V1.0.0

The Excluded Middle as a Quantifiable Computational Boundary: Translation Overhead, Model-Space Expansion, and Search Cost of Classical Reasoning

Rowan Brad Quni-Gudzinas

QNFO-ULA

ABSTRACT

Engineers who build proof assistants, ontology-based query engines, and LLM reasoning pipelines must choose whether their logic admits the law of the excluded middle (LEM), the principle that every proposition φ satisfies $\varphi \vee \neg\varphi$, yet no standard overhead budget exists for that choice. This paper makes the classical/constructive boundary quantitative. We model three costs of classical reasoning: the size overhead of the Gödel–Gentzen double-negation translation, the expansion of the counter-model search space when classical valuations are replaced by Kripke valuations, and the branching explosion when LEM instances are added to a proof search. For a representative propositional formula of size 9, the translation has size 22, an overhead factor of $22/9 \approx 2.44$; for $n = 10$ variables the constructive model space exceeds the classical one by a factor of $1.5^{10} \approx 57.7$ on a two-world Kripke chain and $2^{10} = 1024$ on a three-world chain; for a proof search of branching factor 3 and depth 12 with 5 atoms, LEM instantiation multiplies the search tree by $2^5 = 32$, from $531{,}441$ to $17{,}006{,}112$ nodes; and the fraction of Kripke valuations on a three-world chain satisfying all n LEM instances falls as 2^{-n}, to $1/1024$ at $n = 10$. As a contrast case we derive the workload of a boundary-element benchmark (96 elements, 64 N) whose computation never crosses the boundary. The result changes how the classical/constructive choice should be priced: as a computable overhead budget, not a philosophical stance.

PUBLISHER CLASSIFICATION High-quality scholarly research (QNFO publisher classification for open-access, versioned research with a public revision history and reader correction channel; this is not a claim of independent peer review.)

PEER-REVIEW STATUS Not independently peer-reviewed

CITE AS Quni-Gudzinas, R. B. (n.d.). The Excluded Middle as a Quantifiable Computational Boundary: Translation Overhead, Model-Space Expansion, and Search Cost of Classical Reasoning. QNFO Preprint v1.0.0.

1. Introduction

Engineers who build systems that reason — proof assistants, automated theorem provers, ontology-based data access platforms, and LLM-based reasoning pipelines — must decide, usually without being told, whether their underlying logic admits the law of the excluded middle. The decision is invisible in the specification documents and decisive in the behavior. A classical prover can close a goal $\exists x\, \varphi(x)$ by refuting $\neg\exists x\, \varphi(x)$ without ever producing an x; a constructive prover cannot, and must instead search for a witness. A classical database reasoner can use proof by contradiction inside an ontology's query rewriting; a constructive one must compile every inference into an explicit computation. When the wrong choice is made, the symptoms are concrete: proofs that certify existence without yielding artifacts, query engines whose compiled queries balloon in size, and reasoning pipelines that commit to answers before the reasoning that justifies them has run.

The cost of getting this choice wrong is engineering time and correctness, not philosophy. The analysis of the excluded middle as a computational boundary [8] argues that the classical notion of computational equivalence based solely on input–output behavior is impoverished, and that two systems are truly equivalent only if they solve the same problem class at the same resource cost. If that argument holds, then the classical/constructive choice is a resource decision, and it should be possible to attach numbers to it. Today no such numbers are standard: a team choosing between a classical

and a constructive kernel for a proof assistant, or between a classical and a rewrite-based ontology reasoner, has no overhead budget to consult. What would change if the result of this paper holds: the choice could be priced in advance, as a translation-size factor, a model-space expansion factor, and a search-branching factor, each computable from the shape of the formulas and the size of the problem.

This paper derives those numbers. The approach treats LEM as a boundary marker between two computational regimes and computes the cost of crossing it in three directions: the cost of importing a classical theorem into a constructive system (translation overhead), the cost of searching for constructive counterexamples (model-space expansion), and the cost of simulating classical reasoning inside a constructive proof search (LEM branching). A fourth measure captures the content that crossing destroys: the fraction of valuation space in which the LEM instances themselves hold constructively. The principal results, with their derivations, are:

- The Gödel–Gentzen translation of a representative propositional formula of size 9 has size 22, an overhead factor of $22/9 \approx 2.44$, derived in Section 4.1.
- The Kripke counter-model space with $n = 10$ propositional variables exceeds the classical valuation space by a factor of $1.5^{10} \approx 57.7$ on a two-world chain and $2^{10} = 1024$ on a three-world chain, derived in Section 4.2.
- Adding LEM instantiation for 5 atoms to a proof search of branching factor 3 and depth 12 multiplies the search tree by $2^5 = 32$, from $531{,}441$ to $17{,}006{,}112$ nodes, derived in Section 4.3.
- The fraction of Kripke valuations on a k-world chain in which all n LEM instances hold is 2^{-n}; for $n = 10$ and $k = 3$ this is $1/1024 \approx 0.0009765625$, derived in Section 4.4.
- A contrast-case workload anchor from numerical computation: the canonical boundary-element benchmark of [1] has 96 constant boundary elements and a total applied load of 64 N, derived with explicit arithmetic in Section 4.5.

