Paraconsistent Epistemic And Contextual Evaluation (PEACE)

Abstract

We present *Paraconsistent Epistemic And Contextual Evaluation* (PEACE), a meta-logical framework that nests on top of classical logic. It allows safe reasoning in the presence of paradoxes, contradictions, and uncertainty by integrating dialetheic paraconsistency, perspectivism, and contextualism. Classical reasoning is preserved as a special case inside stable contexts, while paradoxical or context-laden claims are identified as *category errors* when evaluated purely classically. PEACE provides an epistemically humble methodology for universal claim evaluation.

1 Base Logic: Meta-Dialetheic Core

1.1 Truth Values

The base truth value set is:

$$V = \{T, F, B\}$$

where:

- T: true only
- F: false only
- B: both true-content and false-content (*meta-dialetheic default*)

The designated values are:

$$\mathsf{D} = \{T, B\}$$

meaning that any value with true-content is considered "true enough" for consequence.

1.2 Content Semantics

Define content functions:

$$t(T) = 1, \quad t(F) = 0, \quad t(B) = 1$$
$$f(T) = 0, \quad f(F) = 1, \quad f(B) = 1$$

Connectives are defined compositionally:

Negation

$$t(\neg A) := f(A), \quad f(\neg A) := t(A)$$

Thus:

$$\neg T = F, \quad \neg F = T, \quad \neg B = B$$

Conjunction

$$t(A \wedge B) := t(A) \wedge t(B)$$
$$f(A \wedge B) := f(A) \vee f(B)$$

Disjunction

$$t(A \vee B) := t(A) \vee t(B)$$
$$f(A \vee B) := f(A) \wedge f(B)$$

Implication Defined as:

$$A \to B := \neg A \vee B$$

1.3 Semantic Consequence

We define:

$$\Gamma \vDash \varphi \quad \text{iff} \quad \forall v \, \big(\forall \gamma \in \Gamma, \; v(\gamma) \in \mathsf{D} \Rightarrow v(\varphi) \in \mathsf{D}\big)$$

This is paraconsistent: from A and $\neg A$ we cannot infer arbitrary B.

2 Perspectives

A *perspective* is a pair:

$$P = (I_P, \kappa_P)$$

where:

- I_P: an interpretation or normative constraint on vocabulary in φ
- κ_P: a verdict function κ_P : Formulas $\to \{T, F, B\}$

2.1 Admissibility and Load-Bearing

Admissibility (A): P is admissible for φ in context c iff:

1. *Relevance*: I_P constrains some symbol or speech-act in φ
2. *Coherence*: I_P preserves the content calculus

Load-Bearing (L): P is load-bearing for φ iff it changes $t(\cdot)$ or $f(\cdot)$ of at least one semantically decisive part of φ.

2.2 Conservativity Rule

If P is admissible but not load-bearing, then:

$$\kappa_P(\varphi) := \text{value}(\varphi)$$

This prevents "decorative" perspectives from altering truth values.

2.3 Neutral Policy

Prior to perspective application, all atoms default to B unless restricted by strong, context-independent evidence.

3 Contexts

A context c induces a relevance preorder $\preceq_c$ on admissible perspectives.

3.1 Influential Set

$$\mathcal{P}_c^*(\varphi) := \{P \mid P \text{ admissible and load-bearing for } \varphi\}$$

3.2 Contextual Verdict

Single-valued:

$$\text{Verdict}_c(\varphi) := \begin{cases} \kappa_{P^*}(\varphi) & \text{if } \mathcal{P}_c^*(\varphi) \neq \varnothing \text{ and } P^* \text{ is } \preceq_c\text{-maximal} \\ \text{value}(\varphi) & \text{otherwise} \end{cases}$$

Set-valued: Return the set of $\kappa_P(\varphi)$ for all $\preceq_c$-maximal P; optionally aggregate $\{T, F\}$ to B.

4 Entailment Variants

- **Designated entailment:** $\models$ as defined above.
- **Cautious entailment:** $\Gamma \models_T \varphi$ iff all models designate φ as T.
- **Contextual entailment:** $\Gamma \models_c \varphi$ iff φ is designated in all $\preceq_c$-maximal influential perspectives.

