

The Unit Gap Is Base-Dependent: A Mechanically Verified Refutation and a Cross-Base Census

Luiz Antonio Busnello
Independent Researcher
São Paulo, Brazil
lab@generantis.com.br

Abstract

The gap between a Boolean function’s minimum formula size (fan-out one) and its minimum circuit size — what a formula pays for being unable to share — is **base-dependent**; in particular, the number of variables at which it first exceeds one gate depends on the basis. In the And-Inverter Graph (AIG) basis it does so already at $n = 3$ and reaches 6 at $n = 4$; in the XOR-AND-graph (XAG) basis it stays within a single gate through $n = 4$ (a solver-gated census, its UNSAT side not proof-certified) and first exceeds it at $n = 5$, where two public per-class tables put its maximum at 3. In AIG this refutes a recent claim of Krinkin (*The Unit Gap: How Sharing Works in Boolean Circuits*, arXiv:2603.08033v2; since withdrawn) that in that model the gap is at most one gate for every Boolean function — Theorem 2: $\text{gap} \in \{0, 1\}$ — with a decomposition corollary bounding a shared-gate term $s \in \{0, 1\}$ (Corollary 6). We refute both. Under the paper’s own definition of a formula (fan-out one at every gate), the parity of three variables ($\oplus_3$) is a counterexample: $\text{opt}(\oplus_3) = 6$ — UNSAT for every gate count $k = 1, \dots, 5$, each certified by a DRAT proof checked with **drat-trim** — while $\text{tree}(\oplus_3) = 9$, giving gap 3; in the optimal six-gate circuit the two output-child sub-DAGs intersect in a three-gate parity-2 block, so Krinkin’s own term $s = |D_a \cap D_b| = 3$. **The refutation itself is classical.** The AIG basis with free inversions is U_2 , where the circuit complexity of $\oplus_n$ is known exactly to be $3n - 3$ — Schnorr’s gate-elimination argument, with the exact equality stated and proved in Jukna, Theorem 1.29 — so $\text{opt}(\oplus_3) = 6$; Khrapchenko forces $L(\oplus_3) \geq 9$ leaves, hence $\text{tree}(\oplus_3) \geq 8$ gates (a formula with L leaves has $L - 1$ gates). **So $\text{gap}(\oplus_3) \geq 2 > 1$ follows from two published results and the basis identification above, with no computation of ours in the chain: Theorem 2 was already contradicted by the literature.** The exact value 3 needs $\text{tree}(\oplus_3) = 9$, for which we found no determination in the refereed literature. That formula-side value is certified here by two independent implementations of the formula-size recurrence and the Lean development, and agrees with an exhaustive enumeration published informally in 2011; the circuit side, $\text{opt} = 6$, rests on the DRAT chain above. At $n = 4$ the exact gap is classical again: for $n = 2^k$ the classical construction attains Khrapchenko’s n^2 , so the De Morgan formula size of $\oplus_n$ is exactly n^2 leaves (as recorded by Tarui, who proves the optimal formulas essentially unique), giving 15 gates against $\text{opt}(\oplus_4) = 3 \cdot 4 - 3 = 9$, so $\text{gap}(\oplus_4) = 15 - 9 = 6$ — the maximum of the census below. The error is locatable in the paper: §2 displays $\text{tree}(f) = \min(1 + \text{opt}(a) + \text{opt}(b))$ (the DAG measure opt in the children), while §3’s Bellman operator is $(Tv)(f) = \min(1 + v(a) + v(b))$ (the recursive measure); these are incompatible characterizations of formula size, the first making Theorem 2 a one-line consequence and the second making it false (its fixed point overshoots $\text{opt}(\oplus_3)$ by 3). An exhaustive census over the 222 NPN classes of $n = 4$ shows the failure is structural: 72 of 222 classes have $\text{gap} \geq 2$, the maximum gap 6 attained by three distinct NPN classes (distinguished by their XAG optima 6/5/3), one of which is parity-4. Finally, the phenomenon is base-dependent. In the XAG basis the unit-gap property holds for $n \leq 4$ (with the caveat above): $\text{gap}_{\text{XAG}} = 0$ for all 256 functions of $n = 3$ and $\text{gap}_{\text{XAG}} \in \{0, 1\}$ for all 222 classes of $n = 4$. It fails at $n = 5$. The five-variable function 0x00063357 has $\text{opt}_{\text{XAG}} = 6$ and

$\text{tree}_{\text{XAG}} = 8$, which we establish by a simulated circuit, by SAT (solver-decided, without DRAT), and by a formula-size dynamic program written independently of any published table. Joining two public per-class tables for all 616,126 NPN classes of $n = 5$ — the minimum formula lengths of Lee, Jiang, Mishchenko and Brayton (IWLS 2019) and Knuth’s optimal circuits as distributed in `optimal5` — gives $\text{gap}_{\text{XAG}} \in \{0, 1, 2, 3\}$, with 6,828 classes at $\text{gap}_{\text{XAG}} \geq 2$; that distribution rests on the optimality of the two sources, which publish no certificate. An earlier directed search of ours over 25,373 functions of $n = 5$, all with $\text{opt}_{\text{XAG}} \leq 6$, found no separator among the 25,169 functions whose gap it decided, and agrees with the joined tables on every function it decided. A replicated MIG census matching Soeken et al.’s published size distribution bucket by bucket gives an aggregate external cross-check. The large AIG gaps are consistent with the cost of simulating linear (parity) structure without a native XOR gate.

1 Introduction

The relationship between **formula size** (the smallest fan-out-one circuit for a function) and **circuit size** (the smallest circuit of any fan-out) is one of the oldest quantitative questions in Boolean complexity. Their difference measures how much *sharing* — reusing an intermediate result at more than one place — can buy. Krinkin’s *The Unit Gap* claims that in the And-Inverter Graph model this difference is **at most one gate** for every Boolean function (Theorem 2), with a decomposition corollary bounding the sharing (Corollary 6).

It is not true. This paper does three things.

1. **We refute Theorem 2 and Corollary 6** at $n = 3$ (parity), where the gap is 3 and Krinkin’s shared-gate term is 3. *The counterexample is not new mathematics*: the contradiction with Theorem 2 was already implied by the published literature, since Schnorr’s gate-elimination argument (the exact equality is stated and proved in Jukna) fixes the circuit side and Khrapchenko bounds the formula side, and the two together give $\text{gap}(\oplus_3) \geq 2 > 1$. What we add is to notice it, to certify it mechanically (a DRAT chain at every $k = 1, \dots, 5$ checked by an independent tool, two dynamic programs, and a Lean 4 formalization), and above all **to locate the error inside the paper**: §2 and §3 give incompatible characterizations of $\text{tree}(f)$, one making Theorem 2 trivially true and the other making it false. A refutation that only exhibits a counterexample leaves the author guessing where the argument broke; this one does not.
2. **We show the failure is structural**, computing the complete gap census over all 222 NPN classes of $n = 4$: 32.4% of classes violate the unit-gap bound, the maximum gap, 6, attained by three distinct NPN classes, one being parity.
3. **We show the phenomenon is base-dependent**. The property, false in AIG already at $n = 3$, holds in the XAG basis for $n \leq 4$ (exhaustive, by a solver-gated census whose UNSAT side is not proof-certified) and fails there at $n = 5$, where we exhibit a separator and, from public per-class data and under the optimality of its two sources, the whole gap distribution. The arity at which the unit gap breaks depends on the basis. The large parity AIG gaps are consistent with the price of simulating parity without a native XOR gate.