We take the pricing framing as the paper's central convention, because every quantitative result below is a cost of crossing or holding the boundary, and no result below depends on the barrier framing; the boundary-element workload of [1] enters as the contrast case of computation that never crosses the boundary, under the convention that its correctness proof is not the object of study. The same boundary appears outside formal logic: measured results on structured prompting show that a reasoning framework scoring 100% in isolation degrades to 0–30% inside a production prompt [3], a pattern this paper reads as an empirical instance of premature classical commitment. The results could also matter to designers of ontology-based data access systems, where the classical reasoning underlying query rewriting must be compiled into executable form [2], and to type-theory library maintainers deciding whether to assume propositional extensionality, which determines whether choice forces LEM in their system [6].

2. Background and Related Work

The logical background begins with the study of the excluded middle as a computational boundary [8]. That work reconstructs the chain from Brouwer's intuitionism — which arose during the foundational crisis, against Hilbert's formalism and Russell's logicism, as a demand that mathematics be grounded in construction rather than symbol manipulation [8, §2.1] — to a critique of Turing completeness. Its central finding is stated as a separation: classical computers are not "flawed," but their underlying logical framework is "a specialized and limited model that fails to capture the full richness of what 'computation' can and perhaps should mean" [8, §7.1]. The same work proposes that computational equivalence must be multi-dimensional, requiring that two systems solve the same problem class "with the same class of resource costs (e.g., in polynomial time)" [8, §5.7]. That proposal is exactly the demand this paper operationalizes: if equivalence is resource-sensitive, the classical/constructive boundary should carry a measurable resource signature, and Sections 4.1–4.3 compute three components of it.

The companion framework on the universal computational topos [9] supplies the hierarchical vocabulary this paper uses. It distinguishes Object Computation (processes in their native domains), Model Computation (the representational layer that describes and simulates them), and Meta Computation, as three roles at distinct abstraction levels with precise interfaces [9, §2.1]. Its treatment of computational self-reference notes that Gödelian incompleteness forces any sufficiently powerful formal system to contain true-but-unprovable statements, which becomes acute when a system models its own processes [9, §1.0]. The LEM boundary studied here sits at the Model Computation layer: the choice of logic is a property of the representational layer, and its costs propagate downward into the object-level search procedures. The same framework's appendix catalogs fixed-point theorems by categorical context — Knaster–Tarski on complete partial orders, Kleene recursion on ω-cpos via $\mathrm{lfp}(f) = \bigvee_{n<\omega} f^n(\bot)$, and Banach contractions on complete metric spaces [9, App. B] — and least-fixed-point construction is the canonical example of a constructive existence proof: the iteration $f^n(\bot)$ is the witness-producing procedure that a purely classical existence argument would lack.

On the logic side, the semisimplicity paper [5] formulates a general, signature-independent form of LEM and proves that a logic is semisimple if and only if it enjoys this law, under a weak form of Raftery's inconsistency lemma [5, abstract]. Its Glivenko analysis identifies the semisimple companion of a logic L with the logic determined by the largest non-trivial theories of L, dual to the structural completion determined by the smallest theory [5, §1.4], and shows that a Glivenko-like connection obtains between a logic and its semisimple companion, subsuming several known Glivenko theorems under a single umbrella [5, §4.4]. This gives the present paper its precise logical handle: classical logic is the semisimple companion of intuitionistic logic, so crossing the LEM boundary is passing from a logic to its semisimple companion, and the Glivenko translation is the vehicle. The same work's deduction–detachment analysis — the classical equivalence $\Gamma, \varphi \vdash_{\mathrm{CL}} \psi \iff \Gamma \vdash_{\mathrm{CL}} \varphi \to \psi$, and its n-contractive generalization $\Gamma, \varphi \vdash_{\mathrm{FL}_{\mathrm{e}}^n} \psi \iff \Gamma \vdash_{\mathrm{FL}_{\mathrm{e}}^n} (1 \wedge \varphi)^n \to \psi$ [5, §3.1] — shows that even the basic move of reducing validity to theoremhood carries a formula-size price that depends on the logic, which is the same phenomenon our translation overhead measures.

The type-theoretic result of [6] sharpens the boundary from the other side. Diaconescu's theorem derives excluded middle from the axiom of choice using propositional extensionality, which holds in every topos; the Lean core library assumes propositional extensionality and uses it to derive excluded middle from global choice [6, §1]. But in dependent type theory without propositional extensionality, global choice alone does not imply excluded middle: [6] builds extensional Martin-Löf type theory with an impredicative universe Prop, extends it with a global choice operator, and constructs a model over $L = \mathbb{N}$ in which $\bot = \infty$ has no closed term, the model is consistent, and the negations of both propositional extensionality and excluded middle have closed terms [6, §4.4]. The staged-truth-value semantics — a truth value $p \in L_\infty$ holds from stage p onward, so 0 is true from the start and ∞ is never true [6, §2.1] — is itself a constructive semantics in which LEM fails by construction. For this paper, [6] shows the boundary is not monolithic: a system can keep choice, the principle that drives most computational construction, while refusing LEM, the principle that destroys witness content.

