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

Krinkin's *The Unit Gap: How Sharing Works in Boolean Circuits* (arXiv:2603.08033v2) asserts that for every Boolean function the minimum formula size exceeds the minimum circuit size by at most one gate — Theorem 2: gap $\in \{0, 1\}$ in the And-Inverter Graph (AIG) cost model — 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: $\mathrm{opt}(\oplus_3) = 6$ — UNSAT for every gate count $k = 1, \ldots, 5$, each certified by a DRAT proof checked with `drat-trim` — while $\mathrm{tree}(\oplus_3) = 9$, 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 Schnorr's theorem gives the circuit complexity of $\oplus_n$ as exactly $3n-3$, so $\mathrm{opt}(\oplus_3) = 6$; Khrapchenko forces $\mathrm{tree}(\oplus_3) \geq 9$ leaves, hence ≥ 8 gates (a formula with L leaves has $L-1$ gates). **So** $\mathrm{gap}(\oplus_3) \geq 2 > 1$ **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 $\mathrm{tree}(\oplus_3) = 9$, which is not in the refereed literature. That formula-side value is certified here by two independent dynamic programs and the Lean development, and agrees with an exhaustive enumeration published informally in 2011; the circuit side, opt $= 6$, rests on the DRAT chain above. At $n = 4$ the exact gap is classical again: Tarui determined the De Morgan formula size of $\oplus_n$ for $n = 2^k$ to be exactly n^2 leaves, giving $\mathrm{gap}(\oplus_4) = 15 - 9 = 6$ — the maximum of the census below. The error is locatable in the paper: §2 displays $\mathrm{tree}(f) = \min(1 + \mathrm{opt}(a) + \mathrm{opt}(b))$ (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 $\mathrm{opt}(\oplus_3)$ by 3). An exhaustive census over the 222 NPN classes of $n = 4$ shows the failure is structural: 72 of 222 classes have gap ≥ 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 XOR-AND-graph (XAG) basis the unit-gap property holds for $n \leq 4$ (under a separately gated exact-synthesis pipeline whose UNSAT side is not proof-certified): $\mathrm{gap}_{\mathrm{XAG}} = 0$ for all 256 functions of $n = 3$ and $\mathrm{gap}_{\mathrm{XAG}} \in \{0, 1\}$ for all 222 classes of $n = 4$; a directed search over 25,373 distinct essential functions of $n = 5$ **found no separator**: all 25,169 gap-decided functions have $\mathrm{gap}_{\mathrm{XAG}} \in \{0, 1\}$, and 204 remain unresolved. 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 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 $k = 1, \ldots, 5$ 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, holds in the XAG basis for $n \leq 4$ (exhaustive); a large directed search at $n = 5$ found no counterexample among the functions it decided. 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* $k = 1, \ldots, m - 1$ — 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 for the gap-by-basis census; the individual optimal-size tables in XAG and MIG are classical or previously published.

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* $\mathrm{tree}(f) = \min$ *over a, b: $f = a \wedge b$ or $f = \neg(a \wedge b)$ of* $(1 + \mathrm{opt}(a) + \mathrm{opt}(b))$*. The trivial decomposition $f = 1 \wedge f$ always achieves* $1 + \mathrm{opt}(f)$*.*

An **AIG** over $x_1, \ldots, x_n$ and the constant 1 is a DAG of two-input AND gates with free edge complementation; $\mathrm{opt}(f)$ 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 are for $n \leq 4$.

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 $\mathrm{opt}(f) = k$ a minimality lemma guarantees the k-encoding is satisfiable. Hence $\mathrm{opt}(f)$ is the least k with SAT, and a lower bound $\mathrm{opt}(f) \geq m$ requires UNSAT at *every* $k = 1, \ldots, m-1$. Restricting selection to fan-out one turns the same encoder into a $\mathrm{tree}(f)$ oracle (validated in Appendix B).