Our method is **exact synthesis via SAT**: “is there a k -gate circuit for f ?” is a CNF decided by a modern solver. The encoder is a normalized exact-size encoding (every non-output gate used, no duplicate gates); by a minimality lemma, $\text{opt}(f) = k$ implies the k -encoding is satisfiable, so a lower bound $\text{opt}(f) \geq m$ requires UNSAT at *every* $k = 1, \dots, m - 1$ — which is why our circuit lower bounds certify the whole chain, not just the last step. The package was reviewed adversarially

by four independent language-model families before external communication; the mathematics rests on the certificates, not that review. We claim novelty only for the refutation and, pending the specialist check of §7, for the gap-by-basis census; the individual optimal-size tables in XAG and MIG are classical or previously published, and at $n = 5$ the XAG gap is a subtraction of two public per-class columns (§5), so what is ours there is the join and the separator check, not the data.

On refuting a withdrawn preprint. Krinkin withdrew [10] after checking the counterexample himself and confirming it the same day; the acknowledgements record how. A withdrawal retracts a claim; it does not answer the question the claim was about. This notice is unusually complete — it records the counterexample in checkable form, the diagnosis, the exact value $\text{gap}(\oplus_3) = 3$, and the scope: Theorem 2 false, Corollary 6 false, Theorem 7 without its proof, Theorems 3 and 4 surviving — and it attributes the counterexample and the repair of Theorem 3 by name. The notice also gives a short analytic refutation and links the artefacts; this paper supplies the detailed verification account and the cross-base census. The companion catalogue [9] was not withdrawn, supplies the $n = 4$ optima this census rests on, and had two open entries that are closed here, so the surviving half of the work is load-bearing for what follows. The conjecture is also natural enough to be arrived at again by someone who never saw the preprint, which is the case in which a published counterexample is worth more than a retraction.

Above all, the negative verdict is not what this paper is for. Its claim is the boundary: the gap reaches 6 in AIG already at $n = 4$, and stays within a single gate in XAG through $n = 4$, on a census whose UNSAT side is solver-gated rather than proof-certified (§6), before exceeding it at $n = 5$. That is a statement about bases: the unit-gap intuition fails in both AIG and XAG, at an arity that depends on the basis, and no withdrawal establishes that.

2 Preliminaries

We quote the paper’s definitions to fix the target precisely (arXiv:2603.08033v2, §2; notation-normalized).

A circuit is an AIG where gates may have fan-out greater than one. A formula (or tree circuit) has fan-out exactly one at every gate. We write $\text{tree}(f)$ for the minimum formula size. Since every formula is a circuit, $\text{tree}(f) \geq \text{opt}(f)$. The gap is $\text{gap}(f) = \text{tree}(f) - \text{opt}(f)$.

And, for the decomposition (§2):

Every AIG gate computes $f = a \wedge b$ (with possible edge complementation). The minimum formula size satisfies $\text{tree}(f) = \min \text{ over } a, b: f = a \wedge b \text{ or } f = \neg(a \wedge b) \text{ of } (1 + \text{opt}(a) + \text{opt}(b))$. The trivial decomposition $f = 1 \wedge f$ always achieves $1 + \text{opt}(f)$.

An **AIG** over $x_1, \dots, x_n$ and the constant 1 is a DAG of two-input AND gates with free edge complementation; $\text{opt}(f)$ is the minimum number of AND gates. With free complementation an AND gate realises every binary Boolean function except XOR and XNOR at unit cost, so the AIG basis is exactly U_2 , and opt coincides gate for gate with U_2 -circuit size. *The exclusion of XOR is what makes this basis the right one for the classical results used below: U_2 is the full binary basis B_2 minus XOR and XNOR, so parity is not free in it.* (The output gate has fan-out zero; the “fan-out one” condition is Krinkin’s, for the non-output gates of a tree.) An **XAG** adds a native XOR gate; an **MIG** (Majority-Inverter Graph) uses 3-input majority gates. All exhaustive results computed here are for $n \leq 4$; the $n = 5$ XAG distribution of §5 is a join of published per-class data.

Exact synthesis. For fixed (n, k) we one-hot-select each gate’s operation and operands among earlier nodes and constrain the output to compute f up to polarity. The encoding is *normalized*: every non-output gate must be used and duplicate gate options are forbidden. These are sound for minimality: SAT at k yields a valid k -gate circuit, and if $\text{opt}(f) = k$ a minimality lemma guarantees the k -encoding is satisfiable. Hence $\text{opt}(f)$ is the least k with SAT, and a lower bound $\text{opt}(f) \geq m$ requires UNSAT at *every* $k = 1, \dots, m - 1$. For formula synthesis, we impose fan-out one and disable the circuit-only duplicate-gate prohibition, which turns the same encoder into a $\text{tree}(f)$ oracle (validated in Appendix B).

3 The refutation at $n = 3$

The paper gives incompatible characterizations of $\text{tree}(f)$. The §2 identity puts the DAG measure opt in the children; with opt there, the trivial decomposition $f = 1 \wedge f$ yields $\text{tree}(f) \leq 1 + \text{opt}(f)$ for every f , so Theorem 2 is a one-line consequence — but the minimized object is not a tree (the children may be internally shared DAGs). In §3, however, the paper defines the Bellman operator (notation-normalized):

$$(Tv)(f) = \min \text{ over } a, b: f = a \wedge b \text{ of } (1 + v(a) + v(b)); \text{ iterating } T \text{ from the base case } v_0(x_i) = v_0(1) = 0 \text{ computes } \text{tree}(f).$$

with the **recursive** measure v in the children. Here and below the operator is read with literals and constants fixed at cost zero and with free output complementation, so that the decomposition is over $f = a \wedge b$ or $f = \neg(a \wedge b)$. At its fixed point this reads $\text{tree}(f) = \min(1 + \text{tree}(a) + \text{tree}(b))$, a different object from the §2 identity: as global characterizations of formula size the two coincide term-by-term only on children for which $\text{opt} = \text{tree}$. Krinkin further states — as a *consequence of Theorem 2*, not a definition — that T ’s fixed point “overshoots $\text{opt}(f)$ by exactly $\text{gap}(f) \in \{0, 1\}$ ”, which collapses together with Theorem 2.¹ For $\oplus_3$ the recursive T computes 9, overshooting $\text{opt} = 6$ by 3; under the §3 reading, Theorem 2 is false.

