3}$. No single cut gives more: for a vertex cut with k rows and k columns on the small side, every fixing of the other entries gives rank at most $\binom{2k}{k} < 4^k$. If the skeleton is a tree of maximum*

[2] Module `lean/EthFollowup/Barriers/RowCut.lean`.

degree Δ with the clusters $\{i, i'\}$, then $U \geq (4/\log_2(\Delta-1))\, n \log_2 n - O(n \log\log n)$. On the clusters $\{i, i'\}$, no star skeleton with one shared bit per leaf computes per *for $n \geq 3$. The function $[A$ is a permutation matrix$]$ has a Model $V(1)$ network with $U = 0$ but needs $U \geq n \log(n/s)$ in Model $R(s)$, so the one-1-per-row family gives no bound in Model V.*

The star statement uses Cayley's hyperdeterminant and is checked numerically for $n = 3, 4$; the Catalan cut rank of the family $\{I + P_\sigma\}$ is checked for $b \leq 5$.

Conjecture 3.6. (Model V; `barrier.md` Conjecture 6.5) *Every network family for* per *in Model $V(s)$ has $U \geq (1 - o(1))\, n \log_2(n/s)$. Proved: $2n - o(n)$ for $s \leq n^{1/3}$, and $(4/\log_2(\Delta - 1) - o(1))\, n \log_2 n$ for trees of bounded degree; the star with two bits per leaf is the first skeleton on which every cut passes.*

3.5. **Substitution algebras and characteristic p.** A different way to evaluate per is to substitute ring elements for the entries and read a coefficient, as Clifford, group-algebra and Kronecker tricks do.

Theorem 3.7. (`EthFollowup.Barriers.AlgebraRank.algebra_rank_half`, `EthFollowup.Barriers.AlgebraRank.algebra_rank_sum`; machine-checked) *Let K be a field and R a finite-dimensional K-algebra, not necessarily commutative.*[3] *Let $\varphi : R \to K$ be K-linear and $r_{ij} \in R$ $(i, j < n)$ with $\varphi(r_{0f(0)} r_{1f(1)} \cdots r_{n-1,f(n-1)}) = [f$ bijective$]$ for every $f : [n] \to [n]$. Then $\binom{n}{\lfloor n/2 \rfloor} \leq \dim_K R$. For finitely many such algebras A_t with $\sum_t \varphi_t(r^{(t)}_{0f(0)} \cdots r^{(t)}_{n-1,f(n-1)}) = [f$ bijective$]$, $\binom{n}{\lfloor n/2 \rfloor} \leq \sum_t \dim_K A_t$.*

Proof. For $|S| = n/2$ let u_S be the product of the first $n/2$ factors along a fixed bijection onto S, and v_S the product of the last $n/2$ along a bijection onto the complement. Then $\varphi(u_S v_{S'}) = [S = S']$, so the $\binom{n}{n/2}$ elements u_S are linearly independent in R. For several algebras apply this to the product algebra with the functional $\sum_t \varphi_t$. □

Theorem 3.8. (`EthFollowup.Barriers.PermModP.permanent_eq_zero_of_low_rank`; machine-checked) *Let R be a commutative ring of characteristic p, p prime, and ι, κ finite types.*[4] *If $A = UV^{\mathsf{T}}$ with $U, V \in R^{\iota \times \kappa}$ and $|\kappa|(p-1) < |\iota|$, then* $\operatorname{per} A = 0$.

Proof. Expanding $A = UV^{\mathsf{T}}$ column by column gives $\operatorname{per} A = \sum_{\ell : \iota \to \kappa} \big(\prod_i V_{i\ell(i)}\big) \operatorname{per} U_\ell$, where column i of U_ℓ is column $\ell(i)$ of U. If $|\kappa|(p-1) < |\iota|$, some value of ℓ is taken at least p times, so U_ℓ has p equal columns; a p-cycle permuting them acts freely on the terms of $\operatorname{per} U_\ell$ and leaves each orbit sum p times one term, which is 0 in characteristic p. □

So every saving that characteristic p gives through low rank dies once p is of order n, and the Chinese remainder theorem needs primes up to about $0.9\, n \ln n$.

Proposition 3.9. (on paper; `wild-ideas.md` F1, `perm-mod-p.md` Lemma B4) *If $\operatorname{per}(A) = \sum_{t \leq T} w_t \, \#\mathrm{SAT}(F_t)$ for all 0/1 matrices A, with integers $|w_t| \leq 2^W$, then some F_t has at least $\log_2 n! - W - \log_2 T = n \log_2 n - 1.45n - W - \log_2 T$ variables. The same holds modulo any M with $\log_2 M \geq \log_2 n! + 2$.*

So plain #SAT encodings of per have $n \log n - O(n)$ variables, where the Lean #k-SAT bounds of both labs cost $2^{\Theta(n\sqrt{\log n} \log\log n)}$ or more.

3.6. **The Tutte plane off the hyperbolas.** Write $q = (x-1)(y-1)$, $H_q = \{(x, y) : (x-1)(y-1) = q\}$ and $\mathcal{H} = \bigcup_{q \in \{1,2,\dots\}} H_q$, $\mathcal{L} = \{y = 1\}$. On $\mathcal{H} \cup \mathcal{L}$ the engine reaches the Tutte polynomial of sparse graphs, conditionally on [A] and informally (`tutte-plane.md` Theorems P1, P2; not claimed here). Off them, the information is the connectivity pattern of the private edges, and the shared edges can be fixed to a matching.

Proposition 3.10. (on paper; `tutte-plane.md` Theorem B, Lemmas 3.2–3.4, Proposition 3.5) *Let $y \neq 1$ and $(x-1)(y-1) \notin \{1, 2, \dots\}$. Let N be a Model $V(s)$ family on subcubic graphs (maximum degree 6 at $x = 0$), with a fixed skeleton and $s \leq \lceil n/\log_2 n \rceil$, that computes $c(n, m, k(G))\, T_G(x, y)$ for some nonzero normalisation c. Then $U \geq \sum_{groups} \log_2 \mathrm{Bell}(b_G) \geq (1 - o(1))\, n \log_2 n$. For a sum of N_{net} such networks on one common clustering, some summand has $U \geq (\frac{1}{2} - o(1))\, n \log_2 n - 2 \log_2 N_{\mathrm{net}}$.*

[3] Module `lean/EthFollowup/Barriers/AlgebraRank.lean`.

[4] Module `lean/EthFollowup/Barriers/PermModP.lean`.

The ingredients are Lindström's factorisation [41, 66] $[q^{|P\vee Q|}]_{P,Q\in\Pi_b}=\zeta\,\mathrm{diag}((q)_{|S|})\,\zeta^{\mathsf T}$, whose rank over $\mathbb{Q}$ is $\mathrm{Bell}(b)$ for $q\notin\{0,\dots,b-1\}$ and $\sum_{j\le q}S(b,j)$ otherwise (the connection rank of [30, 42]); linear-forest gadgets that realise every boundary partition with triangular gluing vectors; and a deletion–contraction step that replaces forests by connected subcubic graphs. For $v=y-1\neq 0$ the gluing matrix is $[q^{|P\vee Q|}]$ up to invertible diagonal scalings, so the rank depends on (x,y) only through q and on whether $v=0$; at $q=0$ (that is, $x=1$) the matrix $[P\vee Q=\hat1]$ takes its place and has rank $\mathrm{Bell}(b)$ by Möbius inversion, and on $\mathcal{L}$ ($v=0$) the acyclic matrix of Theorem 8.5 does.

Corollary 3.11. (on paper; `tutte-plane.md` Corollary B′) *Under the hypotheses of Theorem 3.10,* [A]*'s guarantee is at least* $2^{(1/2-o(1))n\sqrt{\log n}}$*, and for any pieces with* $\rho\ge 1/2$ *it is at least* $2^{(1/2-o(1))n}$.

Remark 3.12. (checked numerically) The ranks of $[q^{|P\vee Q|}]$ were computed exactly for $b\le 6$, the cut ranks of the test families for $b\le 4$ at fifteen points, and the acyclic gluing rank on $\mathcal{L}$ equals the Catalan number for $b\le 6$ and is at least $\mathrm{Cat}(7)$ at $b=7$. That it equals $\mathrm{Cat}(b)$ for all b is a conjecture.

Two loopholes are open: sums of networks over many *different* clusterings (the analogue of sums of read-once branching programs in different orders), and skeletons that depend on private edges. Nothing in Theorem 3.10 covers either. The matching upper bound holds everywhere: at every point $Z_G(q,v)$ is a Lagrange combination of $n+1$ colour networks with $U\le 2m\log_2(n+1)+g^2$ (`tutte-plane.md` Proposition P3, informal and conditional), so the minimal vertex-local resource is $\Theta(n\log n)$ off $\mathcal{H}\cup\mathcal{L}$ and the whole question is the rate e_K.

3.7. The general model admits no unconditional bound.

Proposition 3.13. (on paper; `barrier.md` Proposition 4.1) *In* [A]*'s Definition 2, with tables given by polynomial-time procedures on the whole input:*

(a) a network for per *with* $g=1$*, one setting,* $U=0$ *and* $\tau=\mathrm{poly}(n)$ *exists if and only if* $\mathrm{per}\in\mathrm{FP}$*; so an unconditional lower bound* $U\ge 1$ *in the general model would imply* $\mathrm{P}\neq\mathrm{P}^{\#\mathrm{P}}$*;*

(b) Ryser's formula [51] *is a network with* $g=1$*,* $U=0$ *and* $P=n$*;*

(c) "a network with $U+\sum_a P_a+\log\tau=o(n)$*" is equivalent to "*$\mathrm{per}\in\mathrm{TIME}(2^{o(n)})$*".*

So in the general model the barrier question is the open algorithmic question itself. The conditional lower bound of Dell, Husfeldt, Marx, Taslaman and Wahlén [28] (no $2^{o(n)}$ algorithm for the permanent under #ETH) cannot be invoked, since the same engines refute #ETH. What we bound is the information that a local network routes.

4. The rate is capped in every analysed model

In this section we bound the saving ρ of one level, and hence the rate exponent $\alpha'=\log_2(1/\rho)$, in four kinds of model of increasing generality: pieces (Section 4.1), graded two-order pieces (Section 4.2), recursions that take the cheaper of several groupings (Section 4.3), and correlated decompositions that need not be products (Section 4.4). The first three have product terms, and there Majdoddin's rewrite-and-merge class [43] contains our models with sharper constants; we include them because their rank statements are machine-checked here, because the covering bound for arbitrary families of groupings is not in [43], and because Section 4.4 reuses their contraction. Section 4.5 says what the caps mean for the engine. Throughout, $N=D=2^\lambda$ is the alphabet of one crossing equality.

4.1. Pieces have $\rho\ge 1/2$.

Contract the legs y and u of a piece with vectors β,a. On F this gives $\langle\beta,a\rangle$ times the identity on (x,v), and for any piece the contraction is a submatrix of both flattenings, so its rank is at most the smaller of the two.

Theorem 4.1. (`EthFollowup.Barriers.Rank.pieces_trivial_lower_bound`, `EthFollowup.Barriers.Rank.pieces_exists_large_piece`; machine-checked) *Let* K *be a field,* L *a finite type and* $T_i:L^4\to K$ *(*$i\in I$*,* I *finite) with* $\sum_i T_i(x,y,u,v)=d\cdot[x=v][y=u]$ *for all* x,y,u,v *and some* $d\neq 0$*. Then*

$$|L|\;\le\;\sum_{i\in I}\min\bigl(\operatorname{rank}T_i^{(xy|uv)},\;\operatorname{rank}T_i^{(xu|yv)}\bigr),$$

and if $I\neq\emptyset$ *some piece has both flattening ranks at least* $|L|/|I|$.

Proof. Contract y and u with a common basis vector. The target becomes d times the identity on (x, v), of rank $|L|$, and the contraction of each piece is a submatrix of both of its flattenings. Subadditivity of rank gives the first claim and pigeonhole the second. □

The bound $\rho \geq 1/2$ for pieces is not new: it is the observation of [A]'s Appendix H that one piece needs $2\lambda - O(\log \lambda)$ bits row-wise, and Majdoddin's pigeonhole and two-merge floors [43, Propositions 4.6 and 4.8] are its general form, for every configuration. What Theorem 4.1 adds is the Lean check of the rank form, and a contraction that bounds $\min(\mathrm{rank}^{\mathrm{row}}, \mathrm{rank}^{\mathrm{col}})$ of each term separately, so that no mixed term is needed; Section 4.4 uses exactly this.

