V1.0.0

The Sampling Ceiling: Why LLM-Authored Constraint Systems Cannot Exceed the Reliability of the Model That Authored Them

Rowan Brad Quni-Gudzinas

QNFO-ULA

ABSTRACT

Large language models (LLMs) are increasingly used to author the artifacts that other software systems must obey: specifications, system prompts, interface contracts, and validation rubrics. Practitioners currently assume that better prompting, self-critique, or meta-level instruction can raise the reliability of these authored constraint systems arbitrarily. This paper argues that assumption is false. We model LLM generation as stochastic sampling from a fixed conditional distribution and constraint satisfaction as deductive verification, and we prove a ceiling result: any prompt-level transformation of the sampling process leaves the per-constraint error rate bounded below by a model-intrinsic minimum $\varepsilon_{\min} > 0$ on constraint types the model cannot self-verify, so no composition of prompts can close the reliability gap. We quantify the gap with explicit arithmetic: for a specification with $N = 40$ constraints and a per-constraint error rate $\varepsilon = 0.02$, a prompt-only pipeline achieves a clean-run probability of $0.98^{40} \approx 0.4457$, while a hybrid pipeline in which a formal verifier catches errors with probability $c = 0.95$ achieves $(1 - \varepsilon(1 - c))^{40} \approx 0.9608$, a factor of 2.16 improvement that no prompt change can match under the model. The break-even count is $N^* = 35$ constraints. The result is consistent with reported adherence thresholds of $p^* \approx 0.958$ per chapter and $q^* \approx 0.986$ per constraint in long-form drafting, and it justifies hybrid architectures in which LLMs draft and formal tools enforce.

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 Sampling Ceiling: Why LLM-Authored Constraint Systems Cannot Exceed the Reliability of the Model That Authored Them. QNFO Preprint v1.0.0.

1. Introduction

Engineering teams that deploy LLMs face a problem that is currently solved by trial and error: the model must produce artifacts whose correctness other systems depend on. A specification written by an LLM becomes the contract for a code generator; a system prompt becomes the constitution for an agent; a rubric becomes the standard against which an evaluator scores outputs. When these artifacts contain silent errors — a constraint that contradicts another constraint, a threshold transcribed incorrectly, an edge case omitted — the downstream system fails in ways that are expensive to trace and expensive to fix. The cost is not only rework: an erroneous specification can validate an incorrect implementation, so the failure propagates through every stage that trusts it. In safety-critical settings the same failure mode carries regulatory and physical risk, which is why architects, safety engineers, and compliance officers currently spend effort on prompt refinement hoping it closes the reliability gap.

Today the standard mitigation is more language. Teams refine the prompt, add self-critique instructions, ask the model to check its own output, or wrap the model in a pipeline of models that review each other. Each of these interventions is itself an LLM-authored artifact or an LLM-executed step, and each inherits the same sampling-based generation process. The decision that would change if this paper's result holds is the decision about where to spend engineering effort: if prompt-level intervention cannot raise the reliability of an LLM-authored constraint system above the model's

own reliability, then effort should move from prompt refinement to post-hoc formal enforcement — deductive verifiers, type systems, runtime monitors, model checkers — which operate by a mechanism that is not sampling and is not subject to the ceiling.

The claim is concrete and falsifiable. We model an LLM as a sampler over strings and a prompt as a transformation of the sampling distribution. We show that for constraint types on which the model's induced per-constraint error probability is $\varepsilon > 0$ and on which the model has no internal verification channel, no prompt can reduce the induced per-constraint error rate below a bound determined by the model; the bound is a property of the model, not of the prompt. We then compute what this bound costs in practice and what post-hoc verification recovers.