- $\text{opt}(\oplus_3) = 6$. *Upper bound:* the six-gate circuit

$$\begin{aligned} g_1 &= x_1 \wedge \neg x_2, & g_3 &= \neg g_1 \wedge \neg g_2, & g_5 &= g_3 \wedge x_3, \\ g_2 &= \neg x_1 \wedge x_2, & g_4 &= \neg g_3 \wedge \neg x_3, & g_6 &= \neg g_4 \wedge \neg g_5, \end{aligned} \tag{1}$$

with output $\neg g_6$ (inversions are free, so the output polarity costs nothing). Here $g_3 = \neg(x_1 \oplus x_2)$ is the *only* gate of fan-out two, which is exactly why the circuit is not a tree; the two children of the output, g_4 and g_5 , have sub-DAGs meeting in $\{g_1, g_2, g_3\}$ — this is the $s = 3$ of the Corollary 6 refutation below, and it can be read off the display. Verified by simulation on all eight inputs and in Lean 4. *Lower bound:* UNSAT at **every** $k = 1, 2, 3, 4, 5$ (kissat 4.0.4), each with a DRAT proof checked by `drat-trim` (`s VERIFIED`; transcript and hashes in Appendix C). The full $k = 1, \dots, 5$ chain — not only $k = 5$ — certifies $\text{opt} \geq 6$ under the normalized exact-size encoding. *The value is also classical:* since the AIG basis is U_2 (§2), the classical bound — the U_2 circuit complexity of $\oplus_n$ is exactly $3n - 3$ [5], resting on Schnorr’s induction [17] and

¹Krinkin frames the same collapse in the language of dynamic programming: once gates are shared, an optimum for the whole does not restrict to optima on its parts, so opt cannot be a fixed point of a Bellman operator over disjoint children — which is equally why the §2 identity fails. This is the content of his Remark 5. (K. Krinkin, personal communication, 2026.)

restated in [14]² — gives $\text{opt}(\oplus_3) = 6$ and $\text{opt}(\oplus_4) = 9$ directly, in agreement with the values computed here.

- $\text{tree}(\oplus_3) = 9$. *Upper bound:* the nine-gate formula is (1) with the shared block duplicated, which is what fan-out one forces:

$$\neg(\neg(\neg P_1 \wedge \neg x_3) \wedge \neg(P_2 \wedge x_3)), \quad P_1 = P_2 = \neg(x_1 \oplus x_2), \quad (2)$$

where each P_i is a private three-gate copy of the block that (1) computes once as g_3 , and the outer inversion is free, as for the output $\neg g_6$ of (1). Counting: 3 + 3 for the two copies, plus the three gates above them, is 9; the circuit pays 3 for one copy plus the same three, which is 6. **The gap is the duplicated block**, and its size is Krinkin’s $s = 3$. Checked the same way as (1). *Lower bound:* the exact fixed point of the tree recursion over all 256 functions, independently confirmed by a layer-by-layer enumeration (new functions per cost 1, . . . , 9: 24, 64, 30, 80, 32, 0, 16, 0, 2 — the two cost-9 functions are parity and its complement).³

- **The refutation is citational.** The identification of §2 extends from circuits to formulas: with free complementation every U_2 gate is an AND or an OR of its two arguments up to edge inversions, so pushing inversions to the leaves turns an AIG formula into a De Morgan formula with the same number of binary gates and back. Hence $\text{tree} = \text{tree}_{U_2} = \text{tree}_{\text{De Morgan}}$, and a binary formula with L leaves has $L - 1$ gates. Khrapchenko [6] then gives $L(\oplus_3) \geq |E|^2/(|A||B|) = 144/16 = 9$ leaves, with A and B the four odd- and four even-weight points of $\{0, 1\}^3$ and E the twelve cube edges between them, hence $\text{tree}(\oplus_3) \geq 8$ gates. Together with the explicit six-gate circuit ($\text{opt} \leq 6$), $\text{gap}(\oplus_3) \geq 2$ follows **using only Khrapchenko and that circuit** — independent of the tree DP and the DRAT chain. Substituting the classical bound for the explicit circuit removes even that: $\text{opt}(\oplus_3) = 6$ [5] and $\text{tree}(\oplus_3) \geq 8$ (Khrapchenko) yield $\text{gap}(\oplus_3) \geq 2 > 1$ from two published classical results and the basis identification of §2, with no computation of ours in the chain. **That alone contradicts Theorem 2.**

The *exact* value is a different matter. For $n = 2^k$ the classical balanced parity construction attains Khrapchenko’s n^2 -leaf lower bound, so the formula size is exactly n^2 leaves (Tarui [21] records the exact value and proves the optimal formulas essentially unique); this fixes $L(\oplus_4) = 16$ leaves, i.e. $\text{tree}(\oplus_4) = 15$ gates, and hence $\text{gap}(\oplus_4) = 6$, the maximum gap found in the $n = 4$ census of §4 — an external check on the census, not merely on the counterexample. For $n = 3$ we found no determination in the refereed literature.⁴ Accordingly $\text{tree}(\oplus_3) = 9$ is certified here rather than cited: by the tree dynamic program, independently re-implemented,

²Attribution. The gate-elimination induction is Schnorr’s, in the 1974 paper [17]; the exact equality $3(n - 1)$ is stated and proved by Jukna (Theorem 1.29), whose following remark extends it from $\{\wedge, \vee, \neg\}$ to every two-input basis except $\oplus$ and its negation, which is U_2 . Jukna and Morizumi both attribute the equality to Schnorr’s 1974 paper. In the 1974 original, Korollar 4.11 reads $3n - 5$ at the parameter in question and the form $3(n - 1)$ occurs nowhere (first reported to us by K. Krinkin, and checked at source). We therefore cite Jukna for the exact equality used here.

³The gap distribution at $n = 3$ ({gap 0: 214, 1: 40, 3: 2}) is reproduced by `tree_gap_n3.py`; the per-cost histogram above was re-derived independently during review.

⁴The obvious candidate does not survive. Lee’s rank technique [12] was announced as determining the formula size of parity for every n , as $2^\ell(2^\ell + 3r)$ leaves for $n = 2^\ell + r$, which would give 10 leaves at $n = 3$. Lee has since retracted that claim, crediting Dmitry Cherukhin with finding the error: it is in his Proposition 2, which assumes an optimal selection function takes each s_i to be a power of two. His correction page cites Rychkov [16], a reference also due to Cherukhin, for the current best lower bounds — $n^2 + 3$ for odd $n \geq 5$ and $n^2 + 2$ for even $n \geq 6$ that are not powers of two, neither of which covers $n = 3$. The rank technique itself, and Lee’s observation that no bound above n^2 is reachable by Khrapchenko’s method, are unaffected.

and by the Lean development. The value is not new — an exhaustive enumeration computed by Cox and Healy and reported by Cox [1] gives 9 AND/OR operators for $\oplus_3$ and 15 for $\oplus_4$, and writes out the optimal $n = 3$ formula — and the agreement is an external check on both.

Hence $\text{gap}(\oplus_3) = 3$, refuting Theorem 2.

Corollary 6 is refuted with Krinkin’s own definition. He states (§5; notation-normalized):

Let C be an optimal AIG, and let g be any gate computing $f = a \wedge b$. Let D_a, D_b be the sub-DAGs computing a and b . Then $\text{opt}(f) = 1 + \text{opt}(a) + \text{opt}(b) - s(a, b)$, where $s(a, b) = |D_a \cap D_b| \in \{0, 1\}$ counts the shared gates.