5 Category Error Detection

Define $CatErr(\varphi, c)$ to hold if any of:

1. Self-reference enabling diagonalization
2. Indexicals, vague predicates, figurative operators
3. Normative/pragmatic load (felicity conditions)
4. Strong conflicting evidence: $t(\varphi) = f(\varphi) = 1$

If $CatErr(\varphi, c)$, pure classical evaluation is a *category error*; switch to PEACE meta-evaluation.

6 Evidence Model

Each atom p has evidence scores $(e^+(p), e^-(p)) \in [0,1]^2$. Threshold rules map these to T, F, or B.

7 Theorems

- **Non-explosion**: There exist v with $v(A) = B$, $v(B) = F$ such that $A, \neg A \nvDash B$.
- **Neutral fixed point for Liar**: Any $S \leftrightarrow \neg True(S)$ yields $S = B$ under P_{meta}.
- **Conservativity**: If P is admissible but not load-bearing, $\kappa_P(\varphi) = \text{value}(\varphi)$.
- **Classical preservation**: On B-free fragments, $\vDash$ coincides with classical entailment.

8 Operational Recipe

Given claim φ and context c:

1. Check for $CatErr(\varphi, c)$. If yes, use PEACE.
2. Initialize atoms via evidence; else default B.
3. Compute neutral value value(φ).
4. Identify admissible load-bearing perspectives.
5. Rank via $\preceq_c$; return verdict(s).
6. Reason using chosen entailment variant.

9 Examples and Applications

In this section, we illustrate *Paraconsistent Epistemic And Contextual Evaluation* (PEACE) by applying it to well-known paradoxes and context-laden claims. Each example includes a formal PEACE analysis and a plain-language explanation so that readers without a formal logic background can still understand the reasoning.

We introduce the notion of *Context Completeness* $C_c(\varphi) \in [0, 1]$, measuring how fully a claim φ specifies the background needed for evaluation. The guiding aphorism is:

> **Aphorism:** The less complete the context, the harder it is to evaluate a claim correctly. When C_c is low, more perspectives are admissible and load-bearing, and the verdict set grows larger. A claim with very low C_c will tend to stabilize at the neutral value B in PEACE.

—

9.1 Example 1: The Liar Paradox

Claim: "This sentence is false." Let $L \equiv \neg True(L)$.

Neutral evaluation: In the base PEACE logic with transparent truth, the fixed point is $L = B$.

Context completeness: High ($C_c \approx 0.9$) — the paradox is fully specified; only its resolution method is in question.

Perspectives:

- P_{meta}: transparent semantics $\to B$. (both true and/or false)
- P_{prag}: assertional felicity norms $\to F$ (self-falsifying assertions cannot claim their own truth).
- P_{fig}: "is false" means "has any false-content" $\to T$.

Plain-language: Depending on how we *take* the statement, it can be both true and false at once, just false because it is a bad assertion about itself, or true because it correctly describes its own falsity.

Note: Answers to self-referential statements reference self-reference

—

9.2 Example 2: Russell's Paradox

Claim: "The set of all sets that do not contain themselves."

Neutral evaluation: Under naive comprehension, the self-membership test yields B.

Context completeness: Low ($C_c \approx 0.3$) — the claim omits its own foundational framework (e.g., ZFC vs. naive set theory).

Perspectives:

- P_{meta}: naive set semantics $\to B$.
- P_{formal}: ZFC set theory $\to F$ (no such set exists).
- P_{fig}: Sets that are in sets are subsets, trivially allowing $\to T$.

Plain-language: If we don't say what kind of sets we're talking about, we can't reliably settle the issue. In one mathematics context, the set "doesn't exist" (a paradox in itself); in another, it both exists and doesn't. As a metaphor, or indication of a categorical change it might make perfect sense.

—

9.3 Example 3: Sorites (Heap) Paradox

Claim: "A collection of n grains of sand is a heap."

Neutral evaluation: With a vague predicate "heap," intermediate n values yield B.

Context completeness: Very low ($C_c \approx 0.2$) — the claim does not define "heap" or threshold n.