A note on the form of the ceiling is needed. An alternative formulation states the ceiling as a whole-artifact correctness product, in which generation succeeds with probability p and an independent verification filter succeeds with probability v, giving system reliability $p_{\text{system}} = p \cdot v$. We adopt the per-constraint error form $\varepsilon_T \geq \varepsilon_{\min} > 0$ as the primary convention, because prompting demonstrably reconditions the sampling distribution and can move whole-artifact correctness up or down, whereas the per-constraint floor on constraint types the model cannot self-verify is the invariant that survives reconditioning. The aggregate product form is retained in Section 4 as a complementary estimate under explicitly stated assumptions.

The contributions are:

- A formal model of LLM-authored constraint systems in which generation is sampling and satisfaction is deduction, with an argument that prompt-level transformations cannot reduce per-constraint error below the model's intrinsic bound (Sections 3 and 4).
- An explicit numerical derivation of the reliability gap: for $N = 40$ constraints, $\varepsilon = 0.02$, and verifier catch probability $c = 0.95$, prompt-only pipelines reach 0.4457 clean-run probability while hybrid pipelines reach 0.9608, with the break-even constraint count $N^* = 35$ at which prompt-only pipelines fall below even odds (Section 4).
- A complementary aggregate estimate $p_{\text{system}} = p \cdot v = 0.85 \times 0.54 = 0.4590$ under an assumed base correctness and the characterization yield reported in adaptive reliability qualification (Section 4).
- A reconciliation of the ceiling result with published quantitative evidence on LLM drafting adherence, LLM-generated text detection, LLM-as-judge reliability, and adaptive reliability qualification (Sections 2 and 5).
- A design principle for hybrid architectures — LLMs draft, formal tools enforce — with the smallest experiment that would falsify or de-risk the claim (Sections 6 and 7).

The ceiling result also bears on adjacent problems that share the same structure: any pipeline in which a stochastic generator authors the standard against which its own output is judged, including automated evaluation and automated grading, inherits the same bound; and the formal-layer argument parallels the two-layer separation between human-auditable fragments and machine optimizers known in quantum circuit representation [12].

2. Background and Related Work

This section reviews the works the argument rests on. Each is cited for a specific finding its own text reports.

Constrained LLM drafting and reliability thresholds. The closest quantitative precedent is the study of AI-assisted long-form drafting [11], which computes a draft-completion rate of 76.19% of listed deliverables and an objective-success score of 16.7% under strict half-point scoring for a multi-chapter manuscript pipeline. That work derives a per-chapter reliability threshold of $p^* \approx 0.958$ for a 50% chance of a clean run and a per-constraint threshold of $q^* \approx 0.986$, and it reports that at a realistic adherence of $p = 0.9$ the phase success probability is only $P_{\text{phase}} \approx 0.185$. Its proposed falsification test is explicit: if a pilot yields per-chapter adherence below 0.958 despite refined prompts, the hypothesis that prompting alone can raise adherence sufficiently is falsified. These numbers are exactly the kind of

empirical surface the ceiling argument predicts: adherence saturates near a threshold determined by the number of independent constraints, and prompt refinement alone does not move it past that threshold. The same study reports an automated-checking benefit — the per-constraint generation threshold drops to 0.979 when automated checking is added — which is the hybrid-pattern signal this paper formalizes.

LLM-generated text is detectable but not reliably self-detected. The fine-grained detection study of LLM-generated text [3] builds detectors that recognize the role and involvement of LLM-generated content and reports strong supervised performance — average F1 of 98.75 for the Longformer-based role recognizer on the HNDC test set — but its cross-context evaluation shows degradation: overall F1 drops to 65.75 for RoBERTa in the cross-domain group, and the essay subset falls as low as 50.34. Critically for the ceiling argument, [3] reports that advanced LLMs face challenges in accurately detecting content they have generated themselves, with Mistral and Deepseek showing almost no detection capability in the zero-shot setting, and GPT-4o classifying content as human-authored when it includes specific details such as citations or data. This is direct evidence that the model's internal channel for verifying its own output is weak: the same sampling process that produces the artifact cannot reliably flag its own errors, which is the empirical premise of the ceiling argument.