Structurally, take g to be the output gate of the six-gate optimum for $\oplus_3$: its children a, b share the three-gate parity-2 block (computing $\neg(x_1 \oplus x_2)$ in the Lean witness `c6`; $x_1 \oplus x_2$ up to free output polarity), so $|D_a \cap D_b| = 3$ and, since each child has $\text{opt} = 4$, $s = 1 + 4 + 4 - 6 = 3 \notin \{0, 1\}$. One child, $(x_1 \oplus x_2) \wedge \neg x_3$, has its $\text{opt} = 4$ certified by a DRAT chain (UNSAT at $k = 1, 2, 3$; Appendix C); the other child is NPN-equivalent (input/output polarity), hence also $\text{opt} = 4$. Krinkin’s own $s \leq 1$ derivation is separately unsound: he needs $s \leq \text{tree}(f) - \text{opt}(f)$, but the §2 identity gives $\text{tree}(f) \leq 1 + \text{opt}(a) + \text{opt}(b)$, i.e. $s \geq \text{tree}(f) - \text{opt}(f)$ — the reverse of the comparison the proof requires. (For parity itself $s = \text{tree} - \text{opt} = 3$, an equality, so Krinkin’s first inequality happens to hold there; his chain then breaks at the *second* step, $\text{tree} - \text{opt} \leq 1$, which is Theorem 2.) Krinkin’s Table 3 reports $s = 1$ for the complete $\text{opt}-6$ $n = 3$ bucket, which includes the parity decompositions — consistent with the table having been computed under the constraint $s \leq 1$, and in conflict with the structural count $s = 3$.

Fate of the other results. Theorem 7 (a classification assuming $\text{gap} = 1$) loses its proof, which depends on Corollary 6. The parity counterexample does not itself refute the conditional statement of Theorem 7 under the standard definition, since $\oplus_3$ has $\text{gap} 3$, not 1. A separate note refutes that statement with a four-variable function of $\text{gap} 1$, NPN class `0x03de` ($\text{opt} = 6$ and $\text{tree} = 7$, both DRAT-certified), every optimal circuit of which has at least two gates of fan-out two.⁵

The statement of Theorem 3 stands; Theorem 4 we checked separately, outside this paper. Theorem 3’s published proof applies Lemma 1 (gate elimination) to the sub-DAG S below the shared gate g and writes $|S| \geq k - 1$, which fails when g is at input level; a corrected incidence-count proof is in Appendix A.

Diagnostic signature. Krinkin’s complete $n = 3$ table (Table 1) reports two functions with $\text{gap} = 1$ at $\text{opt} = 6$. Evaluating the §2 identity (with opt in the children) on all 256 functions reproduces every $n = 3$ cell of that table, and its two $\text{gap}-1$ functions at $\text{opt} = 6$ are parity and its complement ($1 + \text{opt}(\text{parity}) + \text{opt}(1) = 7$), whose true formula size is 9. The error is visible in the paper’s data.

The other sixteen functions of that column are affected in the same way. Recomputing the true gap of all 256 functions (opt by SAT, tree by the formula dynamic program above) gives, among the 18 functions with $\text{opt} = 6$, $\text{gap} 1$ for 16 and $\text{gap} 3$ for 2 (parity and its complement), where Table 1 reports 16 with $\text{gap} 0$ and 2 with $\text{gap} 1$. The $\text{opt} = 6$ column of Table 1 therefore records the §2 identity, not the gap ; it is the same error as Theorem 2’s, not a second one. Every other column of

⁵`theorem7/proofs/technote_thm7_refutation.md` in the public repository <https://github.com/totobusnello/unit-gap-artefacts>. That directory is not part of the archived versions of the Zenodo record of Appendix C, and the result is outside the scope of this paper.

the table — the counts per opt and, at opt = 4, the 24 functions with gap 1 — agrees with the true gap, as does the total of 18 functions at opt = 6. (The recount joins the opt table and the tree recursion of `tree_gap_n3.py`; the per-cell script is `check-krinkin-table1-n3-truegap.py`, Appendix C.)

Verification. The refutation is formalized in Lean 4 (4.31.0; the same proof code re-checked under 4.33.0) without Mathlib (`UnitGap.lean`). It proves $\text{gap}(\oplus_3) \geq 3$: a six-gate circuit computes parity, an explicit nine-gate formula computes it, and no formula with at most eight gates does, so $\text{tree}(\oplus_3) = 9$ is in Lean. The matching lower bound $\text{opt}(\oplus_3) \geq 6$ is not; it rests on the DRAT chain ($k = 1, \dots, 5$). In `UnitGap.lean` a function of three variables is its 8-bit truth table ($x_1, x_2, x_3 \mapsto 0xAA, 0xCC, 0xF0$), and an AND gate is bitwise conjunction. The witnesses and the DP’s structural framework are kernel-checked (`decide`, induction); the finite fact $\text{par3} \notin D_8$ — hence $\text{tree}(\oplus_3) \geq 9$ — is discharged by `native_decide` (compiler-trusted, not kernel). We report this precisely rather than as blanket “kernel-checked” (see §6).

4 The failure is structural: AIG gap census at $n = 4$

We computed $\text{tree}(f)$ for all 65,536 functions by a layered dynamic program (independently re-implemented as a global Bellman fixed point with explicit polarities — all 65,536 cells agree), joined with the complete exact opt catalog. The distribution of gap over the 222 NPN classes:

gap	0	1	2	3	4	5	6
classes	93	57	40	13	14	2	3

72 of the 222 NPN classes (32.4% of classes) have $\text{gap} \geq 2$ (a fraction of classes, not of all 65,536 functions, whose orbit sizes differ). The maximum gap 6 is attained by three **distinct** NPN classes $\{0x1668, 0x16e9, 0x6996\}$ — all with opt = 9, tree = 15 — distinguished by their XAG optima ($\text{opt}_{\text{XAG}} = 6, 5, 3$ respectively; opt_{XAG} is NPN-invariant, so distinct values force distinct classes). One of the three is parity-4 ($0x6996$, $\text{opt}_{\text{XAG}} = 3$). For parity-4, Khrapchenko’s bound (≥ 16 leaves $\implies \geq 15$ gates) meets tree = 15 with equality. The embedded $n = 3$ functions reproduce the $n = 3$ table 256/256. Parity is thus a witness of the maximum gap, not its exclusive source: the census is consistent with, but does not by itself isolate, “linear structure without native XOR” as the mechanism for all 72 violating classes.

5 Base dependence: XAG at $n \leq 5$, and MIG

The large parity AIG gaps are consistent with the cost of building parity from AND gates, suggesting a basis artifact.

XAG, $n \leq 4$ (exhaustive). With a native XOR gate, the pipeline (encoder validated by an independent exhaustive enumeration at $n = 3$ and by a formula-encoder gate on all 222 $n = 4$ classes) gives opt_{XAG} maximum 7 (strictly below AIG in 190/222 classes; $\text{opt}_{\text{XAG}}(\oplus_4) = 3$) and tree_{XAG} maximum 7. The gap census (`census/xag-n4/npn4_xag_gap.csv`): $n = 3$ — $\text{gap}_{\text{XAG}} = 0$ for all 256 functions; $n = 4$ — $\text{gap}_{\text{XAG}} \in \{0, 1\}$, distributed $\{0 : 218, 1 : 4\}$. The unit-gap property, false in AIG, **holds in XAG for $n \leq 4$** under this separately gated encoder whose UNSAT instances are not proof-certified (§6).