Theorem 4.2. (`EthFollowup.Barriers.Rank.rank_sum_le`, `EthFollowup.Barriers.Rank.exists_logb_rank_ge`; machine-checked) *For a field K and matrices A_s ($s \in I$, I finite) over K of a common shape,* $\mathrm{rank} \sum_s A_s \leq \sum_s \mathrm{rank}\, A_s$. *If $I \neq \emptyset$ and* $\mathrm{rank} \sum_s A_s = 2^{2\lambda}$, *some s has* $\log_2 \mathrm{rank}\, A_s \geq 2\lambda - \log_2 |I|$.

With certified costs at least $\log_2$ of the ranks and $|I| \leq (2\lambda)^c$, this gives $\rho \geq 1/2 - (3c/4)\log_2(2\lambda)/\lambda$ in [A]'s Definition 7 (`rho-lower-bound.md` Corollary 2.2, on paper). The rank inequalities that a piece family must satisfy are themselves satisfiable at $\rho = 1/2$ (`rho-lower-bound.md` §2.2), so no counting argument on ranks alone beats $1/2$, and the simplest profiles $(N^2, 1) + (1, N^2) + \cdots$ are excluded only by a finer argument (a piece of operator-Schmidt rank 1 forces the others to carry rank at least $N(N-1)/10$, `rho-lower-bound.md` Proposition 2.3, on paper).

4.2. **Graded two-order pieces have $\rho \geq 1/\sqrt{2}$.** [A]'s Theorem B builds its pieces from a graded space $V = \bigoplus_E V_E$: a piece is a slice Q_s of a locality identity of order H, and its certified cost is the dimension of the spaces the products pass through, "max over the terms". Write $D(E) = \dim V_E$ and $f = \log_2 D$.

Theorem 4.3. (`EthFollowup.Barriers.Spread.spread_lemma`, `EthFollowup.Barriers.Spread.spread_slice`, `EthFollowup.Barriers.Spread.spread_slice_zero`; machine-checked) *Let $D : \mathbb{Z} \to \mathbb{N}$, $h \in \mathbb{Z}$, T a finite set of terms with degrees r_j, and $R^{\mathrm{row}} = \sum_{j \in T} D(h + r_j)$, $R^{\mathrm{col}} = \sum_{j \in T} D(h - r_j)$.*[5] *With $f = \log_2 D$ (and $f = 0$ where $D = 0$),*

$$f(h + r_{\max}) + f(h - r_{\min}) \;\leq\; \lceil \log_2 R^{\mathrm{row}} \rceil + \lceil \log_2 R^{\mathrm{col}} \rceil,$$

where $r_{\max} = \max\{r_j : D(h + r_j) > 0\}$ and $r_{\min} = \min\{r_j : D(h - r_j) > 0\}$, both sets nonempty. For the slice s of a locality identity of order H with degrees $r_i = s - h_0 + i$, $i = 0, \ldots, H$, the right side is at least $f(h - h_0 + s + H) + f(h + h_0 - s)$, with no sign hypothesis; for $h_0 = h = h_n \in \mathbb{N}$, $H = 2h_n$ and $s = 0$ it is at least $2f(2h_n)$.

The operators and the two-order identity are not modelled; the theorem is about the dimension counts only, and the certified costs $\lceil \log_2 R^{\mathrm{row}} \rceil$ are those of [A]'s Lemma 3.2 (Lean `Nat.clog 2 (Rrow h s)` in its statement of Theorem B).

Proposition 4.4. (on paper; `rho-lower-bound.md` Corollaries 3.4, 3.5, Lemma 4.1, Propositions 5.1, 5.2; `sparse-fock.md` Theorem 2.1) *For graded two-order decompositions with spread $H = 2h$ and $\lambda \leq f(h)$, the saving is at least $f(2h)/(2f(h)) - c\log(2\lambda)/(4\lambda)$. On Fock space ($f = \log$ of the partition function, with the Hardy–Ramanujan bounds) this is $\rho \geq 1/\sqrt{2} - O((c+1)\log\lambda/\lambda)$, and the same holds in every C_2-cofinite vertex operator algebra. The spread $2h$ follows from the pairing being the top pole. Conversely, every family of pieces is a graded two-order decomposition on a sparse Fock space at two extra bits per piece, so the question "is $\rho^* < 1/\sqrt{2}$?" is the general pieces question.*

De Brito [25] proves, after [A]'s Appendix H, a rank floor $|\mathcal{X}| \cdot \dim V_h \leq T_V(h)$ for families of the shape of [A]'s Theorem B, which closes the case of growth exponent $\beta < 1/2$ for that shape and gives $\rho \geq \max(2^{\beta-1}, 2^{-\beta}) \geq 1/\sqrt{2}$. Majdoddin [43, Proposition 4.24 and the remark after it] proves $\rho \geq 2/3 - o(1)$ for every concave profile in that shape and the same $\max(2^{\alpha-1}, 2^{-\alpha})$ for power profiles $D(E) = cE^{\alpha}$. Our corollary, in a note dated 28 October, gave the branch $\rho \geq 2^{\beta-1}$ and left $\beta < 1/2$ open; what Theorem 4.4 adds is the C_2-cofinite generality and the converse. Outside the shape, "$\rho^* < 1/\sqrt{2}$?" is open.

Remark 4.5. (checked numerically) For $h \leq 10$ the true ranks of the Theorem B pieces are 93–97% of the certified counts wherever the count is below N^2. This says nothing about the asymptotic ρ.

[5] Module `lean/EthFollowup/Barriers/SpreadLemma.lean`.

4.3. **Any min-over-groupings recursion has $\alpha' \le 1$.**

Proposition 4.6. (on paper; `clique-evaluation.md` §4.3) *Let a level write each rectangle as $F = \sum_{s\in I} A_s$ with $|I| = 2^{o(\lambda)}$, expand the product over the rectangles, and evaluate every term in the cheaper of the row and column groupings. Then $\rho \ge 1/2 - o(1)$, and with $K_{\ell+1} \ge K_\ell^2$ the recursion has $e_K \ge c/\log K$, for any pieces, any algebra and any subexponential term count.*

The proof uses Theorem 4.2: some piece s_r has row cost at least $2\lambda - o(\lambda)$ and some s_c column cost at least $2\lambda - o(\lambda)$; the mixed term using s_r on half of the rectangles and s_c on the other half costs $(1/2 - o(1))N$ in both groupings. So it is the schedule, not the pieces, that is capped. Majdoddin [43, Theorem 4.13 and Corollary 4.19] proves the sharper statement that every grid algorithm has exponent at least $1/\log_2 K$ per edge bit, with the exact constant and without assuming $K_{\ell+1} \ge K_\ell^2$; our note of 29 October [60] has the weaker form above.

Proposition 4.7. (on paper; `engine2-probabilistic.md` Theorem A, proof as repaired in §7.3) *Let a level write each of the $|R| = \binom{g}{2}^2$ grid rectangles as $F = \sum_{s\in I} A_s$ with $|I| = 2^{o(\lambda)}$ and expand the product over the rectangles. Let $\mathcal{G}$ be any family of M partitions of the g^2 grid nodes into blocks of size at most g, and let every term choose its grouping from $\mathcal{G}$, adaptively or at random. Then the level saving satisfies $\rho_{\text{eff}} \ge 1/2 - 1/(g-1) - \sqrt{\ln(2M)/(2|R|)} - o(1)$; for all partitions, $\rho_{\text{eff}} \ge 1/2 - O(\sqrt{\log g}/g)$. So $\alpha' \le 1$ for every level of this kind that expands into $2^{o(N)}$ terms and picks the best grouping.*

Proof. Call a rectangle with corners x, y, u, v (Fig. 1) *diagonal* under a partition π if x, v or y, u lie in one block of π. A pair of nodes in one block, in different rows and columns, is a diagonal of exactly one rectangle, so at most $\sum_B \binom{|B|}{2} \le g\binom{g}{2} = 2|R|/(g-1)$ rectangles are diagonal. A rectangle that is not diagonal is *row-like* if x, y or u, v share a block, *column-like* if x, u or y, v do, and *four-block* otherwise; it cannot be both row-like and column-like, since for instance $x \sim y$ and $x \sim u$ would put y, u in one block. For a row-like rectangle the row cut $(xy \mid uv)$ is a union of blocks restricted to its corners, so the bonds of the evaluated network that cross it carry at least the row log-rank of its piece; likewise for column-like ones, and a four-block one pays both. By Theorem 4.2 there are pieces s_r, s_c of row, respectively column, log-rank at least $2\lambda - o(\lambda)$. Draw $\iota \in \{s_r, s_c\}^R$ uniformly. For a fixed π, each row- or column-like rectangle pays its full cost with probability $1/2$, independently, and each four-block one always does, so Hoeffding's inequality and a union bound over the M partitions give a term that pays at least $(1 - o(1))\, 2\lambda\big(\frac{1}{2}(|R| - 2|R|/(g-1)) - \sqrt{|R|\ln(2M)/2}\big)$ under every $\pi \in \mathcal{G}$. For all partitions $\ln M \le g^2 \ln g^2$. □

The configurations here are the grid rectangles with per-rectangle product identities. Majdoddin leaves three or more merges open in his class ([43, §4.8 and Open Problem 4]); for rectangles, whose relevant cuts are the row, column and diagonal ones, Theorem 4.7 treats any number of groupings. The first version of the note charged every rectangle that meets at least three blocks both of its costs, which is right only for four-block rectangles; the proof above reclassifies the others (and the constant is $1/(g-1)$, not $1/g$, since $|R| = \binom{g}{2}^2$ exactly).

Proposition 4.8. (on paper; `engine2-algebraic.md` Theorems 7.1, 7.3, Corollary 7.4) *For gadget recursions with m groupings whose gadgets meet exactly two classes in every grouping, the saving satisfies $\rho \ge E[1/d_\gamma] - o(1)$ over the number d_γ of distinct cuts of a gadget, and, under homogeneity and regularity with exceptional fraction $\delta < 1/(d-1)$, the node count grows as $K' \ge K^{d-o(1)}$. Hence $\alpha' \le 1 + O(\delta/\ln d)$: each additional distinct cut costs one factor K in the node count.*

Majdoddin argues the same for line merges of the d-dimensional grid [43, §4.8]; neither statement implies the other.

Other mechanisms fail for exact reasons (`engine2-algebraic.md` M1–M5, M8, on paper): algorithms based on tensor rank have $e \ge 1/4$ (Majdoddin's flattening floor [43, Proposition 2.4(b)] in our normalisation); node-local invertible transforms leave every rank-based cost unchanged; by the Chinese remainder theorem e_K over $\mathbb{Z}$ is the maximum over primes; monomial pieces in any product basis have $\rho \to 1$; the best contraction tree has $e = \min(\omega/9, 1/4) + O(1/K)$; and recursion on the bits of λ gives nothing.

4.4. **Correlated decompositions: N^2, N^3, N^8.** All the caps above assume that the terms form a full product $\prod_q I_q$ over the rectangles, which is what lets the mixed term exist. Majdoddin's class makes the

same assumption (his (L2)), and he notes that summing terms over a set of choices that is not a product set leaves the class ([43, Remark 4.5]). A recursion could instead write the identity of a whole level as $\sum_{t\in\mathcal{C}} T_t$ over a correlated set of terms, each a row network or a column network. For such a term, the flattening across any union of rows has rank at most 2^{cost} (`engine2-decorrelated.md` §1, on paper), and the *rank sum* $S=\sum_t 2^{\mathrm{cost}_t}$ satisfies $\max_t \mathrm{cost}_t \ge \log_2 S - \log_2|\mathcal{C}|$. So $S \ge N^Q$ on a configuration of Q rectangles is $\rho \ge 1/2 - o(1)$ there. The trick is one contraction that is a submatrix of a row flattening and of a column flattening at the same time.