Perspectives:

- P_{meta}: vague predicate semantics $\to B$ for mid-range n.
- P_{precise}: defines explicit threshold (e.g., $n \geq 1000$) $\to T$ or F depending on n.
- $P_{\text{figurative}}$: heap as metaphor (e.g., "a heap of trouble") $\to$ *variable*.

Sometimes we just treat paradoxes as jokes

Plain-language: Without saying exactly how many grains make a heap, we can't decide for sure. If you set a rule, you can decide — but different rules give different answers. In strict formal logic, this contextless claim is "a heap of trouble"!

—

9.4 Example 4: Barber Paradox

Claim: "The barber shaves all and only those men who do not shave themselves."

Neutral evaluation: Self-reference in membership to the shaved group yields B.

Context completeness: Moderate ($C_c \approx 0.5$) — missing whether barber can shave himself or is outside the group.

Perspectives:

- P_{meta}: formal semantics $\to B$.
- P_{legal}: resolves by rule (barber is exempt) $\to F$ or T depending on rule. I personally say a barber is a professional role, not a fixed identity. (Even a "barber" is shaving himself, he's just a man shaving himself, not a barber)
- P_{story}: fictional resolution where barber alternates days $\to$ *both*.

Aphorism *(from this paradox):* **My wife doesn't call me a philosopher when I philosophize; she calls me "annoying"**

Plain-language: As stated, the barber both shaves and does not shave himself. You need an extra rule to resolve it — without one, it's both true and false.

—

9.5 Example 5: Schrödinger's Cat

Claim: "The cat in the box is alive."

Neutral evaluation: In the quantum metaphor interpretation before observation, both alive and dead $\to B$.

Context completeness: High ($C_c \approx 0.8$) if physics model specified; low if not.

Perspectives:

- P_{quantum}: superposition $\rightarrow B$.
- $P_{\text{classical}}$: after observation, definite T or F.
- $P_{\text{figurative}}$: alive = "still relevant" $\rightarrow$ *context*.

Plain-language: Before looking, the cat is in a both and/or true/false state; after looking, it's trivially one or the other. In everyday talk, we don't think that way we know it's one or the other all along.

—

9.6 Example 6: Ship of Theseus

Claim: "The ship with all parts replaced over time is the same ship."

Neutral evaluation: Identity over time without clear criteria yields B.

Context completeness: Moderate ($C_c \approx 0.5$) — missing explicit identity conditions.

Perspectives:

- $P_{\text{classical\_identity}}$: strict part-identity $\rightarrow F$.
- $P_{\text{functional}}$: function continuity $\rightarrow T$.
- P_{cultural}: cultural narrative $\rightarrow T$ or B.

Plain-language: Whether it's "the same" depends on what you care about — the material, the function, or the story.

—

9.7 General Observations

In all cases, the PEACE framework:

1. Identifies missing or ambiguous context via low C_c (identifiable by if an answer can be easily identified or not by the answering party).
2. Generates a set of admissible perspectives.
3. Applies load-bearing tests to limit verdict changes.

4. Preserves classical reasoning inside any single perspective with T/F values only.

5. Flags classical-only evaluation as a category error when $CatErr(\varphi, c)$ holds.

9.8 Example 7: P vs NP as a Category Error

Claim: "For all decision problems in NP, there exists a polynomial-time algorithm that solves them" (P = NP).

Neutral evaluation in PEACE: In the classical formalism of computational complexity theory, NP is defined as the set of decision problems for which a given *certificate* can be *verified* in polynomial time by a deterministic Turing machine. This formalization *explicitly strips away all context* from the problem other than a bare string membership condition. It is a deliberate idealization: NP questions are encoded as total functions on formal strings; all semantic, pragmatic, or domain-specific structure is removed.

Context completeness: Extremely low for real-world NP problems once formalized in this way: $C_c \approx 0.1$. The act of reduction to the NP definition destroys much of the original context in which humans (or other algorithms) actually solve such problems.

PEACE diagnosis:

1. In real settings, many NP problems are *contextually grounded*: they have semantic constraints, domain-specific heuristics, and background knowledge that guide efficient solutions.

2. The NP definition forces all such problems into a *contextless classical shell*, treating them as arbitrary string sets.