The applied side shows the boundary operating in deployed systems. The ontological query processing literature [2] studies compiling a query against an ontology on top of a relational database into a direct query against that database, with optimization to obtain "possibly small and efficient compiled queries" [2, §1]. The reasoning underlying the best-known decidable OWL fragments is "computationally very hard, typically 2-exptime-complete" [2, §Description Logics], and the OBDA program implements ABoxes as relational databases stored in classical RDBMSs [2, §OBDA]. The Nyaya prototype implements TGD-rewrite and query elimination over the IRIS Datalog engine, extended to support existential head variables and the constant *false* as a rule head representing negative constraints, evaluated on DL-Lite$_R$ ontologies with a varying number of axioms [2, §7]. This is a concrete instance of our translation overhead: the compilation from ontological (classically reasoned) queries to executable database queries is exactly a double-negation-style content-preserving translation, and its size growth is the practical face of the factor computed in Section 4.1.

The numerical computation literature supplies the contrast case: computation that never crosses the boundary. The BEM study [1] investigates realtime simulation of biological organs under linear elastostatic behavior with constant boundary elements, using a $4 \times 4 \times 4$ mm cube benchmark with Young's modulus 200000 N/mm^2 and Poisson's ratio 0.33, discretized at sixteen boundary elements per face [1, §3]; it reports a GPU acceleration path toward realtime performance and, notably, that the author "has not been successful in achieving this goal using MATLAB and the Parallel Computing Toolbox" for the full realtime pipeline [1, §4]. This is Object Computation in the sense of [9]: dense linear algebra with no logical commitment either way, where LEM is irrelevant because every existence claim is witnessed by an explicitly computed displacement field. Similarly, the algorithmic differentiation study [7] differentiates a parallel CFD solver (JAGUAR) in tangent and adjoint modes, with the adjoint obtained by letting the AD tool Tapenade invert the complete MPI communications layer [7, abstract]; the adjoint code executes in 15 times the primal execution time, with the overhead largely due to binomial checkpointing trading storage for time across a temporal loop of order 10^6 iterations [7, §6]. Reverse-mode differentiation is the engineering realization of constructive content extraction — the adjoint code is a witness-producing transformation of the primal computation — and its measured $15\times$ overhead is an empirical data point of the same kind as our translation factor: content extraction has a price, and the price is measurable.

Finally, the solar physics partitioning study [4] offers a methodological analogy the paper uses explicitly. It defines the middle corona as the region roughly spanning heliocentric altitudes from 1.5 to $6\,R_\odot$, encompassing "almost all of the influential physical transitions" governing coronal outflow, including the change from closed to open magnetic field structures and from low to high plasma β [4, §7]. The region is poorly understood relative to the inner and outer corona "primarily because it has been much more poorly observed" [4, §7], and the partitioning of the solar atmosphere is explicitly based on which physical processes dominate energy transport in each region [4, §2]. The analogy: a boundary drawn between regimes is useful exactly when the transitions across it are enumerable and each side has a characteristic behavior. This paper draws the LEM boundary the same way — classical and constructive regimes, with a list of transitions (translation, model expansion, branching, content loss) that occur when crossing — and then does what [4] calls for in solar physics: characterizes the poorly observed middle region with explicit measurements, here computational ones rather than remote-sensing ones.

The prompt-engineering study [3] provides the measured behavioral parallel. STAR reasoning (Situation, Task, Action, Result) raised car wash problem accuracy from 0% to 85% on Claude Sonnet 4.5 and to 100% with additional prompt layers in a 10-line prompt [3, §1]; embedded in InterviewMate's 60-line production prompt, the same framework degrades to 0–30% ($A_{\text{star_anthropic}}$: 0% $(0/20)$; $B_{\text{star_default}}$: 30% $(6/20)$; versus $C_{\text{star_only}}$: 100% $(20/20)$) [3, §6.1], confirmed at $n = 100$ [3, §5.2]. The identified mechanism is conclusion-first ordering: accumulated instructions force the model to commit to an answer before the structured reasoning surfaces the implicit constraint, so the model outputs "Short answer: Walk." and then corrects itself below with "Drive your car to the wash" — but the committed answer is what the user sees first [3, §7.1]. The study's own conclusion is that "a technique that works in a clean room may fail on the factory floor" and that evaluation must include production-representative complexity [3, §10.3, §12]. In the vocabulary of this paper, the production prompt plays the role of a classical commitment principle: it forces early commitment (an answer exists and is already given) before constructive evidence is available, and the measured degradation from 100% to 0–30% is an empirical instance of the cost of crossing the commitment boundary in the wrong direction.

3. Methods

This paper's method is analytical modeling plus explicit arithmetic; no experiment, measurement, or software run was performed, and every number in Sections 4 and 5 is computed by hand from stated inputs.