Theorem 4.9. (`Decorrelated.rankSum_ge`, `Decorrelated.exists_expensive`; machine-checked) *Let K be a field and α a nonempty finite alphabet.*[6] *For every finite family of tensors T_t on the eight legs $x_1y_1u_1v_1x_2y_2u_2v_2$ with $\sum_t T_t = [x_1=v_1][y_1=u_1][x_2=v_2][y_2=u_2]$,*

$$|\alpha|^2 \;\le\; \sum_t \min\bigl(\operatorname{rank} T_t^{(x_1y_1x_2y_2|u_1v_1u_2v_2)},\ \operatorname{rank} T_t^{(x_1u_1x_2u_2|y_1v_1y_2v_2)}\bigr),$$

and some term has this minimum at least $|\alpha|^2/\#\text{terms}$.

Proof. Fix $y_1=u_1=y_2=u_2=o$. The contraction of $\sum_t T_t$ is the identity on $(x_1,x_2)\mid(v_1,v_2)$, of rank $|\alpha|^2$, and the contraction of each T_t is a submatrix of both flattenings. Subadditivity finishes. □

The Lean statement is one statement about two fixed cuts; its file header calls the configuration "C" (disjoint row pairs and column pairs). The two-class cuts of configurations A (one cell shared) and B (shared row pair) are the same two cuts with the legs assigned to rows and columns differently, and the chains D, E reduce to them by the symmetry $(x,y,u,v)\mapsto(v,u,y,x)$ of F; both readings are on paper. So for every pair of rectangles, correlation buys nothing: $\max_t \mathrm{cost}_t \ge 2\lambda - \log_2 T$.

The running example, twice. Return to Fig. 1 at $D=2$ and take two rectangles. The product of the splits $F = I + (F-I)$ gives four terms $I\otimes I$, $I\otimes(F-I)$, $(F-I)\otimes I$ and $(F-I)\otimes(F-I)$, whose minima of the two flattening ranks are $1,4,4,1$: rank sum 10 and maximum cost 2 bits, which is $Q\lambda$. Theorem 4.9 says that no decomposition, correlated or not, has rank sum below $N^2=4$. A correlated decomposition can repack constants (exhaustive search finds rank sum 7, Theorem 4.14), but it cannot change the exponent, for any λ. At $\lambda=1$ the term count is not $o(\lambda)$, so the example illustrates the bound and is no evidence about ρ.

Theorem 4.10. (`Grid32.rankSum_ge`, `Grid32.exists_expensive`, `Grid32.H_eq_rowTwist_T0`, `Grid32.rank_colFlat_T0`; machine-checked) *Let K be a field and α a nonempty finite alphabet.*[7] *Let H be the tensor of the three rectangles ab, ac, bc on one column pair of the 3×2 grid (12 legs). For every finite family with $\sum_t T_t = H$,*

$$|\alpha|^3 \;\le\; \sum_t \min\bigl(\operatorname{rank}\ \mathrm{flat}_{b|ac} T_t,\ \operatorname{rank}\ \mathrm{flat}_{j|j'} T_t\bigr),$$

and some term has this minimum at least $|\alpha|^3/\#\text{terms}$. Moreover H is a row-local relabelling of a tensor T_0 whose column flattening has rank at most 1.

The proof pairs the two legs at one node and fixes the bc legs; one leg straddles the row cut and one the column cut, each at a factor N, while the target gains N^2. The last sentence is a no-go: any subadditive functional that is invariant under row-local relabellings and bounded on cost-0 column networks takes the same value on H as on a cost-0 term, so it proves nothing.

Theorem 4.11. (`Lossy.rank_contr_le`, `Lossy.rank_contr_graph`, `Grid33.rankSum_ge`, `Grid33.exists_expensive`; machine-checked) *Let K be a field, α a finite alphabet and L a finite set of legs.*[8] *For disjoint $K_X, K_Y \subseteq L$, a cut $S\subseteq L$ and a tensor $T:\alpha^L\to K$, the contraction keeping K_X as rows, K_Y as columns and fixing the other legs satisfies* $\operatorname{rank}(\operatorname{contr} T) \le |\alpha|^{|K_X\cap S|+|K_Y\setminus S|}\cdot\operatorname{rank}(\mathrm{flat}_S T)$*; for a graph tensor whose kept edges match K_X with K_Y and whose other edges are fixed, the contraction is a permutation matrix of rank $|\alpha|^{|K_X|}$. Let G be the tensor of the full 3×3 level (the graph tensor of the 18 crossing edges of $K_3\times K_3$, 36 legs). If α is nonempty, then for every finite family with $\sum_t T_t = G$,*

$$|\alpha|^8 \;\le\; \sum_t \min\bigl(\operatorname{rank}\ \mathrm{flat}_{S_{\mathrm{row}}} T_t,\ \operatorname{rank}\ \mathrm{flat}_{S_{\mathrm{col}}} T_t\bigr),$$

[6] Module `lean/EthFollowup/Barriers/DecorrelatedRank.lean`.
[7] Module `lean/EthFollowup/Barriers/DecorrelatedGrid32.lean`.
[8] Module `lean/EthFollowup/Barriers/DecorrelatedGrid33.lean`.

with S_{row} the legs in row c and S_{col} the legs in column 3, and some term has this minimum at least $|\alpha|^8/\#\text{terms}$. The side conditions (disjointness, matching, $|K_X| = 12$, four straddlers per cut) are proved by `decide`.

Proof sketch. Keep the 12 legs at the block $\{a, b\} \times \{1, 2\}$ on edges leaving the block as rows and their partners as columns, and fix the other legs. The contraction of G is a permutation matrix of rank $|\alpha|^{12}$. Exactly four kept legs lie on the wrong side of the row cut and four of the column cut, so each term's contraction has rank at most $|\alpha|^4$ times either flattening rank. Subadditivity gives $|\alpha|^{12} \le |\alpha|^4 \sum_t \min(\cdot, \cdot)$. □

With "a row network has rank $\mathrm{flat}_{S_{\mathrm{row}}} \le d_{ac} d_{bc} \le 2^{\mathrm{cost}}$" (on paper), Theorem 4.11 is $S \ge N^8$ on the 3×3 level, i.e. $\rho \ge 4/9$ there. The same lossy contraction on larger levels gives the following.

Proposition 4.12. (on paper; `engine2-decorrelated.md` §7.3 and Theorem 5) *Every decomposition of the $g \times g$ level, $g = 2h$, into row and column networks has $\sum_t 2^{\mathrm{cost}_t} \ge N^{3h^4 - 2h^3}$ out of $Q = h^2(2h-1)^2$, hence $\rho \ge h(3h-2)/(2(2h-1)^2) - o(1) \to 3/8$ and $\alpha' \le \log_2(8/3) \approx 1.415$.*

Proposition 4.13. (on paper; `engine2-decorrelated.md` Proposition 7.6, §8.2, §8.3) *No two-party lossy contraction certifies more than $(3/4 + O(1/g))Q$ on a full level, so 3/8 is the limit of the method. On the 3×3 level no* flattening certificate*, that is no combination of flattening inequalities, certifies more than 8 (a profile of tailored "hub" terms passes every flattening test), and no m-party Christandl–Vrana–Zuiddam functional [19], tensor rank, slice rank or geometric rank with straddling legs certifies more than 8. That the optimum of the flattening certificates is exactly $D_3 = 8$ is computed, by an exact integer program (Theorem 4.14).*

Remark 4.14. (checked numerically) Exact integer programs give the flattening optimum $D_g = 8, 32, 84, 189$ for $g = 3, 4, 5, 6$, equal to the single balanced contraction in each case. At $\lambda = 1$, GF(2) SAT search finds rank sums 7 for two rectangles (bound 4, product 10) and none below 8 on the 3×2 grid. Data at $\lambda \le 2$ cannot separate $\rho = 1/2$ from anything, since $\log |\mathcal{C}|$ is not $o(\lambda)$ there.

Conjecture 4.15. (network rank of the 3×3 level; `engine2-decorrelated.md` Conjecture 8.1) *For every field and alphabet of size N, every family of tensors T_t with $\sum_t T_t = G$ in which each T_t is a row network or a column network of cost c_t satisfies $\sum_t 2^{c_t} \ge N^9$. The weakest new statement would be $S \ge N^{8+\varepsilon}$, or a decomposition with maximum cost at most $(9 - \varepsilon)\lambda$. Its $g \times g$ form $S \ge N^Q$ is $\rho \ge 1/2$ for all correlated decompositions.*

A proof cannot be a rank bound on a node-locally processed flattening (Theorem 4.13); it must use that a row network's three cuts come from one triple of bonds.

Relation to Majdoddin and to Dias–Lima. Majdoddin [43] defines a *rewrite-and-merge* class: per-block exact identities on configurations of edges, the terms being all products of one piece per (configuration, block) pair (his (L2)), and merges under which every configuration meets at most two groups (his (L3)). He proves that every grid algorithm of the class has exponent at least $1/\log_2 K$ per edge bit (his Theorem 4.13 and Corollary 4.19), and at least $(\log_2 K)^{-3}$ for two balanced homogeneous merges (Theorem 4.12). For product terms this contains Theorem 4.6 with the exact constant and without assuming $K_{\ell+1} \ge K_\ell^2$, and Theorem 4.1 is the rectangle case of his two-merge floor (Proposition 4.8), itself the observation of [A]'s Appendix H. He notes that non-product groupings of terms lie outside his class (Remark 4.5), calls configurations that meet three or more groups of a merge "the open door", and writes that "for the full model of Signed Network Sums no barrier is known" (§4.8). The correlated decompositions of this section are such non-product groupings: an arbitrary family $\sum_t T_t = G$ in which each term is charged its cheapest pairwise network in the row or the column grouping. (A cost model that charged a hyperedge less than its cheapest pairwise simulation would escape ours.) **??** 4.9–4.12 are, to our knowledge, the first bounds there; they are weaker (N^8 of 9 on the 3×3 level, $\rho \ge 3/8$ on full levels) and restricted to the two grid merges, and Theorem 4.15 is the grid case of his open door (his Open Problem 5). The step that makes them possible is the diagonal contraction, which bounds $\min(\mathrm{rank}^{\mathrm{row}}, \mathrm{rank}^{\mathrm{col}})$ of each term and so needs no mixed term. His Example 4.10 (flattening inequalities do not show that three-cell configurations do not trade) and Theorem 4.13 (flattening certificates stop at 8) are parallel observations in the two models. For [O]'s interpolation algorithm, Dias and Lima [29] prove treewidth barriers for every clause-product scheme; they state that the clause-product form of [O]'s algorithm is an instance of their Definition 5.1, so the M_0-template of Section 5 is a special case, and we compare the bounds there. Neither paper treats a counting problem other than (Max-)SAT, and neither has a Lean development of its barriers.

4.5. **What the caps mean for the engine.** With $\rho \geq 1/2$ the ceiling of [A]'s framework is $2^{O(U/\log U)}$ (`rate.md` §0, on paper), and with $U = \Theta(n \log n)$ this is $2^{\Theta(n)}$. For the permanent the bound is sharper: assignment-class networks have engine exponent at least $(3x_0 + 6)n/2 \geq 30n$, with [A]'s published $x_0 \geq 18$, for every $\rho \in [1/2, 1)$ (`wild-ideas.md` F3, on paper). Table 1 collects the mechanisms.

mechanism	cap	status
pieces, any family ([A], Def. 7)	$\rho \geq 1/2 - o(1)$	Lean (rank form); [A], App. H; [43]
graded two-order pieces	$\rho \geq 1/\sqrt{2} - o(1)$	Lean (counts), paper
two groupings, product terms	$\alpha' \leq 1$	paper; exact in [43, Thm. 4.13]
any family of groupings, adaptive or random	$\alpha' \leq 1$	paper
gadget recursions, m groupings	$\alpha' \leq 1 + O(\delta/\ln d)$	paper
correlated, two rectangles / 3×2 / 3×3	$S \geq N^2, N^3, N^8$	Lean (rank form)
correlated, full levels $g = 2h$	$\rho \geq 3/8$, $\alpha' \leq 1.415$	paper
tensor rank, transforms, characteristic p, trees	$e_K \geq$ const or unchanged	paper

TABLE 1. Caps on the rate. "Rank form" means that Lean bounds flattening ranks and the step to costs is on paper. The two correlated rows are the only ones whose terms need not be products.