3. From the PEACE standpoint, this is a *category error*: the classical NP definition evaluates a claim ("there exists a polytime solver") in a context stripped of exactly the information that makes such a solver possible.

4. This is directly analogous to Russell's Paradox: the definition creates a setting where the question cannot be answered in its original intended sense without importing back the lost context.

Perspectives:

- $P_{\text{meta\_formal}}$: Within the stripped-down formalism, P vs NP is a perfectly well-posed classical statement — but only about decontextualized, syntactic problems. Verdict: *Continued Ambiguity.*

- P_{applied}: Reintroduces real-world context for NP problems; notes that the stripped definition no longer matches the actual problem humans care about. Verdict: B (both possible and impossible, depending on whether context is reinstated).

- $P_{\text{philosophical}}$: Treats the question "Is P = NP?" for *real-world* problems as ill-formed under the formal NP definition. Verdict: F (false) in the sense of category error.

Plain-language: The way we define NP in computer science is a kind of trick: we throw away almost everything that makes a problem human-solvable in the real world, and keep only the bare string verification property. Then we ask whether those stripped problems can always be solved quickly. For many practical NP problems, the answer in the real world is "yes" *only because* of the context we threw away. So in PEACE terms, we've changed the question without realizing it — and the new question doesn't actually mean what we think it does. That's why "NP" in the pure formal sense and "NP" in the applied sense behave differently — and why P vs NP, as people *imagine* it, is a paradoxical category error.

Mathematical link: Let $\mathcal{P}_{\text{real}}$ be a real-world NP problem, with contextual information K that enables efficient solving:

$$\mathcal{P}_{\text{real}} = (I, Q, K)$$

where I is the instance space, Q is the decision predicate, and K is the context. The NP reduction map ρ strips K:

$$\rho(\mathcal{P}_{\text{real}}) = (I, Q)$$

In PEACE terms, $C_c(\rho(\mathcal{P}_{\text{real}})) \ll C_c(\mathcal{P}_{\text{real}})$. Verdict stability drops, admissible perspectives proliferate, and contradictions appear between the applied and formal readings of "efficiently solvable".

Theorem 9.1 (PEACE on Formal $P = NP$)**.** *Let $\mathcal{P}_{\text{real}}$ denote a real-world decision problem with context K. Let $\rho(\mathcal{P}_{\text{real}})$ be its formal NP encoding with K removed. Then:*

$$C_c(\rho(\mathcal{P}_{\text{real}})) \ll C_c(\mathcal{P}_{\text{real}})$$

and $\rho(\mathcal{P}_{\text{real}})$ is, in general, a different computational problem.

Therefore, the formal equation $P = NP$ in complexity theory does not model the real-world claim "Every efficiently verifiable problem can be efficiently solved." The formal $P = NP$ is thus a false abstraction *of the intended question. In PEACE, its truth in the formal model is irrelevant to, and generally false about, the real domain.*

Structural Analogy: Russell's Paradox Meets the Halting Problem

Russell's Paradox arises when one attempts to treat the "set of all sets that do not contain themselves" as if it were an ordinary set in naive comprehension. The definition destroys the necessary context (the type system or axiomatic framework) that would make the claim coherent, producing a self-referential object that cannot exist in the intended sense.

The formal P vs NP question makes an *identical category error in structure*. It takes the class of real-world verification problems — each embedded in a rich context K — and, by the NP definition, strips K away, leaving a contextless encoding $\rho(\mathcal{P}_{\text{real}})$. It then asks whether *these stripped objects* are all solvable in polynomial time, while continuing to speak as though it were asking about the original, context-rich problems.

However, because the P vs NP question is explicitly about *time-bounded* solvability, it also inherits the structural flavor of the *Halting Problem*. The Halting Problem demonstrates that there are questions about algorithmic behavior that cannot be decided in general, because they encode self-reference in the temporal behavior of computation. P vs NP does the same, but with the additional restriction of a polynomial-time bound, creating a *time-constrained self-referential problem*.

In both Russell's Paradox and the Halting Problem:

- The pathology arises from attempting to evaluate an object after removing or ignoring the structural context that ensures coherence.
- Self-reference is smuggled into the definition of the object itself.
- The resulting question cannot be answered in the original intended sense without reinstating the missing context.