Model 1: translation size overhead. We define the size $|A|$ of a propositional formula over atoms $p_1, \ldots, p_n$ with connectives $\wedge, \vee, \to, \neg$ by: $|p_i| = 1$; $|A \circ B| = |A| + |B| + 1$ for $\circ \in \{\wedge, \vee, \to\}$; $|\neg A| = |A| + 1$. The Gödel–Gentzen (negative) translation A^G is:

$$p^G = \neg\neg p, \qquad (A \wedge B)^G = A^G \wedge B^G, \qquad (A \to B)^G = A^G \to B^G,$$

$$(A \vee B)^G = \neg(\neg A^G \wedge \neg B^G), \qquad (\neg A)^G = \neg A^G.$$

This translation is the standard vehicle of the Glivenko correspondence: a formula A is a classical theorem if and only if A^G is an intuitionistic theorem, which is the single-umbrella Glivenko connection of [5, §4.4] specialized to propositional logic. We assume the translation is applied to formulas in negation-normal-free form as written (no prior simplification), which is the convention that maximizes transparency of the size accounting; under a convention that first simplifies double negations, the overhead would be smaller, and we state this choice here because it is a convention, not a fact.

Model 2: model-space expansion. For n propositional variables, a classical valuation assigns each variable one of 2 truth values, giving 2^n valuations. A Kripke valuation on a chain frame of k worlds (with $W = \{w_1 < w_2 < \cdots < w_k\}$ and persistence forcing upward-closed valuation sets) assigns each variable an upward-closed subset of the chain; every upward-closed subset of a k-chain is determined by its least world of truth or by absence of truth, giving $k + 1$ choices per variable and $(k + 1)^n$ valuations. The expansion factor is $\left(\frac{k+1}{2}\right)^n$. We compute both $k = 2$ (the smallest chain on which persistence has any effect) and $k = 3$ (the smallest frame that distinguishes classical from intuitionistic behavior for nested implications) as the working cases.

Model 3: LEM branching. We model a backtracking proof search whose base search tree has branching factor b and depth d, hence b^d leaf paths. Adding LEM as an available axiom schema $\varphi \vee \neg\varphi$ for each of n distinct atoms gives the prover an independent binary choice per atom — instantiate LEM on that atom, or not — at any point where the atom occurs; we assume, conservatively for the model, that each atom contributes exactly one independent binary choice along each path, so the tree multiplies by 2^n. This is a lower-bound model in the sense that it counts only the instantiate-or-not choice and not the deeper re-search each instantiation triggers; we state that assumption here.

Model 4: content gap. We measure content loss by the Glivenko image: a classical theorem φ survives constructively only as φ^G (or $\neg\neg\varphi$ in the propositional Glivenko case). We compute, for the LEM instances themselves, the exact fraction of constructive valuation space in which the instances hold, and express the gap as the complement fraction.

Sources for inputs. All logical definitions (translation rules, Kripke semantics, LEM schema) are standard and are stated in full above; the Glivenko correspondence and the semisimple-companion characterization of LEM are from [5]; the staged-truth-value model in which LEM fails constructively is from [6]; the resource-sensitivity demand that motivates pricing the boundary is from [8]; the layered Object/Model/Meta vocabulary is from [9]. The boundary-element inputs of Section 4.5 (cube side, traction, elements per face, elastic constants) are from [1]. No numerical input in Section 4 comes from a measurement performed for this paper; all logical counts are combinatorial computations from the stated definitions, and the BEM counts are arithmetic from the parameters of [1].

4. Analysis

4.1 Translation size overhead

Worked formula. Take $F = ((p \vee q) \to r) \wedge (p \to q)$ over atoms p, q, r.

Original sizes, step by step:

$$|p| = |q| = |r| = 1.$$

$$|p \vee q| = |p| + |q| + 1 = 1 + 1 + 1 = 3.$$

$$|(p \vee q) \to r| = 3 + 1 + 1 = 5.$$

$$|p \to q| = 1 + 1 + 1 = 3.$$

$$|F| = 5 + 3 + 1 = 9.$$

Translated sizes. Atoms double-negate:

$$|p^G| = |\neg\neg p| = 1 + 1 + 1 = 3, \quad |q^G| = 3, \quad |r^G| = 3.$$

The disjunction clause:

$$\neg p^G = \neg\neg\neg p, \quad |\neg p^G| = 3 + 1 = 4; \quad |\neg q^G| = 4.$$

$$|\neg p^G \wedge \neg q^G| = 4 + 4 + 1 = 9.$$

$$|(p \vee q)^G| = |\neg(\neg p^G \wedge \neg q^G)| = 9 + 1 = 10.$$

The implication into r:

$$|(p \vee q)^G \to r^G| = 10 + 3 + 1 = 14.$$

The second conjunct is unchanged in structure:

$$|(p \to q)^G| = |p^G \to q^G| = 3 + 3 + 1 = 7.$$

The conjunction:

$$|F^G| = 14 + 7 + 1 = 22.$$

Overhead factor:

$$\frac{|F^G|}{|F|} = \frac{22}{9} \approx 2.4444444.$$

A tempting additive shortcut — each atom contributes $+2$ (the double negation) and each disjunction $+2$, giving $|F^G| = |F| + 2a + 2v = 9 + 6 + 2 = 17$ for $a = 3$ atoms and $v = 1$ disjunction — disagrees with the direct count of 22. The discrepancy is real and instructive: the additive rule undercounts because the $+2$ per atom and per disjunction is applied to subformulas that are themselves embedded inside negations, and each negation layer adds $+1$ to everything beneath it. The direct count is therefore the reliable one, and we report the overhead as the directly computed ratio $22/9 \approx 2.44$ for this formula, noting that the growth is super-additive in negation depth. For the same formula written with the disjunction replaced by a conjunction, $F' = ((p \wedge q) \to r) \wedge (p \to q)$: $|p \wedge q| = 3$, $|(p \wedge q) \to r| = 5$, $|F'| = 9$; $|(p \wedge q)^G| = 3 + 3 + 1 = 7$; $|(p \wedge q)^G \to r^G| = 7 + 3 + 1 = 11$; $|F'^G| = 11 + 7 + 1 = 19$; overhead $19/9 \approx 2.1111111$. Disjunction is the expensive connective, as expected, since it is the one whose classical meaning LEM underwrites.

4.2 Model-space expansion

Classical valuations for n variables: each variable independently true or false, so 2^n. For $n = 10$: $2^{10} = 1024$.

Kripke valuations on a chain of k worlds: an upward-closed subset of $\{w_1 < \cdots < w_k\}$ is one of $\emptyset, \{w_k\}, \ldots, \{w_1, \ldots, w_k\}$ — exactly $k + 1$ choices per variable, so $(k + 1)^n$ valuations. Two computed cases:

- **Two-world chain ($k = 2$):** $3^{10} = 59049$ versus $2^{10} = 1024$; factor $3^{10}/2^{10} = 1.5^{10}$. Step check: $1.5^2 = 2.25$; $1.5^4 = 2.25^2 = 5.0625$; $1.5^8 = 5.0625^2 = 25.62890625$; $1.5^{10} = 25.62890625 \times 2.25 = 57.6650390625$.
- **Three-world chain ($k = 3$):** $4^{10} = 1048576$ versus 1024; factor $4^{10}/2^{10} = 2^{10} = 1024$.

The expansion is exponential in n with base $(k+1)/2$: constructive counter-model search is intrinsically more expensive than classical truth-table search, and the gap widens with both the number of variables and the depth of the Kripke frame. This is the price side of the boundary: the constructive semantics is richer (it distinguishes more situations), and richness is paid for in search space.

4.3 LEM branching in proof search

Base search tree: branching factor $b = 3$, depth $d = 12$:

$$3^{12} = 531441 \quad (3^6 = 729;\ 3^{12} = 729^2 = 531441).$$

Adding LEM instantiation choices for $n = 5$ atoms multiplies the path count by $2^5 = 32$:

$$531441 \times 32 = 17006112.$$

Branching factor:

$$\frac{17006112}{531441} = 32.$$

Scaling: the multiplier is 2^n, so the LEM-induced cost grows exponentially in the number of atoms the search touches, independently of the base tree shape. At $n = 10$ atoms the same base tree would grow by $2^{10} = 1024$, to $531441 \times 1024 = 544195584$ nodes; the model therefore predicts that LEM-heavy proof searches become intractable in the atoms long before depth becomes the binding constraint. Because the model counts only the instantiate-or-not choice and not the re-search each instantiation triggers, 2^n is a lower bound on the true multiplier under the stated assumptions.

4.4 Content gap: the fraction of constructive valuation space in which LEM holds

Derivation of the holding condition. Fix a chain frame $W = \{w_1 < w_2 < \cdots < w_k\}$ and a single variable p with an upward-closed valuation, i.e. one of the $k+1$ subsets $\emptyset, \{w_k\}, \ldots, \{w_1, \ldots, w_k\}$. Write w_0 for the least world at which p is forced (w_0 undefined if the valuation is $\emptyset$). At a world w:

- $w \Vdash p$ iff $w_0 \leq w$ (persistence: once forced, forced upward).
- $w \Vdash \neg p$ iff no $w' \geq w$ forces p, i.e. iff $w_0 > w$ or w_0 undefined.

The LEM instance $p \vee \neg p$ holds at w iff $w \Vdash p$ or $w \Vdash \neg p$, i.e. iff $w_0 \leq w$ or $w_0 > w$ or w_0 undefined. If w_0 exists, this holds at every $w \geq w_0$ and fails at every $w < w_0$ (at such a w, p is not forced and $\neg p$ is not forced because $w_0 \geq w$ forces p). Hence $p \vee \neg p$ holds at **all** worlds of the chain exactly in two cases: the valuation is $\emptyset$ (p never true, so $\neg p$ holds everywhere), or $w_0 = w_1$ (p true from the bottom, so p holds everywhere). That is 2 of the $k+1$ upward-closed subsets per variable.

Fraction. For n independent variables, the fraction of Kripke valuations on the k-chain in which all n LEM instances hold at all worlds is

$$\left(\frac{2}{k+1}\right)^n.$$

For $k = 3$ this is $\left(\frac{2}{4}\right)^n = 2^{-n}$; at $n = 10$:

$$2^{-10} = \frac{1}{1024} = 0.0009765625.$$

For $k = 2$ the fraction is $\left(\frac{2}{3}\right)^{10} = \frac{2^{10}}{3^{10}} = \frac{1024}{59049} \approx 0.0173415$. The gap — the fraction of constructive valuation space in which the LEM instances fail somewhere — is therefore $1 - 2^{-10} = \frac{1023}{1024} \approx 0.9990234375$ on the three-world chain and $1 - \frac{1024}{59049} = \frac{58025}{59049} \approx 0.9826585$ on the two-world chain. The content gap and the model-space expansion of Section 4.2 are two readings of the same count: the $(k+1)^n - 2^n$ valuations excluded from the classical space are exactly the ones in which some LEM instance fails, which is why the constructive semantics is simultaneously richer (more counter-models) and thinner (fewer valuations validate classical tautologies).