Human detection of LLM hallucination is poor. The misinformation study [4] measures how hard LLM-generated misinformation is for humans to detect and reports that humans can only successfully spot 9.6% of all the generated hallucinated news on average, reflecting the difficulty of noticing fine-grained hallucinated information such as false dates, names, addresses, numbers and quotes. The same work adopts zero-shot LLM prompting as its representative detector on the argument that zero-shot LLMs have achieved near state-of-the-art performance on human-written misinformation detection. The combination — fine-grained factual errors that neither humans nor zero-shot LLMs reliably catch — is the operational form of the intrinsic error bound: errors of transcription and detail are exactly the constraint types the ceiling covers, and no prompt instructs them away because the model does not know which details are wrong.

The misinformation ecosystem framing. The survey of misinformation in the age of LLMs [5] frames LLMs as a double-edged sword in the fight and categorizes pre-LLM detection methods into seven classes, including capturing linguistic features, leveraging neural models, exploiting social context, and incorporating external knowledge. Its outlook identifies human-LLM collaboration and intervention as open directions. For the present argument, the survey's structure matters: every detection class it lists is either a statistical pattern matcher (subject to the same sampling ceiling when the LLM runs it) or an external-knowledge check (a deductive or database-grounded channel, which is the escape hatch the ceiling argument identifies). The survey thus independently maps the space into sampling-bound and deduction-bound channels.

LLM-as-judge is task-dependent and sometimes unusable. The empirical study of LLM-as-a-judge in software engineering [6] generates and manually scores LLM responses across three SE datasets and finds that judge methods demonstrate task-dependent performance, ranging from near-human to unusable when scoring individual responses, and generally perform worse in pairwise comparisons. Its implications section concludes that individual scoring should be preferred over pairwise comparison and that output-based methods with large LLMs offer the most human-aligning evaluation with proper prompt and inference strategies. The phrase "with proper prompt and inference strategies" is precisely the intervention the ceiling bounds: [6] measures how far prompt and inference strategy can move judge alignment, and the measured spread — near-human to unusable — is bounded by the underlying model's reliability on the judging task, not by the prompt space.

Genre-dependent false positives in LLM credibility assessment. The study of LLM skepticism toward entertainment news [8] uses a within-dataset design on GossipCop with $1{,}421$ real entertainment-gossip articles and 379 real hard-news articles and defines the main outcome as $\Delta\mathrm{FPR} = \mathrm{FPR}_{\text{entertainment}} - \mathrm{FPR}_{\text{hard-news}}$, testing whether zero-shot LLMs misclassify legitimate entertainment news as fake more often than hard news. Its style-swap experiment uses DeepSeek-V3.2 to rewrite 50 real entertainment articles into hard-news style while keeping factual content stable, iso-

lating stylistic form from factual content. The design demonstrates the methodological point the ceiling argument needs: to attribute an error rate to the model rather than to the prompt or the domain, one must hold content fixed and vary only the surface — and when that is done, residual error differences are properties of the model's conditional distribution, which is what the ceiling bounds.

Preference optimization as distribution shaping. The disaster-severity assessment work [1] proposes a two-stage training framework integrating Supervised Fine-Tuning and Direct Preference Optimization to improve both prediction accuracy and explanation quality under scarce annotated data. Preference optimization reshapes the sampling distribution — it is training-level, not prompt-level, intervention — and [1]'s positioning that existing approaches are limited by scarce annotated data and insufficient evaluation of reasoning quality supports the distinction this paper draws: changing the distribution (training) and changing the prompt (conditioning) are different levers, and only the former can move the intrinsic error rate. The ceiling applies to the latter.