5. OpenAI's M_0-template is tight at $\sqrt{Q}$

In this section we bound every instance of [O]'s M_0-template, whatever the matrix M_0 (commuting with the swaps or not), the label set and the number Q of points. The template needs M_0 with (T1) every row of $I - M_0$ in the row space of the measurement matrix, so that #measurements $\geq \operatorname{rank}(I - M_0)$, and the support condition (T2). The core is a triangular submatrix of $I - M_0$.

Theorem 5.1. (`EthFollowup.Barriers.Triangular.triangular_rank_bound_hamming`, `EthFollowup.Barriers.Triangular.independent_sets_rank_bound`; machine-checked) *Let K be a field, B a finite set of coordinates and $a, d : B \to [r]^Q$.*[9] *Let $\varphi : [r]^Q \to \mathbb{R}$ satisfy $|\varphi(x) - \varphi(y)| \leq d_H(x, y)$, let $R \in \mathbb{N}$, and let $M \in K^{B \times B}$ satisfy (T2) in the form $M[u, w] \neq 0 \Rightarrow d_H(d_u, a_w) + 1 \leq R$. Then*

$$\#\{u \in B : \ \varphi(d_u) - \varphi(a_u) \geq R\} \ \leq \ \operatorname{rank}(I - M).$$

For any distance on the labels and sets A, B' at distance at least R, if $M[u, w] \neq 0 \Rightarrow \operatorname{dist}(d_u, a_w) < R$, then $\#\{u : d_u \in A, \ a_u \in B'\} \leq \operatorname{rank}(I - M)$.

Proof. Order the set U on the left by non-increasing $\varphi(a_u)$. For u before or equal to w in U, $\varphi(d_u) - \varphi(a_w) \geq \varphi(d_u) - \varphi(a_u) \geq R$, so $d_H(d_u, a_w) \geq R$ and $M[u, w] = 0$ by (T2). Hence $(I - M)[U, U]$ is triangular with unit diagonal. □

Proposition 5.2. (on paper; `parity-barrier.md` Corollary 1.1, Theorem 2, §5) *On [O]'s full label space $([r] \times [r])^Q$ with every label used equally often, the measurement fraction satisfies $\varepsilon \geq \Pr[\mathrm{Bin}(2Q, \frac{1}{2}) \geq Q + R]$, so $R \geq (1 - o(1))\sqrt{Q \ln(1/\varepsilon)}$ when $\ln(1/\varepsilon) \gg \ln Q$; this matches [O]'s window $R \approx \sqrt{2Q \ln(1/\varepsilon)}$ up to a factor at most $\sqrt{2}$. Modulo the Arora–Rao–Vazirani structure theorem [1], for $r \geq 3$, any schedules obeying the group bound, any coordinate layout and any M satisfying (T2), on (c', κ)-linked bounded-occurrence formulas, $R \leq bQ/(8\sqrt{\log n})$ implies $\operatorname{rank}(I - M) \geq \kappa n/2$. Hence every instance of the template costs $2^{\Omega(n \log \log n/\sqrt{\log n})}$ on such formulas.*

So within Engine I the main-text template is tight up to a factor $\log \log n$, and neither a non-commuting M_0 nor structured or sparse labels help. This is a statement about [O]'s construction as published. For the clause-product form of the interpolation algorithm in general, Dias and Lima [29] prove $2^{\Omega(\sqrt{\mathrm{tw}})}$ at treewidth tw, and $2^{\Omega(\sqrt{n})}$ with high probability on random 3-CNF, in a cost model that charges what the preprint pays (their Theorems 8.1 and 8.3). They state that the clause-product form of [O]'s algorithm is an instance of

[9] Module `lean/EthFollowup/Barriers/TriangularRank.lean`.

their scheme (their §5), so the template is a special case, and Theorem 5.2 is a stronger bound for it, modulo the Arora–Rao–Vazirani theorem, on linked bounded-occurrence formulas. We have not compared linked formulas with their random inputs, and we do not claim to close the gap that they leave open on expanding inputs (their §9). Whether the general interface `def:layout`, in which the measurement part may vary with the point in a non-polynomial way, costs $2^{\Omega(n \log\log n/\sqrt{\log n})}$ as the template does is open: it is strictly weaker as a constraint ("runs instead of windows" reach $\varepsilon = 0.1$ at $R = 9$ for all $Q \le 4096$, where every M_0 needs $R \ge 59$ at $Q = 4096$), but in the regime the engine needs the gain is a constant factor (`parity-barrier.md` §5 and §7).

Remark 5.3. (checked numerically) At $(Q, R) = (3, 2)$ a non-commuting M_0 beats every commuting one, with $\operatorname{rank}(I - M_0) = 8$ against 10 (floating-point search), and the general interface was checked exactly for $Q \le 6$. Neither affects the asymptotic bound of Theorem 5.2.

6. Three linear encodings of Hamiltonicity have size floors

In this section we treat three encodings that we built for Hamiltonian cycles and that would have extended the sparse results to wider classes. These are barriers for *our* linear encodings (`hamiltonicity.md` §8, `linear-encoding-2.md`, `derandomization.md`), not for the engines. All three are machine-checked without paper hypotheses.

Colouring filters. Cut&Count [24] extracts connectivity from a sum over consistent colourings with local weights; such a filter gives the parity of #HC, and the question is whether it can be exact.

Theorem 6.1. (`EthFollowup.Barriers.ColouringGraph.colouring_barrier_graph`; machine-checked) *Let K be a field of characteristic 0, $n, q \in \mathbb{N}$ with $q > 0$, $z : [q] \to K$ and r a vertex of $[n]$.*[10] *For a 2-factor X let its value be $z_0^{|C_r|} \prod_{C \neq C_r} \sum_j z_j^{|C|}$ over the components C of X, C_r the root's. If for every graph G on $[n]$ there is $\mu \neq 0$ with $\sum_{X \in 2F(G)} \operatorname{value}(X) = \mu \cdot \#\mathrm{HC}(G)$ (`ExactFilter` with root colour 0; μ may depend on G), then $n \le q + 4$.*

Proof sketch. The n-cycle has one 2-factor, which is Hamiltonian, so $z_0 \neq 0$. For $3 \le k \le n - 3$ the graph $C_{n-k} \sqcup C_k$, with the root in C_{n-k}, has one 2-factor, which is disconnected, so its value $z_0^{n-k} \sum_j z_j^k$ must be 0. This is `hvan`, and Theorem 6.2 applies: grouping the z_j by value gives an exponential polynomial with at most q bases and positive multiplicities that vanishes at $n - 5$ consecutive integers, which the Vandermonde matrix forbids if $n - 5 \ge q$. □

Theorem 6.2. (`EthFollowup.Barriers.Colouring.colouring_barrier`, `EthFollowup.Barriers.Colouring.colouring_threshold_upper`; machine-checked) *Let K be a field of characteristic 0 and $z : [q] \to K$ with $z_{j_0} \neq 0$ for some j_0 (module `ColouringBarrier`). If (hypothesis `hvan`) $\sum_j z_j^k = 0$ for every k with $3 \le k \le n - 3$, then $n \le q + 4$. Conversely, for a primitive $(n-2)$-th root of unity ζ, $\sum_{j<n-2} (\zeta^j)^k = 0$ for $1 \le k \le n - 3$. The hypothesis `hvan` is derived from the graphs $C_{n-k} \sqcup C_k$ inside the proof of Theorem 6.1.*

So exact colouring filters need about $\log_2 n$ bits per vertex, and the threshold lies in $[n - 4, n - 2]$; the modulo-2 filter escapes because positive multiplicities vanish only in small characteristic. Vertex-dependent weights are not covered.

Turn labellings. A *turn* is a triple $(v; u, w)$ of a vertex and two distinct neighbours. A labelling F of the turns of $G - r$ by $\mathrm{GF}(2)^k$ is *good* if every directed simple cycle of $G - r$ has nonzero label sum; $k^*(G)$ is the least such k. A good labelling gives a signed CNF for $n \cdot \#\mathrm{HC}(G)$ with $O(km)$ variables on bounded-degree graphs (`linear-encoding-2.md` Theorem G, on paper), so $k^* = O(1)$ would have been a route to #HC on all bounded-degree graphs.

Theorem 6.3. (`Kstar.theoremL`, `Kstar.theoremL_log`; machine-checked) *Let H_t be the crossing grid on $2t^2$ vertices (row paths, column paths and rungs; maximum degree 3), given as the Boolean adjacency `gridAdj t`.*[11] *If F assigns a vector in $\mathrm{GF}(2)^k$ to every triple (v, u, w) of vertices and every list of vertices that is a directed simple cycle of H_t (`isDirCycle`) has nonzero label sum, then $t < 2^{20k+10}$; for $t \ge 1$, $\lfloor \log_2 t \rfloor < 20k + 10$. So $k^*(H_t) \ge (\log_2 t - 10)/20$.*

[10] Module `lean/EthFollowup/Barriers/ColouringBarrierGraph.lean`, which imports the 2-factor definitions of `EthFollowup.HamiltonianParity`.

[11] Modules `lean/EthFollowup/Barriers/KstarBasic.lean` and `KstarLower.lean`.

The proof lifts a balanced 24-cycle of $H_{4,4}$ to every choice of four rows and four columns and replaces the Kővári–Sós–Turán theorem [36] of the paper proof by an iterated pigeonhole. The note's constant via [36] is $1/16$, and every bounded-degree expander contains H_t with $t^2 = \Omega(n/\log n)$ as a minor [20] (`kstar.md` §3, on paper). So the gap gadget cannot be made smaller than $\Theta(m \log m)$ on expanders by turn labels.

Residues. Deterministic decision from residues of #HC modulo small q needs a query whose residue is nonzero. For a finite family S of edge sets (the Hamiltonian cycles) put $c_T = \#\{H \in S : T \subseteq H\}$ and $t_q(S) = \min\{|T| : c_T \not\equiv 0 \bmod q\}$.

Theorem 6.4. (`EthFollowup.Barriers.Derand.tp_le_log_of_nonempty`, `EthFollowup.Barriers.Derand.tq_deleted_eq_contains`, `EthFollowup.Barriers.Derand.support_query_eq_zero`; machine-checked) *Let E be a type with decidable equality and S a finite family of finite subsets of E.*[12] *Then $\min\{|D| : \#\{H \in S : H \cap D = \emptyset\} \not\equiv 0 \bmod q\} = t_q(S)$. If $c_T \equiv 0 \bmod q$ for all $|T| < t$, then for every S_0 with $|S_0| < t$ and every f, $\sum_{H \in S} f(H \cap S_0) \equiv 0 \bmod q$. For every prime p and nonempty S, $t_p(S) \le \lfloor \log_2 |S| \rfloor$.*

Proof sketch. The duality is inclusion–exclusion and Möbius inversion between deletion counts and containment counts. For the support statement, $\#\{H : H \cap S_0 = A\} = \sum_{A \subseteq T \subseteq S_0} (-1)^{|T \setminus A|} c_T \equiv 0$. For the bound, fix an inclusion-maximal $H_0 \in S$ and count the sets $H_0 \setminus H$ modulo p: the resulting function is nonzero and orthogonal to all monomials of degree below t_p, so it has at least 2^{t_p} nonzero values, and $|S| \ge 2^{t_p}$. □

So every restriction, weighted or CNF-constrained query of support below t_q returns 0, and restriction search is deterministic $2^{o(n)}$ exactly when $t_Q = o(n)$; the bound $t_p \le \log_2 \#\mathrm{HC}$ says that the obstruction is the only one. A black-box algorithm that sees $\mathrm{HC}(G)$ only through weighted counts modulo $q \le Q$ needs at least $(2^N/\mathrm{lcm}(1..Q) - 1)/\ln Q$ queries (`derandomization.md` Theorem 2, via the Davenport constant, on paper).