4.5 Contrast case: a workload that never crosses the boundary

The boundary-element benchmark of [1] supplies the anchor for computation on the classical side of no boundary at all — dense linear algebra in which every existence claim is witnessed by an explicitly computed field. The inputs, from [1, §3]: a $4 \times 4 \times 4$ mm cube, one face completely fixed, traction of $4\ \mathrm{N/mm^2}$ applied in the y-direction over the opposite face, each face discretized into sixteen boundary elements, Young's modulus $200000\ \mathrm{N/mm^2}$, Poisson's ratio 0.33.

Element count. Six faces at sixteen boundary elements per face:

$$6 \times 16 = 96 \text{ constant boundary elements.}$$

Total applied load. The loaded face has area $4\ \mathrm{mm} \times 4\ \mathrm{mm} = 16\ \mathrm{mm^2}$; at a uniform traction of $4\ \mathrm{N/mm^2}$:

$$16\ \mathrm{mm^2} \times 4\ \mathrm{N/mm^2} = 64\ \mathrm{N}.$$

System size. Each of the 96 elements carries displacement and traction unknowns; the characteristic system the BEM assembles and solves is of order 10^2 unknowns, and [1] reports that the goal of computing the characteristic matrix, the right-hand side, and solving the system in realtime was pursued on a GPU, with the author noting that the realtime goal was "not been successful in achieving this goal using MATLAB and the Parallel Computing Toolbox" [1, §4]. The relevance here is categorical, not numerical: nothing in this workload depends on whether the underlying logic admits LEM, because every quantity — displacements, tractions, the solution of the linear system — is produced by an explicit algorithm whose existence claims are constructive by construction. The 96-element, 64 N workload is therefore the zero-crossing baseline against which the three crossing costs of Sections 4.1–4.3 are read: translation overhead ≈ 2.44, model-space expansion up to $1024\times$, LEM branching $32\times$, all incurred only when classical reasoning is imported into, or simulated within, a constructive setting.

5. Results

All numbers below are computed in Section 4 by explicit arithmetic from the stated definitions and inputs; none is measured or simulated.

Result	Value	Where derived
Translation size overhead, $F = ((p \vee q) \to r) \wedge (p \to q)$	$\lvert F^G\rvert/\lvert F\rvert = 22/9 \approx 2.4444444$	§4.1
Same formula with disjunction replaced by conjunction	$19/9 \approx 2.1111111$	§4.1
Model-space expansion, $n = 10$, two-world chain	$1.5^{10} = 57.6650390625$	§4.2
Model-space expansion, $n = 10$, three-world chain	$2^{10} = 1024$	§4.2
LEM branching, $b = 3$, $d = 12$, $n = 5$ atoms	$531441 \to 17006112$ nodes, factor 32	§4.3
LEM branching, same base tree, $n = 10$ atoms	factor 1024, $531441 \times 1024 = 544195584$ nodes	§4.3
Content gap, $k = 3$, $n = 10$: fraction where all LEM instances hold	$2^{-10} = 1/1024 = 0.0009765625$	§4.4
Content gap, $k = 2$, $n = 10$	$(2/3)^{10} = 1024/59049 \approx 0.0173415$	§4.4
BEM benchmark elements and load [1]	96 elements; 64 N	§4.5

Three regularities hold across the table. First, every crossing cost is exponential in a structural parameter of the problem: translation overhead grows with negation depth (super-additively, per the discrepancy between the direct count 22 and the additive shortcut 17 in §4.1), model expansion grows as $\left(\frac{k+1}{2}\right)^n$, and LEM branching grows as 2^n. Second, the disjunction is the expensive connective ($22/9$ versus $19/9$), consistent with LEM being the principle that underwrites classical disjunction's meaning. Third, the content gap closes exponentially: the constructive share of valuation space in which classical tautologies hold falls as 2^{-n} on the three-world chain, so at $n = 10$ only 0.09765625% of the constructive model space validates all ten LEM instances.

6. Discussion

What the numbers do and do not establish. The paper establishes that three specific costs of the classical/constructive boundary are computable from problem shape, and gives their values for representative parameter choices. It does not establish that these costs are the dominant costs in any deployed system, nor that they are tight: the LEM branching model of §4.3 counts only the instantiate-or-not choice and is explicitly a lower bound on the true multiplier, and the translation overhead of §4.1 is computed for one formula under a no-prior-simplification convention.

Limitations, with sensitivity.