Reliability qualification as sequential decision-making. The adaptive test-planning work [7] frames reliability qualification of semiconductor devices as a sequential decision problem and reports that over $5{,}000$ planning iterations an adaptive MCTS-based planner improves characterization yield from 20% to over 54%, achieving 39% cumulative yield while keeping damage fractions below 0.57. Its metrics — characterization yield, catastrophe rate, test length, final uncertainty — are the quantities a reliability argument must trade off, and its finding that static test plans derived from population-level acceleration models cannot adapt to per-unit variability parallels the present claim that static prompt-level plans cannot adapt to per-constraint error structure. The yield improvement from 20% to 54% was obtained by changing the decision process (closed-loop search with belief-state estimation), not by re-instructing a fixed process — the same lever change this paper advocates for specification reliability.

Deductive reformulation makes simulation possible. The seismic network reliability work [2] proposes two informative limit-state functions that reformulate the traditional binary limit-state function for two-terminal reliability analysis of lifeline networks, thereby making the application of subset simulation possible, and generates network fragility curves in a single run, extended to k-terminal or k-out-of-N reliability. The methodological lesson transfers directly: a reliability question that is intractable in its native formulation becomes computable when reformulated so that intermediate failure events are informative. Section 3 performs the analogous reformulation for LLM-authored specifications, replacing the binary question "is the specification correct" with per-constraint error events that can be counted and bounded.

Statistical rigor as the falsifier of beautiful claims. The adelic-constraints project [9] reports an honest negative result: a numerological hypothesis matching zeta-zero gaps to particle mass ratios was tested with five hundred zeta zeros, Bonferroni correction, and a permutation null model, and all three candidate matches were consistent with chance after correction, demonstrating that mathematical beauty does not imply physical truth. This is the epistemic standard Section 6 adopts: the ceiling argument is elegant, and elegance is not evidence; the falsification conditions stated there are the analogue of [9]'s permutation null.

Formal layers make claims testable. The attractor-calculus work [10] states explicitly that the conjecture is not tested there and that every result is a theorem about a one-dimensional model potential plus labeled assumptions, and that the formal layer makes a specific, falsifiable prediction about cost structure which the empirical program is the test of. This paper takes the same posture: the ceiling result is a theorem about a model of sampling and deduction, its assumptions are labeled in Section 3, and the empirical test is proposed in Section 7, not performed here.

Two-layer architectures for bounded reasoners. The representation-design work [12] argues that when the maps that are easiest to understand are not the ones that are cheapest to compute with, the field's answer is a split into a human-facing auditable fragment and a machine-facing optimizer connected by an interface where a human can still audit what the optimizer did. The hybrid architecture this paper recommends has the same shape with the roles reassigned: the LLM is the machine-facing generator whose output cannot be audited by sampling, and the formal verifier is the au-

ditable layer whose verdicts are deductive. [12]'s quantitative anchor — at six qubits the matrix has 4,096 entries while the diagram has sixteen — illustrates how representation determines what a bounded reasoner can check, which is why LLM self-checking fails: the model checks its output in the same representation in which it generated it.

3. Methods

3.1 The model

We model an LLM as a conditional distribution over strings. For a prompt x and a temperature parameter τ, the model defines

$$P_{\theta,\tau}(y \mid x) = \prod_{t=1}^{|y|} P_{\theta,\tau}(y_t \mid x, y_{<t}),$$

an autoregressive product of per-token conditionals. Generation is sampling: an output y is drawn from this distribution. A prompt x' is itself a string; prompting is the act of choosing x' so that the conditional $P_{\theta,\tau}(y \mid x')$ concentrates on desirable y.

A constraint system is a finite set $C = \{c_1, \ldots, c_N\}$ of checkable propositions. An authored artifact y satisfies constraint c_i or violates it; satisfaction is a deductive question — decidable by a verifier V with $V(y, c_i) \in \{\text{pass}, \text{fail}\}$ — not a sampling question. We call the gap between these two mechanisms the sampling–deduction gap.

3.2 Assumptions

The ceiling result rests on four labeled assumptions.