PEACE Synthesis: Formal P vs NP is structurally a *hybrid* of Russell's Paradox and the Halting Problem:

- Like Russell's Paradox, it destroys decisive context (K), producing pathological abstractions.
- Like the Halting Problem, it encodes a self-referential decision about algorithmic behavior within a time-bound constraint.

Conclusion: In PEACE terms, the popular P vs NP is not a single coherent question at all. It is a *false abstraction* that combines the context-deletion pathology of Russell's Paradox with the time-bounded self-reference of the Halting Problem. As such, it is trivially false about the intended real-world computational phenomena, and any formal resolution of P vs NP applies only to its patently hypothetical contextless mathematical shell.

Goldbach Verification Cost: a Super-Linear Lower Bound (PEACE)

We consider the task: *verify that every even $n \leq N$ can be written as a sum of two primes.* We work in the standard RAM/bit model; "time" means bit operations.

Lemma 9.1 (Prime table cost)**.** *Any algorithm that verifies Goldbach up to N must, at minimum, identify primality up to N. Constructing a correct prime table up to N requires $\Omega(N)$ bit writes and $\Theta(N)$ space, and (with classical sieves) $O(N \log \log N)$ bit operations.*

Sketch. A correct verification must distinguish primes from composites up to N to witness $n = p + q$. Marking N bits (or words) is $\Omega(N)$ writes in any model that prevents writing multiple distinct cells in $o(1)$ time. Classical lower bounds for streaming/array initialization yield $\Omega(N)$ operations; upper bounds are achieved by the sieve of Eratosthenes in $O(N \log \log N)$ bit operations. □

Lemma 9.2 (Certificate size lower bound)**.** *Any algorithm that* explicitly certifies *Goldbach up to N by exhibiting a witness pair (p_n, q_n) for each even $n \leq N$ must output*

$$\Omega\Big(\sum_{n \leq N,\, 2 \mid n} \log n\Big) \;=\; \Omega(N \log N)$$

bits (hence $\Omega(N \log N)$ time).

Proof. Each witness (p_n, q_n) consists of integers $< n$, whose binary encodings require $\Theta(\log n)$ bits each; even if the algorithm outputs only one of them (the other being $n - p_n$) it still outputs $\Theta(\log n)$ bits. Summing over $O(N)$ even values gives $\Omega(N \log N)$ bits. Writing that many bits takes $\Omega(N \log N)$ time. □

Lemma 9.3 (Membership–query bound)**.** *Even without outputting witnesses, any algorithm that* decides *for every even $n \le N$ whether there exists $p \le n$ prime with $n - p$ prime, must perform $\Omega(N)$ distinct membership queries (and typically $\Omega(N/\log N)$), hence super-linear time once the bit-cost of indices/addresses is accounted for.*

Sketch. The algorithm must at least touch each even n (an $\Omega(N)$ step lower bound). Moreover, distinguishing primes from composites entails addressing/reading cells indexed up to N, which carries a per-access bit-cost $\Omega(\log N)$. Aggregated over $\Omega(N)$ necessary touches yields $\Omega(N \log N)$ bit operations unless one assumes unrealistic unit-cost random access to arbitrarily large addresses. □

Theorem 9.2 (Super-linear lower bound for Goldbach verification)**.** *Any correct finite verification of the strong Goldbach conjecture up to bound N in the RAM/bit model takes time*

$$\mathrm{Time}(N) \in \Omega(N) \quad \textit{and in fact} \quad \mathrm{Time}(N) \in \Omega(N \log N)$$

for any method that either (i) produces explicit witnesses or (ii) performs random-access membership over indices up to N with realistic bit costs. In particular, verification to infinite *bounds is infeasible, and raising N by constant factors induces more than linear growth in running time.*

Proof. Combine Lemma 9.1 with Lemma 9.2 (explicit certification) or Lemma 9.3 (implicit decision). Either route forces $\Omega(N)$ work, and under standard bit-cost accounting, $\Omega(N \log N)$. □

Remark (Upper bounds and why it "feels exponential"). Naïve checking yields $O\big(N^2/(\log N)^2\big)$ operations: for each n we scan $O(n/\log n)$ primes and do $O(1)$ membership checks (if a prime hash/set is available). Better methods (segmented sieves, cached prime sets, and convolution-based counting) reduce constants and, in some pipelines, the *average* work per n; nevertheless, the end-to-end cost grows strictly faster than linearly. Empirically, even mild super-linearity compounded across many doublings of N produces curves that appear nearly exponential on practical scales (memory traffic, cache misses, I/O, and parallel overheads exacerbate this effect).