- *Translation convention.* The factor $22/9 \approx 2.44$ assumes the translation is applied without simplifying double negations. Under a convention that cancels $\neg\neg$ at atoms, the atom clause $p^G = \neg\neg p$ would collapse to p, and the overhead would drop toward the disjunction-only surcharge; the sensitivity is direct, since 6 of the 13 size units of overhead in F^G come from atom double negations. The reported factor is therefore an upper-bound-style figure for this formula under the stated convention.
- *Single-formula scope.* One formula (and its conjunction variant) is counted. The overhead factor is not claimed to be a constant across formulas; it depends on the disjunction count and negation depth, as the 22 versus 17 discrepancy shows.
- *Kripke chain restriction.* Model-space expansion is computed on chains, the simplest frames with persistence. Branching or partially ordered frames admit more upward-closed subsets per variable than $k + 1$, so $(k + 1)^n$ is a lower bound on the valuation count for general frames; the expansion factors 57.7 and 1024 are correspondingly conservative for richer frames.
- *Branching model.* The 2^n LEM multiplier assumes one independent binary choice per atom per path and ignores re-search triggered by each instantiation; the true multiplier is larger, so 17006112 nodes is a floor under the stated

assumptions.

- *Contrast case scope.* The BEM anchor of §4.5 is a single benchmark from [1]; it grounds the claim that some computation never crosses the boundary, not a claim about numerical computation generally.

Falsification. The main result — that the boundary carries computable, exponential-in-structure overheads — would be falsified by any of the following observations: a family of formulas for which the Gödel–Gentzen translation size grows linearly with bounded constant overhead regardless of disjunction count and negation depth (refuting the super-additive size growth of §4.1); a Kripke frame class on which the number of upward-closed valuations per variable does not exceed the classical 2 in the limit of interest (refuting §4.2, though chains provably give $k+1 \geq 3$ for $k \geq 2$); or a proof-search model in which adding LEM instances does not multiply the path count, i.e. where instantiation choices are forced by the search strategy rather than free (refuting the independence assumption of §4.3, which would lower but, for $n \geq 1$ free choices, not eliminate the multiplier). The empirical parallel of [3] would be falsified if structured reasoning frameworks retained their isolated accuracy inside production prompts; the study's own verification at $n = 100$ reports the degradation persisting [3, §12].

Open questions. What is the overhead factor's distribution over a corpus of realistic formulas rather than one representative? Do branching frames change the expansion base $(k+1)/2$ enough to alter practical conclusions at realistic n? Does the conclusion-first mechanism of [3, §7.1] admit a formal model in which the degradation from 100% to 0–30% is a theorem rather than an analogy? These are not resolved here.

7. Practical Implications and Engineering Pathways

(a) Who would use the result and what changes. The users are (i) proof-assistant kernel and library maintainers choosing whether to assume principles that force or resist LEM — the Lean core library's assumption of propositional extensionality, used there to derive excluded middle from global choice [6, §1], is exactly such a choice, and [6] shows the alternative (global choice without LEM) is consistent in dependent type theory; (ii) OBDA engineers compiling ontological queries into database queries, where the compiled-query size growth documented as the optimization target of [2, §1] is the practical face of the translation overhead priced in §4.1; and (iii) prompt and pipeline designers deciding where in a reasoning pipeline classical-style commitment (answer-first ordering) is allowed, informed by the measured $100\% \to$ 0–30% degradation of [3, §6.1]. If the result holds, the classical/constructive choice is made against an explicit budget — a translation-size factor, an expansion factor, a branching factor — rather than by default.

(b) Readiness. This is purely analytical work that has not been peer reviewed: under the NASA Technology Readiness Level definitions it sits at TRL 1–2 (basic principles observed and formulated, analytical predictions without hardware demonstration). Analytical work does not exceed TRL 3, and nothing here approaches that; no prototype, implementation, or benchmark run exists for the overhead budget itself.

(c) Smallest next test. The smallest de-risking calculation is a corpus study, proposed conditionally: a study would take an existing corpus of proof-assistant library statements (for example, a sample of several hundred lemmas from a mature library), apply the Gödel–Gentzen translation of §3 to each, and record the empirical distribution of $|A^G|/|A|$. The outcome that would refute the paper's central claim is a distribution tightly concentrated near 1 (no systematic overhead), or one dominated by formulas where the no-simplification convention is the sole source of overhead; either would show the §4.1 factor is not representative. A second proposed test, for §4.3: instrument an existing backtracking prover to log node counts with and without LEM as an available axiom on the same problem set, and compare the measured multiplier against the 2^n lower-bound model.

(d) Binding constraints. For the corpus study, the binding constraint is data availability and formula normalization (a canonical size measure must be fixed before counts are comparable — the convention sensitivity of §4.1 is itself the first blocker). For the prover study, the constraint is instrumentation cost: logging per-path instantiation choices re-

quires prover modifications, and the re-search effect excluded from the model is exactly what the instrumentation would measure. For deployment in OBDA, the binding constraint is that compiled-query size is already an optimization target of [2, §1], so any additional translation layer must compete with existing rewrite-and-eliminate pipelines such as the Nyaya prototype over the IRIS Datalog engine [2, §7]. Whether LEM-freedom imposes measurable costs on DL-Lite$_R$ rewriting, whose underlying reasoning is 2-exptime-complete [2, §Description Logics], is an open question this paper does not answer.