XAG, $n = 5$: the unit gap fails. A per-class XAG census of $n = 5$ need not be computed: it can be read off published data. Lee, Jiang, Mishchenko and Brayton [11] enumerate minimum fanout-free circuits for every NPN class of up to five inputs, over two-input AND and XOR gates with free inverters, and publish all minimum formulas (the `Lf1ib5` library); their length $L(f)$ counts binary operators, so it is our tree_{XAG} , with no -1 . Knuth’s optimal circuits over B_2 for all 616,126 classes, computed for TAOCP [7] and distributed in `optimal5` [2], give $C(f)$, which is our opt_{XAG} , since under free inversion every non-degenerate B_2 gate is one XAG gate. Before using the join we checked the units: the two sources key the same 616,126 classes; each column reproduces its own source’s published distribution; and the 222 classes of $n = 5$ with a fictitious variable, projected to four variables, reproduce our $n = 4$ census class by class in opt_{XAG} , tree_{XAG} and gap_{XAG} (222/222, including the four classes of gap 1, whereas $\text{tree}_{\text{XAG}} = L - 1$ would match none). Over all 616,126 classes:

$\text{gap}_{\text{XAG}} = L - C$	0	1	2	3
classes	365,053	244,245	6,815	13

So **6,828 classes of $n = 5$ have $\text{gap}_{\text{XAG}} \geq 2$** , and the maximum is 3: ten classes with $(\text{opt}_{\text{XAG}}, \text{tree}_{\text{XAG}}) = (8, 11)$ and three with $(9, 12)$. The smallest separators have $\text{opt}_{\text{XAG}} = 6$ (five classes, each with $\text{tree}_{\text{XAG}} = 8$); most have $\text{opt}_{\text{XAG}} \in \{8, 9, 10\}$. These figures rest on the optimality of both sources, and neither publishes a lower-bound certificate. Lee et al. report that their table “appeared to be coherent with” the data in TAOCP; whether Knuth tabulates the difference we have not checked.

One separator without the published tables. That a separator exists does not depend on either source’s optimality claims. Take `0x00063357` (a 32-bit truth table, the least representative of its NPN class). A six-gate XAG circuit, taken from `optimal5` and re-simulated against the truth table, gives $\text{opt}_{\text{XAG}} \leq 6$, and our circuit encoder is UNSAT at every $k = 1, \dots, 5$ (solver-decided, without DRAT), so $\text{opt}_{\text{XAG}} = 6$. On the formula side, $\text{tree}_{\text{XAG}} \geq 8$ holds by two independent routes: a formula-size dynamic program written from scratch, which builds the exact layers of five-variable XAG formulas up to six gates and tests membership in the seven-gate layer directly; and our formula encoder, UNSAT at every $k = 1, \dots, 7$. An eight-gate formula (from `Lf1ib5`, evaluated, and found again by the encoder at $k = 8$) gives $\text{tree}_{\text{XAG}} = 8$. Hence $\text{gap}_{\text{XAG}}(\text{0x00063357}) = 2$. The other four separators with $\text{opt}_{\text{XAG}} = 6$ pass both formula-side checks, and their six-gate circuits re-simulate, so they too have $\text{gap}_{\text{XAG}} \geq 2$ independently of the two sources.

XAG, $n = 5$ (our directed search). Before joining the published data, we had searched for a *separator* ($\text{gap}_{\text{XAG}} \geq 2$) by generating candidate functions from random XAG circuits with sharing and XOR bias, keeping those essential in all five variables, and deciding opt_{XAG} and tree_{XAG} exactly by SAT (ascending search, 20 s cap). This is a **generator-reachable, solver-decided sample**, not a probability statement: functions whose opt timed out or exceeded the cap are not written and thus not counted, so the sample is conditional on easy circuit optimization. The formula encoder was gated before the search (matching the $n = 4$ DP on all 222 classes, including the four with $\text{gap}_{\text{XAG}} = 1$); an early false positive (parity-5) was fixed by ascending from opt (Appendix B).

Over **25,373 distinct** essential functions of five variables (deduplicated across workers):

No separator was certified. All **25,169 gap-decided** functions have $\text{gap}_{\text{XAG}} \in \{0, 1\}$; the 204 inconclusive cases are formula-search timeouts and could in principle be separators. As an added check, all 294 gap-1 functions and a sample of 200 gap-0 functions were re-decoded and

verdict	functions	%
gap 0	24,875	98.04
gap 1	294	1.16
inconclusive (formula-search timeout)	204	0.80
separator ($\text{gap}_{\text{XAG}} \geq 2$)	0	0

simulation-verified (optimal circuit and formula both compute the truth table); 494/494 passed (Appendix C). This is a **directed sample, not a proof**: a separator of high opt, or outside the generation bias, is not excluded. What the search establishes is that no separator appears among 25,169 gap-decided, generator-reachable essential functions of $n = 5$.

The join explains that outcome. The 25,373 functions fall into 903 NPN classes, all with $\text{opt}_{\text{XAG}} \in \{4, 5, 6\}$, and the search agrees with the joined tables on every function it decided: $\text{opt}_{\text{XAG}} = C$ on 25,373 of 25,373 and $\text{tree}_{\text{XAG}} = L$ on all 25,169 gap-decided ones. Separators begin at $\text{opt}_{\text{XAG}} = 6$, where only 5 of the 1,916 classes with $\text{opt}_{\text{XAG}} = 6$ and five essential variables separate, and none occurred among the gap-decided functions of our directed sample. The search is reported as performed; it is not evidence about $n = 5$, which the join settles under the optimality of its two sources.

MIG (cross-check). Soeken, Amarù, Gaillardon and De Micheli [18] published the optimal MIG size distribution for the 222 classes of $n = 4$. We re-derived the census with a different solver (kissat vs. Z3) and encoding and matched the published **size distribution** $\{0:2, 1:2, 2:5, 3:18, 4:42, 5:117, 6:35, 7:1\}$ **bucket by bucket**, including the unique 7-node class. (Their table publishes bucket counts. A per-class map is distributed with the mockturtle library [13] (`mig_npn.hpp`, since 2018), one MIG per NPN class described as size-optimum; our per-class values agree with its circuit sizes on all 222 classes. Those circuits are upper bounds without a published lower-bound certificate, so the agreement checks our per-class values from above, not our UNSAT side.)

	AIG	XAG	MIG
max opt	10	7	7
$\text{opt}(\oplus_4) / \text{tree}(\oplus_4)$	9 / 15	3 / 3	6 / —
gap census	up to 6; 32.4% of classes ≥ 2	$\{0, 1\}$	deferred

Cross-base summary ($n = 4$). (The MIG gap column is not computed here; both marginals, $C(f)$ and $L(f)$, are published by Soeken et al. [18], Table II.)

6 Verification

The two tree implementations are independent *implementations of the same Bellman recurrence* (layered vs. fixed-point) — strong bug detection. For three-variable AIG formulas, the Lean completeness lemma proves the direction needed to turn the finite DP exclusion into a formula lower bound; the four-variable census is checked by agreement of the two implementations. The four-model-family adversarial review is provenance, not part of the verification chain.