3 The refutation at $n = 3$

The paper gives incompatible characterizations of $\mathrm{tree}(f)$. The §2 identity puts the DAG measure opt in the children; with opt there, the trivial decomposition $f = 1 \wedge f$ yields $\mathrm{tree}(f) \leq 1 + \mathrm{opt}(f)$ 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):

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

with the **recursive** measure v in the children. At its fixed point this reads $\mathrm{tree}(f) = \min(1 + \mathrm{tree}(a) + \mathrm{tree}(b))$, a different object from the §2 identity: as global characterizations of formula size the two coincide term-by-term only on children for which opt = tree. Krinkin further states — as a *consequence of Theorem 2*, not a definition — that T's fixed point "overshoots $\mathrm{opt}(f)$ by exactly $\mathrm{gap}(f) \in \{0, 1\}$", which collapses together with Theorem 2.[1] For $\oplus_3$ the recursive T computes 9, overshooting opt = 6 by 3; under the §3 reading, Theorem 2 is false.

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

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

[1] Krinkin 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.)

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 opt ≥ 6 under the normalized exact-size encoding. *The value is also classical:* since the AIG basis is U_2 (§2), Schnorr's theorem — the U_2 circuit complexity of $\oplus_n$ is exactly $3n - 3$ [13], an equality restated in [10][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:

$$\neg(\,\neg P_1 \wedge \neg x_3\,) \wedge \neg(\,P_2 \wedge x_3\,), \qquad P_1 = P_2 = \neg(x_1 \oplus x_2), \tag{2}$$

  where each P_i is a private three-gate copy of the block that (1) computes once as g_3. 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 $1, \dots, 9$: 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 [4] then gives $L(\oplus_3) \geq |E|^2/(|A||B|) = 144/16 = 9$ leaves, hence $\text{tree}(\oplus_3) \geq 8$ gates. Together with the explicit six-gate circuit (opt ≤ 6), $\text{gap}(\oplus_3) \geq 2$ follows **using only Khrapchenko and that circuit** — independent of the tree DP and the DRAT chain. Substituting Schnorr's theorem for the explicit circuit removes even that: $\text{opt}(\oplus_3) = 6$ [13] 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 $n = 2^k$ Tarui [17] showed that Khrapchenko's n^2 is attained, so the formula size is exactly n^2 leaves; this fixes $\text{tree}(\oplus_4) = 16$ leaves, i.e. 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

[2]We cite the 1976 original for the result and a modern, openly accessible source for the statement of the equality; the original was not consulted directly.

[3]The 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.

[4]The obvious candidate does not survive. Lee's rank technique [9] 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: the error is in his Proposition 2, which assumes an optimal selection function takes each s_i to be a power of two, and he points to Rychkov [12] 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.

than cited: by the tree dynamic program, independently re-implemented, and by the Lean development. The value is not new — an exhaustive enumeration by Cox and Healy [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 $\mathrm{gap}(\oplus_3) = 3$, 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* $\mathrm{opt}(f) = 1 + \mathrm{opt}(a) + \mathrm{opt}(b) - s(a, b)$*, 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 opt $= 4$, $s = 1+4+4-6 = 3 \notin \{0, 1\}$. One child, $(x_1 \oplus x_2) \wedge \neg x_3$, has its 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 opt $= 4$. Krinkin's own $s \leq 1$ derivation is separately unsound: he needs $s \leq \mathrm{tree}(f) - \mathrm{opt}(f)$, but the §2 identity gives $\mathrm{tree}(f) \leq 1 + \mathrm{opt}(a) + \mathrm{opt}(b)$, i.e. $s \geq \mathrm{tree}(f) - \mathrm{opt}(f)$ — the reverse of the comparison the proof requires. (For parity itself $s = \mathrm{tree} - \mathrm{opt} = 3$, an equality, so Krinkin's first inequality happens to hold there; his chain then breaks at the *second* step, $\mathrm{tree} - \mathrm{opt} \leq 1$, which is Theorem 2.) Krinkin's Table 3 reports $s = 1$ for the complete opt-6 $n = 3$ bucket, which includes the parity decompositions — consistent with his having imposed the bound, and in conflict with the structural count $s = 3$.