7. Nine problems out of reach, and why

In this section we map nine natural targets after sparse #HC. Table 2 gives, for each, the best known bound, the reason the engines' guarantees do not reach it, and what is proved. "Out of reach" always means: no network or recursion of the analysed kinds, not hardness. We then give the two problems with Lean content a paragraph each.

The iterated product. Exact $\pi(x)$, and a deterministic prime finder by this route, need to isolate $\omega(m) = 1$, and the only identities we know do so through a k-fold product $m = d_1 \cdots d_k$ with $k = \Theta(n/\log n)$ (`prime-finding.md` §4, on paper). In the parity note [63] we proved Theorem 7.1 and the first statement of Theorem 7.2 (`rank_prodMat_pow`); we restate them here. The second statement of Theorem 7.2 (`rank_prodMatFixed_le`) is new here.

Theorem 7.1. (`EthFollowup.IterProd.BTree.cost_ge_entropy`, `EthFollowup.IterProd.BTree.cost_ge_equal`; machine-checked) *For every binary tree T with positive integer leaf weights ℓ_i and $L = \sum_i \ell_i$, where $\mathrm{cost}(T) = \sum_{\text{internal } v} \mathrm{wt}(v)$,*[13] *we have $L \log_2 L \le \mathrm{cost}(T) + \sum_i \ell_i \log_2 \ell_i$. If all k leaves have weight $w > 0$, then $kw \log_2 k \le \mathrm{cost}(T)$.*

Theorem 7.2. (`EthFollowup.IterProd.rank_prodMatFixed_le`, `EthFollowup.IterProd.rank_prodMat_pow`; machine-checked) *Over $\mathbb{Q}$, the matrix with rows $u \in [1, 2^a]$, columns $(v, m) \in [1, 2^b] \times [1, 2^{a+b}]$ and entries $[uv = m]$ has rank 2^a. With $m \ne 0$ fixed, the matrix $[uv = m]$ over any finite sets of rows and columns has rank at most $\tau(m)$.*

So in a tree verifier the cut above a node v carries at least $\mathrm{wt}(v)$ bits (?? 3.1?? 7.2, the link on paper), $U \ge \mathrm{cost}(T) \ge n \log_2 k$, and no tree shape helps. A bound for all verifiers is a different matter.

Proposition 7.3. (on paper; `iterated-product.md` §5) [A]*'s engine simulates every Boolean circuit of size at most Kn by a parsimonious label system with $U = O(\mathit{size})$. Hence a lower bound $U = \omega(n)$ for all parsimonious verifiers of $m = \prod_i d_i$ would be a superlinear lower bound on the circuit size of an explicit* TC^0 *predicate, for which no superlinear bound is known.*

Proposition 7.4. (on paper, oracle model; `prime-finding.md` §3(c)) *An adversary consistent with the prime gap bound of* [3] *and the other known density theorems can make every count $\pi(I) \bmod M$ queried by a deterministic strategy vanish as long as consecutive endpoints are far apart; against it, at least $x^{0.525}/(M \log M)$ queries are needed.*

[12] Module `lean/EthFollowup/Barriers/Derand.lean`. Minima are infima in $\mathbb{N}$.

[13] Modules `lean/EthFollowup/Barriers/IterProdTree.lean` and `IterProdCutRank.lean`.

problem	best known	why out of reach	proved
dense #HC, TSP	$2^n n^2$ [5, 32]; $2^n n^2/2^{\Omega(\sqrt{\log n})}$ [54]; 1.657^n decision, randomised [8]; directed #HC $2^{n-\Omega(\sqrt{n})}$ [39]	permutation witness; vertex-local networks have $U=\Theta(m)$; even $U=\Theta(n\log n)$ is fatal at $\rho\geq 1/2$	Lean: **??** 3.1**??** 3.4; paper: **??** 2.4**??** 3.5
permanent, #PM, hafnian	$2^n n$ [51]; $2^{n-\Omega(\sqrt{n})}$ [39]; $(2-10^{-40})^n$ over any commutative ring [38]	$n\log n$ bits in row-local networks; algebras of dimension $\binom{n}{n/2}$; #SAT encodings need $n\log n$ variables; char. p savings die at $p\approx n$	Lean: **??** 3.2**??** 3.3**??** 3.7**??** 3.8; paper: **??** 3.9**??** 3.13, Theorem 3.6
chromatic number	2^n [13]; k-colouring $(2-\varepsilon_k)^n$ randomised [68, 48]; 1.99982^n under the asymptotic rank conjecture [10]	growing k needs $n\lceil\log k\rceil$ variables; dense encodings need $P=\Theta(n\log k)$; inclusion–exclusion over 2^n sets is not a network	none; heuristic rank count (`ambitious-targets.md` §2.4)
Subset Sum, Partition	$2^{n/2}$ [33]; space $2^{n/4}$ [52], improved [44]	each party's partial sum carries b bits	paper: Theorem 7.5
dense Max-Cut, Max-2-SAT	$2^{\omega n/3}$ [67]	$\Theta(n^2)$ edge terms; the pair interaction has rank $2^{\operatorname{rank} E}$, so $P=\Theta(n)$ on dense graphs	none; one-line rank count
directed HC, dense	2^n; $2^{n-\Omega(\sqrt{n})}$ counting [39]	as dense #HC; sparse digraphs are done (`theoremDHC`)	as dense #HC
prime finding	$x^{1/2+o(1)}$ [37]; parity on short intervals $x^{1/2-c}$ [55]	residues of cheap counts can all vanish; exact isolation needs k-fold products, $k=\Theta(n/\log n)$	Lean: **??** 7.1**??** 7.2; paper: **??** 7.3**??** 7.4
exact $\pi(x)$	$x^{1/2+o(1)}$ [37]; $x^{2/3}/\log^2 x$ [27]	product-tree verifiers have $U=\Theta(n\log n)$, giving $2^{\Theta(n\sqrt{\log n})}$	as prime finding; Theorem 8.3
Tutte off $\mathcal{H}\cup\mathcal{L}$	2^n [12]	boundary states are all set partitions, rank Bell(b); interpolation needs $\Theta(n)$ hyperbolas	paper: Theorem 3.10; Theorem 3.12

TABLE 2. Nine problems out of reach of the engines' guarantees. "Lean" cites machine-checked steps; the step from a network to a matrix is on paper throughout. Best known bounds are deterministic unless marked.

Subset Sum. The natural network gives each of g clusters its items privately and shares partial sums of $b+\log n$ bits along a tree.

Proposition 7.5. (on paper; `ambitious-targets.md` §2.5) *For n items with b-bit weights, a clustered partial-sum network has engine exponent about $(4e_K+6/K)\,g\,b\geq 18K^7 b$ in case (i) of Lemma 8. Beating $2^{n/2}$ needs $b<n/(36K^7)$, where the $O(n2^b)$ dynamic programme is already faster; case (ii) and* [O]*'s model fail similarly. So no clustered-partial-sum network in either engine beats $2^{n/2}$ at $b=\Theta(n)$.*

8. What would have to change, and why proving that it will not is out of reach

In this section we list the five changes that would move a problem of Table 2 into reach, say why an unconditional proof that none of them exists is beyond current techniques, and state the open questions precisely.

Five changes.

(i) A decomposition of the 3×3 level with $\sum_t 2^{c_t}<N^9$ (Theorem 4.15 false), or more generally a correlated decomposition with $\rho<1/2$ on full levels. This is the grid-level case of the "open door" of [43, §4.8 and Open Problem 5].

(ii) Pieces outside the graded two-order family with $\rho<1/\sqrt{2}$: a statement of the Hurwitz–Hrubeš–Wigderson–Yehudayoff type [34], since $\rho\to 1/2$ would give sum-of-squares formulas of nearly linear size.

(iii) An algorithm for (K,λ)-cliques that is not a sum of $2^{o(N)}$ clique-type sub-instances, for example one that eliminates variables or adapts to the tables.

(iv) For prime counting, a parsimonious verifier of a k-fold product with $o(n\sqrt{\log n})$ shared bits (Theorem 8.3).

(v) For the Tutte polynomial, a sum of networks over many different clusterings, or skeletons that depend on private edges.

The meta-barrier. For (iii) the obstacle to an unconditional negative answer is the following. We call an algorithm for the (K,λ)-clique *oblivious algebraic* if it is a straight-line program over $\mathbb{Q}$ in the table entries, that is, an arithmetic circuit whose sequence of operations is fixed in advance and does not depend on the entries; its size is its number of operations. Let $D=2^{\lambda}$ and $P_K(T)=\sum_X \prod_{a\in[K]} T_a(X|_{\delta(a)})$, the value of the (K,λ)-clique as a polynomial in the table entries: it is set-multilinear of degree K in $N_{\text{in}}=K\cdot D^{K-1}$ variables with 0/1 coefficients, and its cost $2^{e_K K^2\lambda}$ equals $(N_{\text{in}}/K)^{e_K K^2/(K-1)}$.

Proposition 8.1. (on paper; `engine2-algebraic.md` M6.1) *Take $\lambda=K$, so that $K\approx\sqrt{\log N_{\text{in}}}$, and index the family by N_{in}: for N between consecutive values of N_{in}, which differ by a factor $N_{\text{in}}^{o(1)}$, take P_K for the largest K with $N_{\text{in}}\le N$, padded with unused variables. Work over $\mathbb{Q}$.*

(i) The family lies in VNP *by Valiant's criterion* [64] *([17, Proposition 2.20]): P_K has degree $K\approx\sqrt{\log N_{\text{in}}}$ and N_{in} variables, both polynomial in N_{in}, and its coefficient function is* 0/1 *and computable in polynomial time (it checks that the K indices of a monomial agree on every edge).*

(ii) If, for all sufficiently large K, every oblivious algebraic algorithm that computes P_K has size at least $2^{e_K K^2\lambda}$ with $e_K\ge\varphi(K)/K$ for a function $\varphi\to\infty$, then the size is at least $(N_{\text{in}}/K)^{\varphi(K)}$, superpolynomial in N_{in}, so the family is not in VP *and* VP $\neq$ VNP*. With $\varphi=cK/\log K$ this covers "no oblivious algebraic algorithm beats $c/\log K$".*

(iii) The value is computable in time $N_{\text{in}}^{K/2+o(1)}=2^{O(\log^{3/2} N_{\text{in}})}$, so the same lower bound for Boolean circuits would put a quasi-polynomial-time function outside P/poly*, that is* QP $\not\subseteq$ P/poly*, and hence* EXP $\not\subseteq$ P/poly*; against uniform algorithms it would be a superpolynomial time lower bound for an explicit problem in quasi-polynomial time, of a kind known today only from diagonalisation.*

This is not a theorem about any existing algorithm. It says that the query bound $(K-1)/K^2$ is the only unconditional lower bound on e_K available without a breakthrough in algebraic or Boolean circuit complexity, so barriers must be proved for restricted classes, as in Section 4. [A]'s recursion is an oblivious algebraic algorithm in this sense: each modified table is a polynomial in the tables of the nodes it absorbs, with coefficients independent of the entries; its products are over disjoint node sets; its sums are over index sets fixed in advance; and the groupings it searches for depend only on the skeleton, not on the entries. Majdoddin makes the fixed-K form of this observation: at $K=4$, a bound above N^3 "would be a superlinear lower bound for an explicit degree-4 polynomial, which is out of reach" [43, Remark 2.9].

Proposition 8.2. (on paper; `engine2-algebraic.md` M6.2) *Lemma 8(i) of* [A]*, with the accounting of the proof of its Theorem A(c) ($g=3K^8\le m$, $P=2k\lceil m/g\rceil$, $U<2km$), reduces #3-SAT on a bounded-occurrence formula with n variables and $m=O(n)$ clauses to one (K,L)-clique with $K^2L=O(n)$, at overhead $2^{O(n/K+K^8\log K)}$. With $K=\Theta(n^{1/9})$ the overhead is $2^{O(n^{8/9}\log n)}$ and $\log K=\Theta(\log n)$, so $e_K=o(1/\log K)$ would give bounded-occurrence #3-SAT in $2^{o(n/\log n)}$. So the hypothesis $H_{\log}$, "bounded-occurrence #3-SAT needs $2^{\Omega(n/\log n)}$", implies $e_K=\Omega(1/\log K)$ for all algorithms. $H_{\log}$ is consistent with everything known but is not a standard hypothesis.*

Any $K=n^c$ with $0<c\le 1/9$ works; the constraint $g=3K^8\le m$ is what rules out larger K in case (i) of Lemma 8.