PEACE interpretation. Finite verification to any N yields a *designated* verdict (T-evidence) but not a proof; the global claim remains at B in PEACE. The super-linear lower bound above explains why practical verification produces ever stronger evidence without exhausting the search space: increases in N incur more-than-linear cost, so exhaustive verification to infinity is infeasible in principle and in practice.

Appendix: Convolution-Based Verification and Its Cost

Let $\mathbf{1}_{\mathsf{P}}(n)$ be the prime-indicator (1 if n is prime, else 0), and define the aperiodic convolution

$$(\mathbf{1}_{\mathsf{P}} * \mathbf{1}_{\mathsf{P}})(n) \;=\; \sum_{k=0}^{n} \mathbf{1}_{\mathsf{P}}(k)\, \mathbf{1}_{\mathsf{P}}(n-k).$$

For even $n \geq 4$, this counts (ordered) Goldbach representations $n = p + q$.

Lemma 9.4 (Reduction to a single convolution)**.** *Fix $N \geq 4$ even. If one computes $c_n := (\mathbf{1}_{\mathsf{P}} * \mathbf{1}_{\mathsf{P}})(n)$ for all $0 \leq n \leq N$, then verifying the strong Goldbach conjecture up to N reduces to checking $c_n \geq 1$ for every even $n \in [4, N]$.*

Proof. By definition, c_n equals the number of pairs (p, q) of primes with $p + q = n$. Thus the claim for each n is equivalent to $c_n \geq 1$. (If desired, unordered pairs can be inferred by halving except when $p = q$.) □

To compute c_n efficiently, one uses fast convolution. Let M be a power of two with $M \geq 2N + 1$, and let $A[0..M-1]$ be the zero-padded prime-indicator array $A[i] = \mathbf{1}_{\mathsf{P}}(i)$ for $0 \leq i \leq N$ and $A[i] = 0$ otherwise. Then the aperiodic convolution equals the deconvolved circular convolution of A with itself.

Lemma 9.5 (Exact integer convolution via CRT/NTT)**.** *There is a word-RAM algorithm that computes all c_n ($0 \leq n \leq N$) exactly with*

$$\tilde{O}\big(N \log N \cdot \log N\big)$$

word operations, using a constant number of $\Theta(\log N)$-bit primes and the Number-Theoretic Transform (NTT) with Chinese Remainder Theorem (CRT) reconstruction.

Sketch. Choose L pairwise coprime NTT-friendly primes $p_1, \dots, p_L$ with product $P = \prod_{\ell=1}^{L} p_\ell$ exceeding the largest possible coefficient magnitude. For Goldbach counts we have the crude bound $c_n \le \pi(n) \le n$ for $n \le N$, and the sharper heuristic $c_n \approx n/(\log n)^2$, so $\log c_n = \Theta(\log N)$ bits. Hence $L = \Theta(1)$ if 64-bit moduli suffice for the target N, or in general $L = \Theta\big(\frac{\log N}{w}\big)$ for word size w.

For each p_ℓ, compute $C^{(\ell)} = \mathrm{NTT}_{p_\ell}(A) \odot \mathrm{NTT}_{p_\ell}(A)$ and invert NTT to get $c_n^{(\ell)} \equiv c_n \pmod{p_\ell}$. CRT on $\{c_n^{(\ell)}\}_{\ell=1}^{L}$ yields c_n exactly for all n. Each NTT costs $O(M \log M)$ modular operations; with $M = \Theta(N)$ and L moduli the total is $O(L\,N \log N)$. CRT reconstruction adds $\tilde{O}(L\,N)$. Taking $L = \Theta(\log N)$ in the worst case gives $\tilde{O}(N(\log N)^2)$ word ops. □

Remark (Floating-point FFT). Using complex FFT with rounding to nearest can be made rigorous by explicit error bounds and multiple-precision guards, but the needed precision grows like $\Theta(\log N)$ bits, so the bit-complexity likewise remains $\tilde{O}(N(\log N)^2)$ after accounting for precision.