Fate of the other results. Theorem 7 (a classification assuming gap $= 1$) loses its proof, which depends on Corollary 6; its conditional statement under the standard definition is not refuted by $\oplus_3$ (gap 3) and is left unproven. **The statements of Theorems 3 and 4 stand.** 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. In Krinkin's complete $n = 3$ table (Table 1), the two functions with gap $= 1$ at opt $= 6$ are parity and its complement — precisely what the §2 identity (with opt in the children: $1 + \mathrm{opt}(\text{parity}) + \mathrm{opt}(1) = 7$) reports, while the true formula size is 9. The error is visible in the paper's data.

Verification. The counterexample is formalized in Lean 4.31.0 without Mathlib (`UnitGap.lean`): the witnesses and the DP's structural framework are kernel-checked (`decide`, induction); the finite fact par3 $\notin D_8$ — hence $\mathrm{tree}(\oplus_3) \geq 9$ — 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 $\mathrm{tree}(f)$ 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 gap ≥ 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 ($\mathrm{opt}_{\mathrm{XAG}} = 6, 5, 3$ respectively; $\mathrm{opt}_{\mathrm{XAG}}$ is NPN-invariant, so distinct values force distinct classes). One of the three is parity-4 (`0x6996`, $\mathrm{opt}_{\mathrm{XAG}} = 3$). 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, MIG, and a directed search at $n = 5$

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 $\mathrm{opt}_{\mathrm{XAG}}$ maximum 7 (strictly below AIG in 190/222 classes; $\mathrm{opt}_{\mathrm{XAG}}(\oplus_4) = 3$) and $\mathrm{tree}_{\mathrm{XAG}}$ maximum 7. The gap census (`census/xag-n4/npn4_xag_gap.csv`): $n = 3$ — $\mathrm{gap}_{\mathrm{XAG}} = 0$ for all 256 functions; $n = 4$ — $\mathrm{gap}_{\mathrm{XAG}} \in \{0, 1\}$, 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, $n = 5$ (directed search). Exhaustive census is infeasible (2^{32} truth tables). We searched for a *separator* ($\mathrm{gap}_{\mathrm{XAG}} \geq 2$) by generating candidate functions from random XAG circuits with sharing and XOR bias, keeping those essential in all five variables, and deciding $\mathrm{opt}_{\mathrm{XAG}}$ and $\mathrm{tree}_{\mathrm{XAG}}$ 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 gap $= 1$); 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):

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

No separator was certified. All **25,169 gap-decided** functions have $\mathrm{gap}_{\mathrm{XAG}} \in \{0, 1\}$; 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 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 $n = 5$.

MIG (aggregate cross-check). Soeken, Amarù, Gaillardon and De Micheli [14] 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, not a per-class map; equal aggregate distributions cross-check the shared exact-synthesis machinery, not the per-class assignments.)

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

Cross-base summary ($n = 4$). ($\mathrm{tree}_{\mathrm{MIG}}$ requires a ternary DP whose naive form is cubic in set sizes; deferred.)

6 Verification

The two tree implementations are independent *implementations of the same Bellman recurrence* (layered vs. fixed-point) — strong bug detection; the semantic bridge (that the recurrence computes formula size) is the Lean completeness lemma. The four-model-family adversarial review is provenance, not part of the verification chain.

7 Related work and novelty

- **Krinkin, arXiv:2603.08033** [8] is the refuted paper; **arXiv:2603.09379** [7] (an NPN-4 AIG catalog whose two open entries 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 [14]; TCAD 2017 [15]); our census is replication, claimed only as an aggregate 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) [5]; our maximum of 7 agrees. We claim **no novelty for the XAG numbers themselves**.
- **Both sizes of parity are known, and neither is ours.** Schnorr [13] determined the U_2 circuit complexity of $\oplus_n$ exactly, $3n - 3$. On the formula side Khrapchenko's n^2 lower bound is classical and, for $n = 2^k$, attained [17]. Since the AIG basis with free inversions is U_2 (§2), that fixes $\mathrm{opt}(\oplus_3) = 6$ and $\mathrm{tree}(\oplus_3) \geq 8$ — already enough to refute Theorem 2 — and fixes