For (iv). Theorem 7.3 plays the role of the meta-barrier: an $\omega(n)$ bound for all verifiers would be a superlinear circuit lower bound for an explicit TC^0 function.

Open questions. We state them as precisely as we can. Theorem 4.15 (the 3×3 level) and Theorem 3.6 (Model V) are above.

Question 8.3. (g-party verification of a product; `iterated-product.md` §6) *The witness $(d_1,\dots,d_k,m)$ with $\sum_i \mathrm{bits}(d_i)=\mathrm{bits}(m)=n$ and $k\approx n/\log n$ is distributed, as the designer likes, among $g\approx n^{1/5}$ parties*

holding at most $n^{4/5}$ bits each. Auxiliary labels are allowed, every label is seen by exactly two parties, each party accepts or rejects its own labels, and exactly one accepting labelling exists per witness with $\prod_i d_i = m$ and none otherwise. What is the least total U of label bits? Known: $\Omega(n) \le U \le (1/5 + o(1))\, n \log_2 n$; tree verifiers have $U \ge n \log_2 k$ (Theorem 7.1). $U = o(n\sqrt{\log n})$ would give exact $\pi(x)$ and deterministic prime finding in $x^{o(1)}$.

Question 8.4. (the pieces constant; `rho-lower-bound.md` §7) *Let ρ^* be the infimum of the ρ for which pieces with parameters (ρ, c) exist for some c. Is $\rho^* < 1/\sqrt{2}$? Proved: $\rho^* \ge 1/2 - o(1)$ (Theorem 4.1); $\rho \ge 1/\sqrt{2} - o(1)$ for graded two-order pieces with top-pole pairing on spaces of square-root growth, in particular all C_2-cofinite vertex operator algebras (Theorem 4.4). Open: the zero-mode loophole (pairing in a lower pole), two-order structures of growth below $\exp(\sqrt{E})$, and true ranks below the certified counts.*

Conjecture 8.5. (Catalan rank on $y = 1$; `tutte-plane.md` Lemma 3.2(c)) *For every b and every $u \ne 0$, the acyclic gluing matrix $A_u[P, Q] = u^{|P \vee Q|}[|P| + |Q| - |P \vee Q| = b]$ on the partitions of $[b]$ has rank* $\mathrm{Cat}(b)$. *Proved: rank at most 2^{2b-1}. Checked: $b \le 6$, and at least* $\mathrm{Cat}(7)$ *at $b = 7$ (Theorem 3.12).*

Question 8.6. (the two Tutte loopholes; `tutte-plane.md` §5) *Does Theorem 3.10 hold for sums of $2^{o(n \log n)}$ vertex-local networks with* different *clusterings, and for skeletons that depend on private edges?*

9. The Lean development

In this section we list the modules, the cited theorems and the trusted base. The development is the directory `lean/EthFollowup/Barriers` of the public repository [60], written in Lean 4 [26] on Mathlib [56] only, toolchain v4.34.1, Mathlib commit `d13f23b`. It does not use [A]'s or [O]'s developments. In this paper the macro for Lean names is reserved for theorems of this directory; names from other developments (`groupingExplicit`, `theoremDHC`, ...) are set in plain typewriter type.

module	content	lines
`RankBounds`	cut lemma, subadditivity, Model R and tree summations, pieces	373
`RowCut`	Laplace expansion, row-cut rank $\binom{n}{k}$	312
`AlgebraRank`	algebra dimension $\binom{n}{n/2}$	168
`PermModP`	per $A = 0$ for low rank in characteristic p	135
`SpreadLemma`	spread inequality for graded two-order pieces	123
`DecorrelatedRank`	two rectangles, N^2	126
`DecorrelatedGrid32`	3×2 grid, N^3; row-local no-go	260
`DecorrelatedGrid33`	lossy flattening lemma; 3×3 level, N^8	286
`TriangularRank`	triangular rank bound for the M_0-template	121
`ColouringBarrier`	colouring barrier, polynomial form, converse	114
`ColouringBarrierGraph`	colouring barrier for graphs	566
`KstarBasic`, `KstarLower`	turn labellings, Theorem L	53, 768
`Derand`	duality, support lemma, $t_p \le \log_2 \lvert S \rvert$	573
`IterProdTree`, `IterProdCutRank`	product trees, rank of the product indicator	142, 148
total (16 modules)		4268

Table 3. The modules of `lean/EthFollowup/Barriers`, with line counts (`wc -l`).

Cited theorems. We cite 45 theorems, all in this directory: from `RankBounds` twelve (?? 3.1?? 3.3?? 3.4?? 4.1?? 4.2), from `RowCut` three (Theorem 3.2), from `AlgebraRank` two (Theorem 3.7), from `PermModP` one (Theorem 3.8), from `SpreadLemma` three (Theorem 4.3), from `DecorrelatedRank` two (Theorem 4.9), from `DecorrelatedGrid32` four (Theorem 4.10), from `DecorrelatedGrid33` four (Theorem 4.11), from `TriangularRank` two (Theorem 5.1), from the two colouring modules three (?? 6.1?? 6.2), from `KstarLower` two (Theorem 6.3), from `Derand` three (Theorem 6.4), and from the two `IterProd` modules four (?? 7.1?? 7.2). The repository README counts 38 theorems for an earlier state; the number here is what this paper cites.

Ten statements verbatim. The following are verbatim from the source, up to line breaks; we render Unicode symbols as mathematics and omit the proofs. Section variables are recalled in comments.

```
-- RowCut: variable {K : Type*} [Field K] {ι : Type*} [Fintype ι] [DecidableEq ι]
--   (I : Finset ι)
theorem rowCut_rank : (rowCutMatrix (K := K) I).rank = (Fintype.card ι).choose I.card
-- RankBounds
theorem modelR_bound_nlogn {J : Type*} (cls : Finset J) (n s U : ℕ) (k w : J → ℕ)
    (hk1 : ∀ j ∈ cls, 1 ≤ k j) (hks : ∀ j ∈ cls, k j ≤ s)
    (hsum : ∑ j ∈ cls, k j = n)
    (hrank : ∀ j ∈ cls, n.choose (k j) ≤ 2 ^ w j) (hU : ∑ j ∈ cls, w j ≤ U) :
    (n : ℝ) * Real.logb 2 ((n : ℝ) / s) ≤ ∑ j ∈ cls, Real.logb 2 (n.choose (k j)) ∧
      ∑ j ∈ cls, Real.logb 2 (n.choose (k j)) ≤ U
-- AlgebraRank: variable {K : Type*} [Field K] {R : Type*} [Ring R] [Algebra K R]
theorem algebra_rank_half {n : ℕ} [FiniteDimensional K R] (φ : R →ₗ[K] K)
    (r : Fin n → Fin n → R)
    (h : ∀ f, φ (word r f) = if Function.Bijective f then 1 else 0) :
    n.choose (n / 2) ≤ Module.finrank K R
-- SpreadLemma: variable ... (D : ℤ → ℕ) ...
theorem spread_slice_zero (hn : ℕ) :
    2 * Real.logb 2 (D (2 * hn)) ≤
      (Nat.clog 2 (Rrow D hn (range (2 * hn + 1)) (fun i => 0 - hn + i)) : ℝ) +
        Nat.clog 2 (Rcol D hn (range (2 * hn + 1)) (fun i => 0 - hn + i))
-- DecorrelatedRank (namespace Decorrelated):
--   variable {K : Type*} [Field K] {α : Type*} [Fintype α] [DecidableEq α]
theorem rankSum_ge [Nonempty α] {ι : Type*} (s : Finset ι) (T : ι → Tensor2 K α)
    (hsum : ∑ t ∈ s, T t = FF) :
    Fintype.card α ^ 2 ≤ ∑ t ∈ s, min (rowFlat (T t)).rank (colFlat (T t)).rank
-- DecorrelatedGrid33 (namespace Grid33): same variables; L = Fin 18 × Bool (edge, end)
theorem rankSum_ge [Nonempty α] {ι : Type*} (s : Finset ι) (T : ι → Tensor K L α)
    (hsum : ∑ t ∈ s, T t = G) :
    Fintype.card α ^ 8 ≤ ∑ t ∈ s, min (flat Srow (T t)).rank (flat Scol (T t)).rank
-- TriangularRank: variable {K : Type*} [Field K] {B : Type*} [Fintype B] [DecidableEq B]
theorem triangular_rank_bound_hamming {Q r : ℕ} (a d : B → (Fin Q → Fin r))
    (φ : (Fin Q → Fin r) → ℝ) (hφ : ∀ x y, |φ x - φ y| ≤ hammingDist x y)
    (R : ℕ) (M : Matrix B B K)
    (hM : ∀ u w, M u w ≠ 0 → hammingDist (d u) (a w) + 1 ≤ R) :
    (univ.filter fun u => (R : ℝ) ≤ φ (d u) - φ (a u)).card ≤ (1 - M).rank
-- KstarLower (namespace Kstar)
theorem theoremL {t k : ℕ}
    (F : Fin (2 * t * t) → Fin (2 * t * t) → Fin (2 * t * t) → (Fin k → ZMod 2))
    (hF : Good (gridAdj t) F) : t < 2 ^ (20 * k + 10)
-- IterProdTree (namespace EthFollowup.IterProd.BTree)
theorem cost_ge_equal (T : BTree) (w : ℕ) (hw : 0 < w) (hT : allLeaves w T) :
    (leaves T : ℝ) * w * logb 2 (leaves T) ≤ cost T
-- ColouringBarrierGraph: variable {n : ℕ} {K : Type*} [Field K]
theorem colouring_barrier_graph [CharZero K] {q : ℕ} (hq : 0 < q) (z : Fin q → K)
    (r : Fin n) (hex : ExactFilter z r ⟨0, hq⟩) : n ≤ q + 4
```

The definitions these use are short and stated in the files: `rowCutMatrix` and `joinMat` (the permanent of $(X;Y)$), `word` (the ordered product $r_{0f(0)}\cdots r_{n-1,f(n-1)}$), `Rrow`, `Rcol`, `FF`, `rowFlat`, `colFlat`, the leg list `edges`, `graphT`, `flat`, `Srow`, `Scol`, `Good`, `isDirCycle`, `gridAdj`, `BTree`, `cost`, `leaves`, `allLeaves`, `ExactFilter`, `compValue`, and the 2-factor and Hamiltonian-cycle sets of `EthFollowup.HamiltonianParity`.

Axioms. `#print axioms` reports `propext`, `Classical.choice` and `Quot.sound`, or a subset, for all 251 `theorem` declarations of the directory (audit of 5 November 2026). No module contains `sorry`, `native_decide` or a new axiom. All 16 modules have build artefacts; the axiom audit imports all of them in one file.

Hypotheses that stand for paper steps. In `modelR_bound_nlogn`, `hrank` is the cut lemma applied to the row cut of a given network and `hU` the pair count (whose counting step is `sum_crossing_le_total`); in `treeSkeleton_bound`, `hrank` is the cut lemma with the vertex-cut rank. The hypothesis `hvan` of `Colouring.colouring_barrier` is discharged in Lean by `colouring_barrier_graph`. The Lean statements on correlated decompositions bound flattening ranks; that a network of cost c has flattening rank at most 2^c is on paper.

What is not formalised. The network models of Theorem 2.2 and the step from a network to a factorised matrix; Model V beyond tree summation; the Tutte barrier; the lossy lemma for general configurations with

pairings and Theorem 4.12; the grouping caps; the template corollaries; the meta-barrier. Scratch files of the repository (`lean/EthFollowup/Scratch`) contain checks by `#eval` and `decide` on instances, for example the $\lambda = 1$ product decomposition of two rectangles with rank sum 10; they are not part of the audited development and no statement of this paper rests on them.

10. Related work

We organise related work by idea: rank methods for restricted algebraic models, tensor-network complexity, connection matrices and the Tutte plane, exact algorithms for the dense problems, the recent follow-ups to the two refutations, and the labs' own limits.