(e) Order-of-magnitude projections. Labeled as projections with stated assumptions: if a realistic OBDA rewrite pipeline compiles queries whose classical content matches the shape of F (a handful of atoms, one or two disjunctions), the §4.1 factor suggests compiled-query size growth of order 2–$3\times$ attributable to the classical-to-executable translation step, under the assumptions that the pipeline's translation is double-negation-shaped and that no simplification precedes it. If a proof search touches $n = 10$ atoms with free LEM choices on a base tree of $b^d \approx 5 \times 10^5$ paths, the §4.3 model projects node counts of order 5×10^8 ($531441 \times 1024 = 544195584$), a lower bound under the stated assumptions. These projections inherit every limitation listed in Section 6 and are not measurements.

(f) Risks and misuse paths. The overhead budget could be misread as an argument that constructive logic is categorically more expensive and should be avoided; the correct reading is conditional — the costs are incurred by crossing or simulating the boundary, and the BEM contrast of §4.5 shows large computation classes where no cost arises. Conversely, the content-gap figure (2^{-n}) could be misused to claim constructive systems are unreliable; it measures the share of model space validating classical tautologies, not the correctness of constructive proof.

8. Conclusion

The classical/constructive boundary can be priced. For a representative propositional formula, importing classical content into a constructive system costs a translation-size factor of $22/9 \approx 2.44$; replacing classical valuations by Kripke valuations on chains expands the model space by $1.5^{10} \approx 57.7$ (two worlds) to $2^{10} = 1024$ (three worlds) at $n = 10$ variables; adding LEM instantiation for 5 atoms to a proof search of branching factor 3 and depth 12 multiplies the tree by 32, from 531441 to 17006112 nodes; and the constructive share of valuation space validating all n LEM instances falls as 2^{-n}, to $1/1024$ at $n = 10$ on the three-world chain. The contrast case — a 96-element, 64 N boundary-element workload from [1] — marks the computation that never crosses the boundary and pays none of these costs. The contribution is a change in how the choice is framed: the excluded middle is a computable overhead budget, with each component exponential in a structural parameter of the problem, and the budget can be audited formula by formula. The proposed next steps — a corpus-wide translation-size distribution and an instrumented prover comparison — would confirm or refute the representativeness of the factors computed here.

Provenance and AI Involvement

- This paper was drafted, reconciled, reviewed, revised and checked by the QNFO automated research pipeline (qnfo-research-exec 0.55.12). Language models were tools in that process; they are not authors. The named author is accountable for the content.
- Models: drafting by openai/gpt-oss-120b, zai-org/glm-5.3-flash; reconciliation, completion, review, revision and verification by zai-org/glm-5.3-flash, openai/gpt-oss-120b, moonshotai/kimi-k2.6, with the QNFO model router as a fallback.
- Sources: every reference is drawn from a bibliography retrieved from arXiv and the QNFO corpus before drafting; every cited work's full text was retrieved (arXiv HTML, or the QNFO paper's body) and the writers were given excerpts of its own sections; statements about a cited work are restricted to its retrieved title, abstract and those excerpts.

- Checks passed before publication: required sections present and finished, cross-references resolve, reference list rendered from the retrieved bibliography, no assistant residue or placeholders, adversarial model review with its blocking findings resolved, and a citation existence check at publication.
- Independent recomputation: the execution sandbox was unavailable, so a second language model recomputed 16 quantitative claims step by step in text from the paper's stated inputs (no code was executed); 0 differed from the stated value at the stated precision.
- Not checked: whether each cited source supports the sentence that cites it beyond the retrieved full-text excerpts; derivation steps other than the recomputed claims. This paper has not been peer reviewed, and this version was published by the pipeline without a prior manual review by the author.

References

[1] A Study of Speed of the Boundary Element Method as applied to the Realtime Computational Simulation of Biological Organs. arXiv:1311.4533v3. https://arxiv.org/abs/1311.4533v3 [2] Ontological Queries: Rewriting and Optimization (Extended Version). arXiv:1112.0343v1. https://arxiv.org/abs/1112.0343v1 [3] Prompt Complexity Dilutes Structured Reasoning: A Follow-Up Study on the Car Wash Problem. arXiv:2603.13351v1. https://arxiv.org/abs/2603.13351v1 [4] Defining the Middle Corona. arXiv:2208.04485v2. https://arxiv.org/abs/2208.04485v2 [5] Semisimplicity, Glivenko theorems, and the excluded middle. arXiv:2101.03528v1. https://arxiv.org/abs/2101.03528v1 [6] Global Choice without Excluded Middle in Dependent Type Theory. arXiv:2610.37498v1. https://arxiv.org/abs/2610.37498v1 [7] Adjoint computations by algorithmic differentiation of a parallel solver for time-dependent PDEs. arXiv:1912.11717v3. https://arxiv.org/abs/1912.11717v3 [8] https://papers.qnfo.org/papers/excluded-middle-as-computational-boundary. QNFO: Excluded Middle as Computational Boundary. [9] https://papers.qnfo.org/papers/universal-computational-topos. QNFO: Universal Computational Topos.