7 Related work and novelty

- **Krinkin**, [arXiv:2603.08033](#) [10] is the refuted paper (since withdrawn); [arXiv:2603.09379](#) [9] (a constructive bound on circuit size under truth-table perturbation, with an NPN-4 AIG catalog whose two open entries — lower bounds; ten-gate circuits for both were already public — we closed) is a companion note. Both were communicated to the author, who confirmed the refutation independently; the exact values closing the catalogue’s two open entries, and a proof for the unit-gap paper’s Theorem 3, were contributed back ([krinkin/bounds#2](#), [krinkin/unit-gap#2](#)).
- **MIG**: the optimal $n = 4$ sizes were previously published (Soeken et al., DATE 2016 [18]; TCAD 2017 [19]), and a per-class database ships with *mockturtle* [13]; our census is replication, claimed only as a cross-check.
- **XAG total-gate count** is essentially the classical combinational complexity over B_2 , tabulated for small n by Knuth (TAOCP 7.1.2) [7]; our maximum of 7 agrees (for $n = 5$, Knuth’s values are public per class in *optimal5* [2]). We claim **no novelty for the XAG numbers themselves**.
- **Both sizes of parity are known, and neither is ours.** The U_2 circuit complexity of $\oplus_n$ is known exactly, $3n - 3$: the gate-elimination induction is Schnorr’s [17] and the equality is stated by Jukna [5]. On the formula side Khrapchenko’s n^2 lower bound is classical and, for $n = 2^k$, attained [21]. Since the AIG basis with free inversions is U_2 (§2), that fixes $\text{opt}(\oplus_3) = 6$ and $\text{tree}(\oplus_3) \geq 8$ — already enough to refute Theorem 2 — and fixes the $n = 4$ values exactly ($\text{opt} = 9$, $\text{tree} = 15$, $\text{gap} = 6$, the census maximum). For $\text{tree}(\oplus_3) = 9$ we found no determination in the refereed literature, but the value is not new either: the exhaustive enumeration that Cox reports, computed with Healy [1], finds parity a hardest function at both $n = 3$ and $n = 4$, needing 9 and 15 AND/OR operators, and exhibits the optimal $n = 3$ formula. Both agree with the values computed here. We claim **no novelty for any of these sizes**; they are used as an external check and to make the refutation citational. What we did not find in the literature is the two placed side by side: the circuit bound and the formula bound come from different traditions — gate elimination and Khrapchenko’s measure — and reading them together requires only the basis identification above. The ingredients are classical — Khrapchenko’s formula bound from 1971 and Schnorr’s gate-elimination induction from 1974, with the exact equality in the form used here stated and proved by Jukna — so the facts that contradict Theorem 2 were on the record well before it was stated; what was missing was a reason to subtract one from the other.
- **Exhaustive catalogs of small functions** — exact synthesis over NPN-4 classes (Soeken et al. [20]; Haaswijk [3]; the per-class AIG, XAG and MIG databases distributed with *mockturtle* [13], whose AIG and XAG circuits attain the optimum of every non-constant class by our computation, although the library states no optimality for them), SAT-based circuit search [8], and ESOP exact synthesis [15] — tabulate optimal *circuits* (or, for ESOP, representations minimizing the number of product terms, a different objective from formula gate count). Public data do report both measures for the same functions over the *complete* two-input basis B_2 : Knuth’s tables for $n \leq 4$ (TAOCP 4A, §7.1.2), which Lee et al. describe as coherent with their own formula table (we have not consulted TAOCP itself), and, for every NPN class of $n = 5$, the formula library of Lee et al. [11] together with Knuth’s optimal circuits as distributed in *optimal5* [2]. These B_2 sizes coincide with our XAG sizes under free

inversion, so our XAG sizes reproduce previously published values: at $n = 5$ the columns used here come from Lee et al.’s Lfib5 and from Knuth’s circuits as distributed in `optimal5`, and we claim no novelty for them. Our AIG sizes instead use U_2 , where XOR and XNOR are excluded. We did not find a catalog reporting both measures for the same functions *in the AIG basis*. The closest related line is the study of **fanout-free (read-once) functions** [4], which concerns formulas in which each *variable* occurs once. That is strictly stronger than $\text{gap}(f) = 0$, which only asks that some optimal circuit have fan-out one at every *gate*, variables being free to reappear as leaves: $x_1 \oplus x_2$ has $\text{gap} = 0$ in AIG ($\text{opt} = \text{tree} = 3$) and is not read-once. Neither line tabulates the gap.

- **Contributions reported here:** (i) the refutation of Theorem 2 and Corollary 6, with machine-checked certificates and a Lean formalization, and the localization of the error to the paper’s own two incompatible characterizations in §2 and §3; and (ii) an *explicitly tabulated* gap-by-basis comparison (tree – opt per NPN class) in AIG and XAG at $n \leq 4$, and in XAG at $n = 5$ by joining the public per-class data above, with one separator checked without relying on their optimality claims. The tree – opt difference is a subtraction of columns each of which is classical, so (ii) is a contribution of organized data and the base-dependence observation, not of a new measure; it is likely implicit in any catalog publishing both sizes, and at $n = 5$ the two columns are public. We searched Knuth’s B_2 data as distributed in `optimal5`, the MIG/AIG and NPN-4 exact-synthesis catalogs (including mockturtle’s databases) and the IWLS 2019 formula library, the ESOP and read-once/fanout-free literature, and the standard EDA benchmark collections (IWLS, the EPFL combinational suite, LGSynth) without finding the gap-by-basis comparison stated explicitly, but treat (ii) as “apparently unreported” pending a specialist check.

8 Open questions

1. What generates $\text{gap}_{\text{XAG}} \geq 2$ at $n = 5$? Version 1.0 of this paper asked whether gap_{XAG} stays in $\{0, 1\}$ at $n = 5$; the answer is no (§5). The 13 classes of gap 3 ($\text{opt}_{\text{XAG}} \in \{8, 9\}$) and the five separators with $\text{opt}_{\text{XAG}} = 6$ are natural starting points.
2. Is there a structural reason the XAG unit gap holds through $n = 4$? No theorem can bound gap_{XAG} by one for every n , since the bound fails at $n = 5$; what remains open is why it holds below.

Acknowledgements

I thank Kirill Krinkin. Told that Theorem 2 of [10] was false, he checked the counterexample himself — by Khrapchenko’s bound, independently of anything presented here — confirmed it the same day, and said so plainly. He has withdrawn the preprint (arXiv:2603.08033). His reading of the damage at the time of the withdrawal is the one adopted in §3, including the distinction that is easiest to get wrong: Corollary 6 falls with Theorem 2, and Theorem 7 loses its proof without being refuted by the parity counterexample, while Theorems 3 and 4 stand. He raised no question of priority.

Conceding a refutation quickly and in full is harder than it looks, and rarer than it should be. His companion catalogue [9] also supplied the $n = 4$ optima that the census rests on; its two open entries were closed in the course of this work.

Use of generative AI

This work used AI models extensively, under the author’s direction: Anthropic’s Claude for the analysis, code, encodings and drafting, and models of other families for read-only review. The author is responsible for all claims. Correctness rests on the DRAT proofs checked by `drat-trim`, the Lean development and the cited classical results, not on the models.