Rank methods for restricted algebraic models. Nisan [46] characterised the size of noncommutative algebraic branching programs by the ranks of partial-derivative matrices along the variable order, and proved exponential lower bounds for the permanent and determinant there. Hrubeš, Wigderson and Yehudayoff [34] reduced lower bounds for noncommutative circuits to the sum-of-squares problem; Raz [49] proved superpolynomial bounds for multilinear formulas; Limaye, Srinivasan and Tavenas [40] for constant-depth circuits; and Bhargava and Tengse [6] relate partial derivatives to commutative ROABPs. Closest to our Model R bound is Nisan's. In contrast, our cuts are cuts of a network, and locality of the node tables replaces the variable order.

Tensor-network complexity. Austrin, Kaski and Kubjas [2] studied the cost of evaluating multilinear maps by tensor networks. Christandl, Vrana and Zuiddam built the quantum functionals, monotone under restriction and additive under direct sums [19], and computed asymptotic ranks of graph tensors [18]; the complete-graph tensor is exactly the clique of [A]'s recursion. Majdoddin [43] uses these functionals to prove that configurations on at most three cells of two merges cannot trade cost (his Lemma 4.11), and shows that flattening inequalities alone cannot prove this (his Example 4.10), in a model where terms are products. In contrast, our lower bounds are for sums of networks whose terms need not be products, and the 3×3 level shows where flattening certificates, and these functionals with straddling legs, stop (Theorem 4.13).

Connection matrices and the Tutte plane. Lindström [41] and Wilf [66] factorised meet matrices on lattices; Freedman, Lovász and Schrijver [30] and Lovász [42] related connection ranks to graph parameters. Jaeger, Vertigan and Welsh [35] classified the easy points of the Tutte plane; Dell, Husfeldt, Marx, Taslaman and Wahlén [28] and Brand, Dell and Roth [16] proved #ETH lower bounds across it; Björklund, Husfeldt, Kaski and Koivisto [12] compute it in 2^n. Rank methods for treewidth dynamic programmes are due to Cygan, Kratsch and Nederlof [23], Curticapean, Lindzey and Nederlof [22], and, for counting connected spanning edge sets modulo p, Groenland, Mannens, Nederlof and Szilágyi [31]. In contrast, we bound the shared bits of networks, not the states of a decomposition, and the bound is unconditional within the model.

Exact algorithms for the dense problems. For TSP and #HC: Bellman and Held–Karp [5, 32], Björklund [8, 9], Björklund and Husfeldt for the parity [11], Stoian [54], and Li [39], who also treats the hafnian. For the permanent: Ryser [51], Curticapean's $(2-\varepsilon)^n$ algorithm with $\varepsilon = 10^{-40}$ over every commutative ring (arXiv, 2 November 2026; Binet–Minc with dynamic programming) [38], which answers Knuth's question and is independent of the engines, Valiant's #P-completeness [65], Barvinok for fixed rank [4], Björklund, Husfeldt and Lyckberg modulo prime powers [14], and Björklund and Williams over finite rings [15]. For colouring: Björklund, Husfeldt and Koivisto [13], Zamir [68] and Pratt [48], and the conditional bound of [10]. For Max-Cut, Williams [67]; for Subset Sum, Horowitz–Sahni [33], Schroeppel–Shamir [52] and Nederlof–Węgrzycki [44]; for $\pi(x)$, Lagarias–Odlyzko [37], Deléglise–Rivat [27] and Tao–Croot–Helfgott [55]. In contrast, we give no algorithm; we explain why the engines' guarantees stop short of these bounds.

Barriers for the SAT algorithms themselves. Closest to our work are two papers on the SAT algorithms. Majdoddin [43] computes explicit crossover points of [A]'s recursion (and compares them with our crossover note) and proves the class barrier discussed in Section 4.4: every grid algorithm of his rewrite-and-merge class, the recursion as written included, has exponent at least $1/\log_2 K$ per edge bit (his Theorem 4.13 and Corollary 4.19). Dias and Lima [29] extract the constants of [O]'s tensor-contraction proof and prove treewidth barriers for its clause-product scheme (their Theorems 8.1 and 8.3). Our product-term cap (Theorem 4.6) is an independent and weaker version of Majdoddin's, from a note dated 29 October, one day before his paper; neither paper cites our barrier notes. In contrast, we bound decompositions whose terms are not products and the counting problems beyond (Max-)SAT, and our rank statements are machine-checked.

The October–November 2026 follow-ups. Rudich and Rousseau [50] and Shunia [53] factor with the engines. De Brito [25] proves, after [A]'s Appendix H, a rank floor for families of the shape of [A]'s Theorem B, which closes the growth case $\beta < 1/2$ for that shape; Majdoddin [43, Proposition 4.24] proves $\rho \geq 2/3 - o(1)$ for

every concave profile in that shape. Our Corollary (the law $\rho \geq 2^{\beta-1}$, note of 28 October) is dated before both and left $\beta < 1/2$ open; outside the shape the question remains open (Theorem 8.4). Bhattacharjee [7] states (his Theorem 7.10) that the Tutte polynomial on H_q is computable for bounded-degree graphs and integer $q \geq 1$ by a white-box extension of [O]'s algorithm; we have not checked the proof. It was public (29 October) before our note on the Tutte plane (30 October), whose positive part we do not claim here. Nguyen [45] (2 November) counts binary CSP solutions on bounded-degree classes in $D^{C_\Delta k/\sqrt{\log k}}$, an independent instance of the same domain-uniform bound, with Lean on imported engine theorems. None of the three cites our notes. In contrast to these three, which are positive or concern one shape of pieces, we bound what the engines do not give.

The labs' own limits. [A] and [O] name what their theorems do not refute (Section 2): Orthogonal Vectors, CNF-SAT of unbounded width, and transfer under polynomial-time reductions. In contrast, our barriers are about problems that do reduce to bounded-width counting, but only with at least of order $n \log n$ variables or shared bits in the models we analyse.

11. Limitations

- *Every barrier is for a named model.* Model R(s), Model V(s) with a fixed skeleton, pieces with certified costs, min-over-groupings recursions with $2^{o(\lambda)}$ pieces, row and column network decompositions, [O]'s M_0-template, tree verifiers, clustered partial-sum networks, substitution algebras and plain #SAT encodings. None is about all algorithms.
- *The general model admits no unconditional barrier* (Theorem 3.13), and an unconditional cap on oblivious algorithms would separate VP from VNP (Theorem 8.1).
- *The Tutte barrier has two loopholes*: sums over different clusterings, and skeletons that depend on private edges (Theorem 8.6).
- *Much is on paper.* **??** 4.12**??** 4.13, the covering bound (Theorem 4.7), the template corollaries (Theorem 5.2, modulo the Arora–Rao–Vazirani theorem), the Tutte barrier and the meta-barrier are proved on paper only.
- *The Lean statements are about ranks.* The links "cost is at least $\log_2$ of a flattening rank" and "a network gives a factorised matrix" are on paper, as are the hypotheses `hrank` and `hU`.
- *Numerics are not evidence about ρ.* Data at $\lambda \leq 2$ cannot separate $\rho = 1/2$ from anything (Theorem 4.14).
- *Scope of the caps.* They cover [A]'s framework. For [O]'s engine they cover the M_0-template of its main text (Section 5), not the general interface, which Dias and Lima bound by $2^{\Omega(\sqrt{\mathrm{tw}})}$ [29], and not its Engine II, whose bound we have not verified.
- *The product-term caps are not the best known.* Majdoddin's bounds for product terms [43] have the exact constant, need no squaring assumption and cover more configurations; ours are independent and weaker. The correlated bounds are restricted to the two grid merges, and they are weaker than the conjectured N^9.
- *No human has checked the work.* The Lean code and this text were written with AI assistance. The barriers are unconditional statements about their models; they do not depend on the refutations being correct, but their interest does.

12. Conclusion

We have shown that the two counting engines meet the same wall from two sides: dense problems force at least of order $n \log n$ shared bits in every local network we can analyse, and every product-term recursion converts shared bits into time at a rate no better than $1/\log K$ (Theorem 4.6, and sharper [43, Theorem 4.13]), while correlated decompositions of full grid levels do no better than $(\log K)^{-1.415}$ (Theorem 4.12). The paper is a step towards, not a proof of, the statement that the engines do not reach dense counting; a proof of that statement even for oblivious algebraic algorithms would separate VP from VNP (Theorem 8.1). One instance is open and small: the 3×3 grid level, where N^8 is proved and N^9 is conjectured, a case of the "open door" of [43, §4.8]. The cheapest experiment that could change the picture is a decomposition of that level with maximum cost at most $(9 - \varepsilon)\lambda$. An interesting direction on the number-theoretic side is the g-party verification of a product (Theorem 8.3), whose answer would decide whether exact $\pi(x)$ is within the engines' reach.

Acknowledgements

This work was done with AI assistance (Claude Code, Anthropic): the research notes, the Lean development, the numerical checks and the text were produced in sessions with Claude Code. The Lean theorems are checked by the Lean kernel; everything else awaits human expert review. Responsibility for any errors rests with the author.

References

[1] Sanjeev Arora, Satish Rao, and Umesh Vazirani, *Expander flows, geometric embeddings and graph partitioning*, J. ACM **56** (2009), no. 2, Art. 5, doi:10.1145/1502793.1502794.

[2] Per Austrin, Petteri Kaski, and Kaie Kubjas, *Tensor network complexity of multilinear maps*, Theory Comput. **18** (2022), Art. 16, doi:10.4086/toc.2022.v018a016.

[3] Roger C. Baker, Glyn Harman, and János Pintz, *The difference between consecutive primes, II*, Proc. London Math. Soc. (3) **83** (2001), no. 3, 532–562, doi:10.1112/plms/83.3.532.

[4] Alexander I. Barvinok, *Two algorithmic results for the traveling salesman problem*, Math. Oper. Res. **21** (1996), no. 1, 65–84, doi:10.1287/moor.21.1.65.

[5] Richard Bellman, *Dynamic programming treatment of the travelling salesman problem*, J. ACM **9** (1962), no. 1, 61–63.

[6] Vishwas Bhargava and Anamay Tengse, *Explicit commutative ROABPs from partial derivatives*, Proc. 44th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS), LIPIcs, vol. 323, 2024, arXiv:2407.10143, pp. 10:1–10:15.

[7] Deep Bhattacharjee, *Subexponential weighted counting and exact optimisation*, Hexagon preprint hexagon:2610.00594v1, 29 October 2026, https://hexagonmath.org/abs/2610.00594, 2026.

[8] Andreas Björklund, *Determinant sums for undirected Hamiltonicity*, SIAM J. Comput. **43** (2014), no. 1, 280–299, Preliminary version in FOCS 2010.

[9] ———, *Below all subsets for some permutational counting problems*, Proc. 15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT), LIPIcs, vol. 53, 2016, pp. 17:1–17:11.

[10] Andreas Björklund, Radu Curticapean, Thore Husfeldt, Petteri Kaski, and Kevin Pratt, *Fast deterministic chromatic number under the asymptotic rank conjecture*, Proc. 36th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2025, arXiv:2404.04987; doi:10.1137/1.9781611978322.91, pp. 2804–2818.

[11] Andreas Björklund and Thore Husfeldt, *The parity of directed Hamiltonian cycles*, Proc. 54th IEEE Symposium on Foundations of Computer Science (FOCS), 2013, pp. 727–735.

[12] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto, *Computing the Tutte polynomial in vertex-exponential time*, Proc. 49th IEEE Symposium on Foundations of Computer Science (FOCS), 2008, pp. 677–686.

[13] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto, *Set partitioning via inclusion-exclusion*, SIAM J. Comput. **39** (2009), no. 2, 546–563, doi:10.1137/070683933.

[14] Andreas Björklund, Thore Husfeldt, and Isak Lyckberg, *Computing the permanent modulo a prime power*, Inform. Process. Lett. **125** (2017), 20–25, doi:10.1016/j.ipl.2017.04.015.