the $n = 4$ values exactly (opt = 9, tree = 15, gap 6, the census maximum). For tree($\oplus_3$) = 9 we found no determination in the refereed literature, but the value is not new either: the exhaustive enumeration of Cox and Healy [1] reports parity as 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. Khrapchenko's bound dates from 1971 and Schnorr's from 1976, so the facts that contradict Theorem 2 were on the record for fifty years 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. [16]; Haaswijk [2]), SAT-based circuit search [6], and ESOP exact synthesis [11] — tabulate optimal *circuits* (or, for ESOP, optimal formulas alone). We did not find a catalog reporting both measures for the same functions. The closest related line is the study of **fanout-free (read-once) functions** [3], 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 gap = 0 in AIG (opt = 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 $n \leq 4$, with a directed $n = 5$ XAG search. 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. We searched Knuth's B_2 data, the MIG/AIG and NPN-4 exact-synthesis catalogs, 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. Does gap_{XAG} remain in $\{0, 1\}$ for all of $n = 5$ (the directed search found no separator among 25,169 gap-decided functions) and for larger n? Parity no longer separates in XAG; other separators are not excluded.

2. Is there a theorem behind the empirical XAG unit gap at small n — e.g. a normal-form argument bounding sharing once linear structure is factored out?

3. An efficient exact DP for tree_{MIG} (the deferred MIG gap column).

4. A proof, or a counterexample, for the conditional statement of Krinkin's Theorem 7 under the standard definition. What $\oplus_3$ removes is the published proof, which routes through Corollary 6; the statement itself is left open.

Acknowledgements

I thank Kirill Krinkin. Told that Theorem 2 of [8] 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:2603.08033v3). His reading of the damage 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 but is not itself refuted, 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 [7] also supplied the $n = 4$ optima that the census rests on; its two open entries were closed in the course of this work.

Methods note

The analysis, code and certificates reported here were produced by an AI system operating under the direction of the author, who selected the target, set the verification standard and is responsible for all claims. Correctness does not rest on that process: the circuit lower bounds carrying the refutation — $\mathrm{opt}(\oplus_3) \geq 6$ and the shared-child bound — are DRAT proofs checked by `drat-trim` (the census UNSAT sides are gated but not proof-certified, as §6 declares), the formula-size counterexample is checked by the Lean 4 kernel except where `native_decide` is declared, and the citational route (Schnorr and Khrapchenko), which is what refutes Theorem 2, is independent of both. Before the results were communicated externally, the package was reviewed adversarially by four independent language-model families; that review is provenance, not verification.

References

[1] Russ Cox and Alex Healy. Minimal boolean formulas. `https://research.swtch.com/boolean`, 2011. Exhaustive enumeration; not peer-reviewed.

[2] Winston Haaswijk. *SAT-Based Exact Synthesis for Multi-Level Logic Networks*. PhD thesis, EPFL, 2019.

[3] John P. Hayes. The fanout structure of switching functions. *Journal of the ACM*, 22(4):551–571, 1975.

[4] V. M. Khrapchenko. Complexity of the realization of a linear function in the class of Π-circuits. *Mathematical Notes*, 9(1):21–23, 1971.

[5] Donald E. Knuth. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. Addison-Wesley, 2011. §7.1.2.

[6] Arist Kojevnikov, Alexander S. Kulikov, and Grigory Yaroslavtsev. Finding efficient circuits using SAT-solvers. In *Theory and Applications of Satisfiability Testing (SAT)*, 2009.

[7] Kirill Krinkin. NPN-4 AIG catalog, 2026.

[8] Kirill Krinkin. The Unit Gap: How Sharing Works in Boolean Circuits, 2026. v2.

[9] 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.

[10] Hiroki Morizumi. Lower bounds for the size of nondeterministic circuits, 2015.