Changes from version 1.0

Version 1.0 was deposited on Zenodo on 7 October 2026 (DOI 10.5281/zenodo.23226734). This version (i) adds the statement on the use of generative AI above; (ii) states exactly what the Lean development proves ($\text{gap}(\oplus_3) \geq 3$, including $\text{tree}(\oplus_3) = 9$, with $\text{opt}(\oplus_3) \geq 6$ resting on the DRAT chain) and the encoding it uses, and updates the attribution note for $3(n-1)$ after reading Schnorr’s 1974 paper at source; (iii) answers the first open question of version 1.0: the XAG unit gap fails at $n = 5$ (§5), and the sentence calling an exhaustive $n = 5$ census infeasible is withdrawn; (iv) cites the mockturtle databases and the IWLS 2019 formula library, which version 1.0 omitted; and (v) reports that the $\text{opt} = 6$ column of Krinkin’s Table 1 records the §2 identity rather than the gap (§3). No result of version 1.0 about AIG changes.

Changes from version 2.0

Version 2.0 was deposited on Zenodo on 12 October 2026 (DOI 10.5281/zenodo.23327908). This version changes presentation only: the verification table of §6 now fits its page; the sentence on Krinkin’s Table 3 and the attribution footnote for $3(n-1)$ are reworded, the latter no longer discussing an uncited attribution; and a caveat repeated in the abstract and a sentence repeated in Appendix C are removed; Appendix C now cites this version’s deposit. No result changes.

References

- [1] Russ Cox. Minimal boolean formulas. <https://research.swtch.com/boolean>, 2011. Exhaustive enumeration; not peer-reviewed.
- [2] A. P. Goucher. optimal5. <https://gitlab.com/apgoucher/optimal5>, 2020. Commit b7668d7; the circuits were computed by D. E. Knuth with BOOLCHAINS.
- [3] Winston Haaswijk. *SAT-Based Exact Synthesis for Multi-Level Logic Networks*. PhD thesis, EPFL, 2019.
- [4] John P. Hayes. The fanout structure of switching functions. *Journal of the ACM*, 22(4):551–571, 1975.
- [5] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer, 2012. Theorem numbers as in the author’s online version of the book.
- [6] V. M. Khrapchenko. Complexity of the realization of a linear function in the class of Π -circuits. *Mathematical Notes*, 9(1):21–23, 1971.

- [7] Donald E. Knuth. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. Addison-Wesley, 2011. §7.1.2.
- [8] Arist Kojevnikov, Alexander S. Kulikov, and Grigory Yaroslavtsev. Finding efficient circuits using SAT-solvers. In *Theory and Applications of Satisfiability Testing (SAT)*, 2009.
- [9] Kirill Krinkin. A simple constructive bound on circuit size change under truth table perturbation, 2026. arXiv:2603.09379. Data: <https://github.com/krinkin/bounds>.
- [10] Kirill Krinkin. The Unit Gap: How Sharing Works in Boolean Circuits, 2026. arXiv:2603.08033v2; withdrawn.
- [11] Siang-Yun Lee, Jie-Hong R. Jiang, Alan Mishchenko, and Robert Brayton. Enumeration of minimum fanout-free circuit structures. In *International Workshop on Logic and Synthesis (IWLS)*, 2019. Formula library Lflib5 at <https://people.eecs.berkeley.edu/~alanmi/research/funenum/>.
- [12] Troy Lee. A new rank technique for formula size lower bounds. In *STACS 2007*, volume 4393 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2007.
- [13] mockturtle. <https://github.com/lils/mockturtle>. Per-class databases in `include/mockturtle/algorithms/node_resynthesis/`: `mig_npn.hpp` (since 2018) and `xag_npn.hpp` (complete AIG and XAG databases since pull request #455, 2021); checked at commit 47d1e70.
- [14] Hiroki Morizumi. Lower bounds for the size of nondeterministic circuits, 2015. arXiv:1504.06731.
- [15] Heinz Riener, Rüdiger Ehlers, Bruno Schmitt, and Giovanni De Micheli. Exact synthesis of ESOP forms. In *13th International Workshop on Boolean Problems*, 2018. arXiv:1807.11103.
- [16] K. L. Rychkov. On lower bounds of complexity of series-parallel contact networks realizing linear boolean functions. *Siberian Journal of Operations Research (Sibirsk. Zh. Issled. Oper.)*, 1(4):33–52, 1994.
- [17] C.-P. Schnorr. Zwei lineare untere Schranken für die Komplexität Boolescher Funktionen. *Computing*, 13(2):155–171, 1974.
- [18] Mathias Soeken, Luca G. Amarù, Pierre-Emmanuel Gaillardon, and Giovanni De Micheli. Optimizing majority-inverter graphs with functional hashing. In *Design, Automation & Test in Europe (DATE)*, 2016.
- [19] Mathias Soeken, Luca G. Amarù, Pierre-Emmanuel Gaillardon, and Giovanni De Micheli. Exact synthesis of majority-inverter graphs and its applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(11):1842–1855, 2017.
- [20] Mathias Soeken, Winston Haaswijk, Eleonora Testa, Alan Mishchenko, Luca G. Amarù, Robert K. Brayton, and Giovanni De Micheli. Practical exact synthesis. In *Design, Automation & Test in Europe (DATE)*, 2018.
- [21] Jun Tarui. Smallest formulas for the parity of 2^k variables are essentially unique. *Theoretical Computer Science*, 411(26–28):2623–2627, 2010.
- [22] kissat 4.0.4; drat-trim (marijnheule @ commit 2e3b2dc); Lean 4.31.0 (re-checked under 4.33.0), 2026. Solver, proof checker and proof assistant used in this work.

A Corrected proof of Theorem 3 (Threshold)

Claim. *If f has n essential variables and $\text{gap}(f) = 1$, then $\text{opt}(f) = m \geq n$.*

Proof. Since $\text{gap}(f) = 1$, take a size-optimal single-output AIG C for f . C is **not a tree**: if it were, $\text{tree}(f) \leq |C| = \text{opt}(f)$, forcing $\text{gap}(f) = 0$, contrary to hypothesis. So some non-output gate of C has fan-out ≥ 2 . Now count input incidences. Every gate has two inputs, so the total number of input slots is $2m$; partition them into I_{var} slots reading a primary input or the constant, and I_{gate} slots reading another gate, so $2m = I_{\text{var}} + I_{\text{gate}}$.

- *Essential variables:* each of the n essential variables must feed at least one slot (else f would not depend on it), so $I_{\text{var}} \geq n$.
- *Gate usage:* each of the $m - 1$ non-output gates feeds at least one later slot (normalized circuit: no unused gate), contributing $\geq m - 1$; the fan-out- ≥ 2 gate contributes at least one further incidence, so $I_{\text{gate}} \geq m$.

Hence $2m = I_{\text{var}} + I_{\text{gate}} \geq n + m$, i.e. $m \geq n$. □

(The bound holds for any non-tree optimal single-output circuit, subsuming the $\text{gap} = 1$ case. The published proof’s $|S| \geq k - 1$ step applies gate elimination to the cone S *excluding* the shared gate g ; when g is at input level the k essential variables of $S \cup \{g\}$ are not all covered by S , and the count is short by one. The incidence argument avoids the case split.)

B Encoders, gates, and the parity-5 false positive

The repository documents the AIG, XAG and multi-output encoders together with the validation gates each had to pass before use. For the XAG formula encoder these were: the complete $n = 3$ check ($\text{tree}_{\text{XAG}} = \text{opt}_{\text{XAG}}$ on all 256 functions), and `tree_via_sat` matching the independent DP on all 222 classes at $n = 4$, including the four with $\text{gap}_{\text{XAG}} = 1$.