Proposition 9.1 (End-to-end convolutional verification cost)**.** *An end-to-end pipeline that (i) sieves primes up to N, (ii) computes c_n exactly by Lemma 9.5, and (iii) checks $c_n \ge 1$ for even $n \le N$, runs in time*

$$\mathrm{Time}(N) \;=\; O\big(N \log\log N\big) \;+\; \tilde{O}\big(N(\log N)^2\big) \;+\; O(N) \;=\; \tilde{O}\big(N(\log N)^2\big),$$

thus strictly super-linear.

Proof. Sieving costs $O(N \log\log N)$ bit/word ops (Eratosthenes). Exact convolution is $\tilde{O}(N(\log N)^2)$ by Lemma 9.5. Scanning even n is $O(N)$. The convolution term dominates for large N. □

PEACE interpretation. Convolution reduces verification to a single auditable numerical task but does not change the asymptotics: the super-linear growth persists. Therefore, any finite bound N yields designated evidence (T-support) while the global statement remains at B in PEACE. The appendix formalizes why pushing N higher produces rapidly escalating costs that are exponential on practical scales, even though the theoretical bound is quasi-linear up to polylogarithmic factors.

Appendix: Implementation Notes for Large-N Goldbach Verification

This note outlines practical choices that make a convolution-based Goldbach verification pipeline auditable and fast at scale, without changing the $\tilde{O}\big(N(\log N)^2\big)$ asymptotic cost (Prop. 9.1).

(1) Prime generation: segmented sieve. Use a segmented sieve of Eratosthenes with block size tuned to LLC (last-level cache), e.g., $B \in [1, 8]$ MiB. Pre-sieve by small primes up to 64 (wheel $2 \cdot 3 \cdot 5 \cdot 7$) to cut memory traffic. Keep two structures: (i) a dense bitset $\mathsf{isPrime}[0..N]$ for convolution input; and (ii) an optional packed list of primes for spot checks. Time is $O(N \log\log N)$, and the memory footprint is $\approx N$ bits for the bitset plus $\tilde{O}(N/\log N)$ words if storing the prime list.

(2) Memory layout for the convolution array. Create an aligned, zero-padded array $A[0..M-1]$ with M the next power of two $\geq 2N+1$. Use an 8-bit or 16-bit element type for A (0/1 flags) to reduce bandwidth; promote to the modulus word width inside NTT kernels. Pin A and the NTT twiddle tables in huge pages (if available) to improve TLB locality.

(3) Exact convolution by CRT/NTT. Choose L pairwise coprime NTT primes $p_\ell = k \cdot 2^t + 1$ with t large enough to support length-M transforms. In practice, $L = 2$ or 3 64-bit primes suffice to cover coefficient growth (cf. Lemma 9.5). Precompute twiddle factors (ω, ω^{-1}) per prime; store in SoA (structure-of-arrays) format for SIMD. Perform forward NTT on A once per modulus, pointwise square, inverse NTT, then CRT-reconstruct c_n for $n \leq 2N$.

(4) CRT reconstruction and bounds. Use Garner's algorithm (mixed radix) or a precomputed CRT with floating-point reciprocal hints for speed. Upper bounds for c_n are $c_n \leq \pi(n) \leq \frac{n}{\log n}(1+o(1))$; ensure $\prod_\ell p_\ell > 2 \max_n c_n$ to avoid wrap. Validate reconstruction by spot-checking random n with a direct $O(\pi(n))$ test.

(5) Floating-point FFT (optional). If using complex FFT instead of NTT, set working precision to $\geq \lceil \log_2 N \rceil + \Delta$ bits to dominate rounding

error, where Δ covers accumulation and cancellation; round to nearest and verify that $|c_n - \mathrm{round}(c_n^*)| < \frac{1}{2}$ holds at random test points. For full rigor, repeat with two unrelated FFT sizes or mixed radix plans and compare.