[11] Heinz Riener, Rüdiger Ehlers, Bruno Schmitt, and Giovanni De Micheli. Exact synthesis of ESOP forms. In *International Workshop on Logic & Synthesis (IWLS)*, 2018. arXiv:1807.11103.

[12] K. L. Rychkov. On lower bounds of complexity of series-parallel contact networks realizing boolean functions. *Siberian Journal of Operations Research*, 1:33–52, 1994.

[13] C.-P. Schnorr. The combinational complexity of equivalence. *Theoretical Computer Science*, 1(4):289–295, 1976.

[14] 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.

[15] 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*, 2017. Extended version of [14].

[16] 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.

[17] Jun Tarui. Smallest formulas for the parity of 2^k variables are essentially unique. *Theoretical Computer Science*, 411(26–28):2623–2627, 2010.

[18] kissat 4.0.4; drat-trim (marijnheule @ commit 2e3b2dc); Lean 4.31.0, 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* $\mathrm{gap}(f) = 1$*, then* $\mathrm{opt}(f) = m \geq n$.

Proof. Since $\mathrm{gap}(f) = 1$, take a size-optimal single-output AIG C for f. C is **not a tree**: if it were, $\mathrm{tree}(f) \leq |C| = \mathrm{opt}(f)$, forcing $\mathrm{gap}(f) = 0$, 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 $2m = I_{\mathrm{var}} + I_{\mathrm{gate}}$.

- *Essential variables:* each of the n essential variables must feed at least one slot (else f would not depend on it), so $I_{\mathrm{var}} \geq n$.
- *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 $I_{\mathrm{gate}} \geq m$.

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

(The bound holds for any non-tree optimal single-output circuit, subsuming the 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 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 opt $+1$ means "no formula of *exactly* opt $+1$ gates", not tree $\geq$ 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 $k \geq$ opt with a satisfiable formula encoding); parity-5 then correctly returns gap 0. All results in §5 use the fixed procedure.

C Reproducibility

Every artifact named below 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. *Cite a DOI rather than the repository URL:* the DOI resolves to an immutable snapshot, and the URL does not. 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, discharged here by cross-checking the encoder against independent exhaustive enumeration (§6). 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. 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 $k = 1, \ldots, 5$ DRAT proofs for $\text{opt}(\oplus_3)$ and $k = 1, \ldots, 3$ 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.

Claim	Level of certification
$\mathrm{opt}(\oplus_3) = 6$ (lower bound)	DRAT proofs for UNSAT at **every** $k = 1, \ldots, 5$, `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
$\mathrm{tree}(\oplus_3) = 9$; witnesses	Lean 4 kernel (`decide`) for witnesses + DP structural framework; the finite exclusion par3 $\notin D_8$ (hence tree ≥ 9) by `native_decide` (compiler-trusted); two implementations of the same recurrence agree
$\mathrm{gap}(\oplus_3) \geq 2$	Analytic (Khrapchenko + six-gate circuit), independent of the tree DP and the DRAT chain; and fully citational via Schnorr + Khrapchenko
$\mathrm{gap}(\oplus_3) = 3$ (exact)	Certified here, not cited: the DRAT chain for opt = 6, and two independent dynamic programs plus the Lean development for tree = 9. No determination of $\mathrm{tree}(\oplus_3)$ appears in the refereed literature (§3); the value agrees with the exhaustive enumeration of [1]
$\mathrm{gap}(\oplus_4) = 6$ (exact, census maximum)	Fully citational: Schnorr [13] for opt = 9 and Tarui [17] for tree = 15, since $4 = 2^2$ is where Khrapchenko's bound is attained
$\mathrm{opt}(\oplus_3) = 6$, $\mathrm{opt}(\oplus_4) = 9$ (external agreement)	Classical: the AIG basis is U_2 and Schnorr's theorem gives the U_2 complexity of $\oplus_n$ as exactly $3n - 3$ [13] — 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 [7], 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 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)
MIG census $n = 4$	Same gating + exact match with an independently published size distribution (aggregate)