The parity-5 false positive. The first $n = 5$ search implementation tested formula-satisfiability at exactly $\text{opt} + 1$ gates in isolation and declared a separator when that single instance was UNSAT. This is unsound for a normalized exact-size encoder: UNSAT at exactly $\text{opt} + 1$ means “no formula of *exactly* $\text{opt} + 1$ gates”, not $\text{tree} \geq \text{opt} + 2$ (a formula of a different size may exist). The bug surfaced on parity-5 (0x69969669), a linear function with $\text{gap}_{\text{XAG}} = 0$, which the isolated test misreported as a separator. The fix searches **ascending from** opt (tree is the first $k \geq \text{opt}$ with a satisfiable formula encoding); parity-5 then correctly returns $\text{gap} 0$. All results in §5 use the fixed procedure.

C Reproducibility

Every artifact named in this paragraph is archived on Zenodo under DOI 10.5281/zenodo.21630762, and is public at <https://github.com/totobusnello/unit-gap-artefacts>; the repository root carries a table mapping this paper’s principal claims to the files that back them. That DOI resolves to the latest deposited version, and the Zenodo record lists each version with its own DOI for readers who want the exact bytes a claim was checked against. For reproducibility, cite the version-specific DOI and retain the recorded file hashes. Re-checking that the archived CNFs are unsatisfiable requires `drat-trim` alone — not the encoders, not the solver, and nothing else in the repository. What a DRAT proof does *not* certify is that a CNF faithfully encodes the question asked of it. That

is a separate obligation, empirically checked here against independent exhaustive enumeration at the tested sizes (§6); interpreting the certificates as circuit lower bounds also relies on the encoding and minimality argument. The recorded transcript carries the SHA-256 of every CNF and DRAT file, so a reader can confirm they are checking the same bytes.

kissat 4.0.4; **drat-trim** (marijnheule @ commit 2e3b2dc); Lean 4.31.0 (re-checked under 4.33.0). Versioned artifacts: encoders, gates, censuses (AIG **npn4_gap.csv**, XAG **npn4_xag_gap.csv** with per-class **opt_xag/tree_xag/gap_xag**, MIG); CNFs and the $k = 1, \dots, 5$ DRAT proofs for $\text{opt}(\oplus_3)$ and $k = 1, \dots, 3$ for the representative child; the **drat-trim** transcript **certificates/verify_lowk.log** (s VERIFIED, exit code, and CNF/DRAT SHA-256 for all eight proofs); the $n = 5$ search CSVs (raw per-worker and globally deduplicated) and the 494/494 re-verification log; and the Lean sources. Both implementations of the tree recurrence are included — the layered dynamic program and the independent global Bellman fixed point — together with the transcript recording their agreement on all 65,536 cells. The rerun of the source catalogue’s own checker is under **verification/**; its two inputs and the checker itself are *not* redistributed (that repository carries no licence), and the SHA-256 needed to confirm them are. The $n = 5$ search seeds are deterministic per worker; the bench is a 2-vCPU VPS.

Added in version 2.0. The $n = 5$ XAG join of §5 reads the **Lf1ib5** text library as distributed in 2019 [11] and **optimal5** at commit b7668d7 [2]; neither library is redistributed, and the only material taken from either is the five six-gate circuits of the separators, as **optimal5** stores them (MIT licence, notice included). The joining scripts, the per-class table (616,126 rows: canonical truth table, essential variables, C , L , gap) with its SHA-256, the checks of the separator 0x00063357 and of the other four separator classes with $\text{opt}_{\text{XAG}} = 6$ (circuit simulation, the independent formula DP, and the SAT logs), and the Table 1 recount **check-krinkin-table1-n3-truegap.py** with its output are deposited on Zenodo with this version of the paper (DOI 10.5281/zenodo.23328820), not in the artefact record above.

Claim	Level of certification
$\text{opt}(\oplus_3) = 6$ (lower bound)	DRAT proofs for UNSAT at every $k = 1, \dots, 5$, drat-trim s VERIFIED ; transcript + hashes archived (Appendix C)
opt of the shared child = 4; other child NPN-equivalent	DRAT proofs for UNSAT at $k = 1, 2, 3$ on the representative child, drat-trim s VERIFIED ; the second child is NPN-equivalent
$\text{tree}(\oplus_3) = 9$; witnesses	Lean 4 kernel (decide) for witnesses + DP structural framework; the finite exclusion $\text{par3} \notin D_8$ (hence $\text{tree} \geq 9$) by native_decide (compiler-trusted); two implementations of the same recurrence agree
$\text{gap}(\oplus_3) \geq 2$	Analytic (Khrapchenko + six-gate circuit), independent of the tree DP and the DRAT chain; and fully citational via Jukna/Schnorr + Khrapchenko, given the basis identification of §2
$\text{gap}(\oplus_3) = 3$ (exact)	Certified here, not cited: the DRAT chain for $\text{opt} = 6$, and two independent implementations of the formula-size recurrence plus the Lean development for $\text{tree} = 9$. We found no determination of $\text{tree}(\oplus_3)$ in the refereed literature (§3); the value agrees with the exhaustive enumeration of [1]
$\text{gap}(\oplus_4) = 6$ (exact, census maximum)	Fully citational: the U_2 parity bound [5] for $\text{opt} = 9$, and for $\text{tree} = 15$ Khrapchenko’s n^2 bound, attained at $n = 4 = 2^2$ by the classical construction (the exact value is recorded by Tarui [21])
$\text{opt}(\oplus_3) = 6$, $\text{opt}(\oplus_4) = 9$ (external agreement)	Classical: the AIG basis is U_2 and the U_2 complexity of $\oplus_n$ is known exactly, $3n - 3$ [5] — an independent check of the encoder at two points
AIG gap census $n = 4$	Two implementations of the tree recurrence agree on all 65,536 cells; the opt side is the author’s companion catalogue [9], re-derived here class by class (222/222) — its UNSAT side is gated but not DRAT-certified (declared)
XAG census $n \leq 4$	Gated encoder (independent exhaustive enumeration at $n = 3$; formula encoder matches DP on all 222 $n = 4$ classes); SAT models simulation-verified; UNSAT side not proof-certified (declared)
XAG separator at $n = 5$ (0x00063357, $\text{gap}_{\text{XAG}} = 2$)	Circuit re-simulated; circuit encoder UNSAT at $k = 1, \dots, 5$ (no DRAT); $\text{tree}_{\text{XAG}} \geq 8$ by an independent formula DP and by the formula encoder (UNSAT at $k = 1, \dots, 7$, no DRAT); witnesses taken from the published tables are re-checked, and neither source’s optimality claims are used
XAG gap distribution $n = 5$ (616,126 classes)	Join of published data [11, 2], without certificate; each column reproduces its source’s published distribution, and the projection reproduces our $n = 4$ census 222/222
XAG search $n = 5$	Encoder gated at $n \leq 4$; 494 instances (all gap1 + a gap0 sample) re-decoded and simulation-verified; sample, not exhaustive (declared); agrees with the join on every decided function
MIG census $n = 4$	Same gating + exact match with an independently published size distribution (aggregate); per-class agreement with the mockturtle MIG database on 222/222 (upper bounds only)