(6) Parallelism. Exploit three levels: (i) data-parallel NTT butterflies (SIMD), (ii) modulus-level parallelism (run each p_ℓ on a separate thread), and (iii) segment-level parallelism for the sieve. Pin threads to cores to stabilize bandwidth; avoid over-subscription. Use work stealing for irregular tail segments.

(7) I/O and checkpoints. For very large N, checkpoint the sieve bitmap and intermediate $c_n^{(\ell)}$ arrays (mod p_ℓ) every fixed wall-time interval. Emit a manifest (JSON or YAML) with: N, M, primes p_ℓ, block size B, seed/wheel used, compiler flags, CPU model, wall-time, and commit hash. This makes runs reproducible and auditable.

(8) Verification strategy. After reconstructing c_n, scan only even $n \in [4, N]$ and test $c_n \geq 1$. On failure ($c_n = 0$), immediately recompute c_n by a slow direct method to rule out soft errors (e.g., memory, overclocking). On success, randomly sample a fixed fraction of even n and cross-check via direct counting using the prime list (or a second convolution with different moduli) to ensure end-to-end integrity.

(9) Memory footprint. Rough budget:

- Sieve bitset: N bits ($\approx 0.125\,N$ bytes).
- NTT arrays: L moduli $\times$ M elements $\times$ word size.
- Twiddles: $O(L\,M)$ words (reusable).

For $N = 10^{10}$, expect tens of gigabytes unless you stream segments and convolve in blocks; prefer multi-pass convolution with overlap-save/overlap-add to keep peak memory within budget at the cost of extra transforms.

(10) Auditable logging. Log: chosen parameters, timing per stage (sieve, forward NTT, pointwise square, inverse NTT, CRT, scan), maximum and average c_n, random check results, and a SHA-256 of the final even-n indicator

vector $\mathsf{ok}[4..N]$ (1 iff $c_n \geq 1$). Publish the manifest and logs with the binary & source.

(11) Failure handling and PEACE verdicts. Map outcomes to PEACE values:

- **F**: a verified counterexample $c_n = 0$ with independent re-check.
- **T** (designated evidence): all even $n \leq N$ pass; record N and artifacts.
- **B**: pipeline aborted/inconclusive (e.g., precision or memory failure).

This preserves epistemic humility: finite N never proves the global claim.

10 Conclusion

Manifesto — The PEACE Revolution. The Paraconsistent Epistemic And Contextual Evaluation (PEACE) framework reveals that many of the most famous open problems in mathematics — from Goldbach's Conjecture to P vs NP — are not merely "unsolved," but *misframed* within classical logic. By restoring context, perspective, and epistemic humility to truth evaluation, PEACE exposes that the traditional definition of "provable" is built on idealized abstractions which fail for these problems. In classical form, such problems demand answers from a logic that cannot validly evaluate them; exhaustive verification scales beyond all practical and theoretical reach, trapping them in a perpetual state of **B** ("both possible and impossible") without contradiction. Thus, their unprovability is not a tragic mystery, but an inevitable consequence of asking a flawed question in a flawed framework — and PEACE offers the first structured method to see, explain, and navigate that reality.

Afterword: PEACE in Plain Language

Some problems in math and logic — like whether every even number is the sum of two primes (Goldbach's Conjecture) or whether

P = NP — have been chased for decades without resolution. The usual tools of "true" and "false" just don't cut it, because they assume the question is perfectly framed and can be answered in an ideal, context-free world.

But in the real world, every statement comes with context, perspective, and uncertainty. **PEACE** — *Paraconsistent Epistemic And Contextual Evaluation* — adds a third state to logic: **B**, which means "both true and false depending on perspective." It treats **B** as the natural default whenever we don't have enough context to lock something down.

When you apply PEACE to big unsolved problems, you discover something wild: many of them aren't "hard" in the normal sense — they're *category errors*. They've been forced into a framework that can't actually handle them. That's why all the computing power in the world only gives stronger and stronger evidence without ever crossing the finish line.

With PEACE, we stop pretending that "true" or "false" is the only goal, and we start reasoning in a way that's honest about limits, context, and contradictions. It's logic that's built for paradox, complexity, and the messy reality we live in (whether that's in philosophy, mathematics, or deciding if the store's going to run out of milk before your shift ends).