[15] Andreas Björklund and Ryan Williams, *Computing permanents and counting Hamiltonian cycles by listing dissimilar vectors*, Proc. 46th International Colloquium on Automata, Languages, and Programming (ICALP), LIPIcs, vol. 132, 2019, pp. 25:1–25:14.

[16] Cornelius Brand, Holger Dell, and Marc Roth, *Fine-grained dichotomies for the Tutte plane and Boolean #CSP*, Algorithmica **81** (2019), no. 2, 541–556, Preliminary version in IPEC 2016; arXiv:1606.06581.

[17] Peter Bürgisser, *Completeness and reduction in algebraic complexity theory*, Algorithms and Computation in Mathematics, vol. 7, Springer, 2000, doi:10.1007/978-3-662-04179-6.

[18] Matthias Christandl, Péter Vrana, and Jeroen Zuiddam, *Asymptotic tensor rank of graph tensors: beyond matrix multiplication*, Comput. Complexity **28** (2019), no. 1, 57–111, doi:10.1007/s00037-018-0172-8.

[19] ———, *Universal points in the asymptotic spectrum of tensors*, J. Amer. Math. Soc. **36** (2023), no. 1, 31–79, Preliminary version in STOC 2018; doi:10.1090/jams/996.

[20] Julia Chuzhoy and Rachit Nimavat, *Large minors in expanders*, arXiv:1901.09349, 2019.

[21] Claude (Anthropic), *A $2^{o(n)}$ time algorithm for 3-SAT*, Preprint dated 26 October 2026. Paper and Lean formalisation in the repository https://github.com/anthropics/formal-math, folder `eth`; we use commit `16a5ce41304fcdaea4e0be7c132e1c361dc26da5`, 2026.

[22] Radu Curticapean, Nathan Lindzey, and Jesper Nederlof, *A tight lower bound for counting Hamiltonian cycles via matrix rank*, Proc. 29th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2018, arXiv:1709.02311, pp. 1080–1099.

[23] Marek Cygan, Stefan Kratsch, and Jesper Nederlof, *Fast Hamiltonicity checking via bases of perfect matchings*, J. ACM **65** (2018), no. 3, Art. 12, Preliminary version in STOC 2013.

[24] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michał Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk, *Solving connectivity problems parameterized by treewidth in single exponential time*, Proc. 52nd IEEE Symposium on Foundations of Computer Science (FOCS), 2011, Journal version in ACM Trans. Algorithms 18 (2022), no. 2, pp. 150–159.

[25] Carlos Stein Naves de Brito, *The partition exponent in the graded pieces of the Claude 3-SAT algorithm is optimal for pieces of that shape*, Hexagon preprint hexagon:2610.00692v1, 31 October 2026, https://hexagonmath.org/abs/2610.00692, 2026.

[26] Leonardo de Moura and Sebastian Ullrich, *The Lean 4 theorem prover and programming language*, Automated Deduction – CADE 28, Lecture Notes in Comput. Sci., vol. 12699, Springer, 2021, pp. 625–635.

[27] Marc Deléglise and Joël Rivat, *Computing $\pi(x)$: the Meissel, Lehmer, Lagarias, Miller, Odlyzko method*, Math. Comp. **65** (1996), no. 213, 235–245, doi:10.1090/S0025-5718-96-00674-6.

[28] Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlén, *Exponential time complexity of the permanent and the Tutte polynomial*, ACM Trans. Algorithms **10** (2014), no. 4, Art. 21.

[29] Fabio Dias and Yvens Lima, *Limits and constants of OpenAI's claimed ETH refutation*, arXiv:2611.00517v1, 31 October 2026.

[30] Michael Freedman, László Lovász, and Alexander Schrijver, *Reflection positivity, rank connectivity, and homomorphism of graphs*, J. Amer. Math. Soc. **20** (2007), no. 1, 37–51, doi:10.1090/S0894-0347-06-00529-7.

[31] Carla Groenland, Isja Mannens, Jesper Nederlof, and Krisztina Szilágyi, *Tight bounds for counting colorings and connected edge sets parameterized by cutwidth*, Proc. 39th International Symposium on Theoretical Aspects of Computer Science (STACS), LIPIcs, vol. 219, 2022, arXiv:2110.02730, pp. 36:1–36:20.

[32] Michael Held and Richard M. Karp, *A dynamic programming approach to sequencing problems*, J. Soc. Indust. Appl. Math. **10** (1962), no. 1, 196–210.

[33] Ellis Horowitz and Sartaj Sahni, *Computing partitions with applications to the knapsack problem*, J. ACM **21** (1974), no. 2, 277–292, doi:10.1145/321812.321823.

[34] Pavel Hrubeš, Avi Wigderson, and Amir Yehudayoff, *Non-commutative circuits and the sum-of-squares problem*, J. Amer. Math. Soc. **24** (2011), no. 3, 871–898, doi:10.1090/S0894-0347-2011-00694-2.

[35] François Jaeger, Dirk L. Vertigan, and Dominic J. A. Welsh, *On the computational complexity of the Jones and Tutte polynomials*, Math. Proc. Cambridge Philos. Soc. **108** (1990), no. 1, 35–53.

[36] Tamás Kővári, Vera T. Sós, and Pál Turán, *On a problem of K. Zarankiewicz*, Colloq. Math. **3** (1954), 50–57, doi:10.4064/cm-3-1-50-57.

[37] Jeffrey C. Lagarias and Andrew M. Odlyzko, *Computing $\pi(x)$: an analytic method*, J. Algorithms **8** (1987), no. 2, 173–191, doi:10.1016/0196-6774(87)90037-X.

[38] Radu Curticapean, *Get permanent results faster by counting stars*, arXiv:2611.03636, 2 November 2026.

[39] Baitian Li, *Counting perfect matchings and Hamiltonian cycles faster*, Proc. 53rd International Colloquium on Automata, Languages, and Programming (ICALP), LIPIcs, vol. 374, 2026, arXiv:2309.15422v2, pp. 138:1–138:16.

[40] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas, *Superpolynomial lower bounds against low-depth algebraic circuits*, Proc. 62nd IEEE Symposium on Foundations of Computer Science (FOCS), 2021, doi:10.1109/FOCS52979.2021.00083, pp. 804–814.

[41] Bernt Lindström, *Determinants on semilattices*, Proc. Amer. Math. Soc. **20** (1969), no. 1, 207–208, doi:10.1090/S0002-9939-1969-0238738-4.

[42] László Lovász, *Large networks and graph limits*, Amer. Math. Soc. Colloquium Publications, vol. 60, Amer. Math. Soc., 2012, doi:10.1090/coll/060.

[43] Ruhollah Majdoddin, *Lowered crossover points of Anthropic's subexponential 3-SAT algorithm*, arXiv:2610.41622v1, 30 October 2026.

[44] Jesper Nederlof and Karol Węgrzycki, *Improving Schroeppel and Shamir's algorithm for subset sum via orthogonal vectors*, Proc. 53rd Annual ACM Symposium on Theory of Computing (STOC), 2021, doi:10.1145/3406325.3451024, pp. 1670–1683.

[45] Troy Tri Nguyen, *A sublinear treewidth exponent for binary constraint satisfaction*, Hexagon preprint hexagon:2611.00004v1, 2 November 2026, https://hexagonmath.org/abs/2611.00004, 2026.

[46] Noam Nisan, *Lower bounds for non-commutative computation*, Proc. 23rd Annual ACM Symposium on Theory of Computing (STOC), 1991, doi:10.1145/103418.103462, pp. 410–418.

[47] OpenAI, *The Exponential Time Hypothesis is false*, Preprint dated 17 October 2026, revised 25 October 2026. Paper and Lean formalisation in the repository https://github.com/openai/math; we use commit `7ef5c0cc5ca4b0e072b7a050fca09ab24bc6fa3a`, 2026.

[48] Kevin Pratt, *Breaking the 2^n barrier for graph k-coloring*, arXiv:2607.27159, 2026.

[49] Ran Raz, *Multi-linear formulas for permanent and determinant are of super-polynomial size*, J. ACM **56** (2009), no. 2, Art. 8, doi:10.1145/1502793.1502797.

[50] Isaac Rudich and Louis-Martin Rousseau, *Deterministic factoring in time $N^{o(1)}$*, Hexagon preprint hexagon:2610.00600v1, 29 October 2026, https://hexagonmath.org/2610.00600v1, 2026.

[51] Herbert J. Ryser, *Combinatorial mathematics*, Carus Mathematical Monographs, vol. 14, Math. Assoc. America, 1963.

[52] Richard Schroeppel and Adi Shamir, *A $T = O(2^{n/2})$, $S = O(2^{n/4})$ algorithm for certain NP-complete problems*, SIAM J. Comput. **10** (1981), no. 3, 456–464, doi:10.1137/0210033.

[53] Joseph Shunia, *An improved subexponential bound for deterministic integer factorization*, Hexagon preprint hexagon:2610.00711v1, 2 November 2026, https://hexagonmath.org/2610.00711v1, 2026.

[54] Mihail Stoian, *TSP escapes the $O(2^n n^2)$ curse*, arXiv:2405.03018, 2024.

[55] Terence Tao, Ernest Croot, III, and Harald Helfgott, *Deterministic methods to find primes*, Math. Comp. **81** (2012), no. 278, 1233–1246, arXiv:1009.3956; doi:10.1090/S0025-5718-2011-02542-1.

[56] The mathlib Community, *The Lean mathematical library*, Proc. 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP), ACM, 2020, pp. 367–381.

[57] Vaibhav Tulsyan, *Class numbers, square roots and quadratic non-residues in time $2^{O(n/\sqrt{\log n})}$, assuming the new #SAT algorithms*, Hexagon preprint hexagon:2611.00115v1, 5 November 2026, https://hexagonmath.org/abs/2611.00115, 2026.

[58] ———, *Counting Eulerian circuits in subexponential time, assuming the new #SAT algorithms*, Hexagon preprint hexagon:2611.00088v1, 4 November 2026, https://hexagonmath.org/abs/2611.00088, 2026.

[59] ———, *Counting Hamiltonian cycles in sparse graphs in subexponential time, assuming the new #SAT algorithms*, Hexagon preprint hexagon:2610.00672v1, 31 October 2026, https://hexagonmath.org/abs/2610.00672, 2026.
[60] ———, *eth-hamiltonicity: Lean 4 formalisation and research notes*, Public repository https://github.com/xennygrimmato/eth-hamiltonicity, commit `946c49993b78aa163fcac3f8cb78ce351dd5dcc6`; Lean sources of this paper in `lean/EthFollowup/Barriers`, research notes in `research/`, 2026.
[61] ———, *Exact amplitudes of sparse quantum circuits in subexponential time, assuming the new #SAT algorithms*, Hexagon preprint hexagon:2611.00090v1, 4 November 2026, https://hexagonmath.org/abs/2611.00090, 2026.
[62] ———, *An independent machine-checked proof of deterministic factoring in time* $2^{O(n/\sqrt{\log n})}$*, via the new #SAT algorithms*, Preprint, November 2026; source in the directory `factoring-note/` of [60], 2026.
[63] ———, *The parity of the prime-counting function in time* $x^{o(1)}$, Hexagon preprint hexagon:2611.00116v1, 5 November 2026, https://hexagonmath.org/abs/2611.00116, 2026.
[64] Leslie G. Valiant, *Completeness classes in algebra*, Proc. 11th Annual ACM Symposium on Theory of Computing (STOC), 1979, doi:10.1145/800135.804419, pp. 249–261.
[65] ———, *The complexity of computing the permanent*, Theoret. Comput. Sci. **8** (1979), no. 2, 189–201, doi:10.1016/0304-3975(79)90044-6.
[66] Herbert S. Wilf, *Hadamard determinants, Möbius functions, and the chromatic number of a graph*, Bull. Amer. Math. Soc. **74** (1968), no. 5, 960–964, doi:10.1090/S0002-9904-1968-12104-4.
[67] Ryan Williams, *A new algorithm for optimal 2-constraint satisfaction and its implications*, Theoret. Comput. Sci. **348** (2005), no. 2–3, 357–365, doi:10.1016/j.tcs.2005.09.023.
[68] Or Zamir, *k-coloring is faster than computing the chromatic number*, arXiv:2607.25973v2, 2026.