- **A1 (Fixed model).** The model parameters θ and decoding temperature τ are fixed; no gradient update occurs between generations. Prompting selects x'; it does not change $P_{\theta,\tau}$.
- **A2 (Intrinsic error).** There exists a class of constraints $\mathcal{C}_{\text{hard}}$ (for example: transcribed numeric thresholds, cross-references between distant parts of the artifact, factual details) on which the model's induced per-constraint error rate is $\varepsilon = \Pr[V(y, c) = \text{fail} \mid c \in \mathcal{C}_{\text{hard}}] > 0$ for every prompt x' in the admissible prompt space, and the model has no internal channel that identifies which of its own outputs violate c.
- **A3 (Prompt is conditioning only).** An admissible prompt transformation T maps prompts to prompts; the transformed generation is $P_{\theta,\tau}(y \mid T(x))$. No transformation inserts an execution of V.
- **A4 (Independent constraint errors).** Constraint violations are conditionally independent given the prompt, so the clean-run probability is the product of per-constraint success probabilities. This is the same independence structure used to derive the thresholds in [11]; we state it as an assumption and test its sensitivity in Section 6.

Assumption A2 is the empirical premise. It is supported, not proved, by the cited evidence: [3] reports that advanced LLMs cannot reliably detect their own generated content, and [4] reports 9.6% average human detection of hallucinated news, indicating that fine-grained hallucination is not flagged by any cheap channel. The argument shows what follows if A2 holds; Section 7 proposes the measurement of ε.

3.3 The ceiling argument

Let T be any admissible prompt transformation and let $\varepsilon_T = \Pr[V(y, c) = \text{fail} \mid c \in \mathcal{C}_{\text{hard}},\ y \sim P_{\theta,\tau}(\cdot \mid T(x))]$. Under A1 and A3, T acts only by reconditioning: it changes the distribution over y but cannot execute V, cannot compare y against an external ground truth, and cannot access a channel that labels the model's own errors. Under A2, for every admissible T,

$$\varepsilon_T \geq \varepsilon_{\min} > 0,$$

where $\varepsilon_{\min}$ is the infimum of the induced error rate over the admissible prompt space, a property of $(\theta, \tau, \mathcal{C}_{\text{hard}})$ alone. The proof structure is a no-go argument: if some T drove ε_T to zero, then the pair $(T, P_{\theta,\tau})$ would constitute an internal verification channel for $\mathcal{C}_{\text{hard}}$, contradicting A2. The composition of k prompt transformations is itself an admissible transformation (the concatenated prompt), so no composition escapes the bound. Self-critique pipelines — model reviews model — are compositions of sampling steps and inherit the same bound, which is the empirical content of [3]'s finding that models cannot detect their own output and [6]'s finding that judge performance is bounded and task-dependent.

By contrast, a post-hoc verifier V applied to y operates by deduction: its catch probability c is a property of the verifier and the constraint class, not of the sampler. A hybrid pipeline that generates with the LLM and then runs V, regenerating or repairing on failure, has residual per-constraint error

$$\varepsilon_{\text{hyb}} = \varepsilon_T(1 - c),$$

which can be driven toward zero by increasing c — a lever that does not exist at the prompt level.

3.4 Computation plan

Section 4 computes, with shown arithmetic: (i) the clean-run probability of a prompt-only pipeline as a function of N and ε; (ii) the clean-run probability of the hybrid pipeline as a function of N, ε, and c; (iii) the break-even constraint count N^* at which the prompt-only pipeline falls below 0.5; (iv) the ratio of hybrid to prompt-only reliability; (v) a sensitivity sweep over ε and c; and (vi) a complementary aggregate estimate $p_{\text{system}} = p \cdot v$. All inputs are either stated as assumptions or cited from [11] and [7].

4. Analysis

4.1 Inputs and their sources

We fix the following inputs. Numbers marked *assumption* are stand-ins for data no cited source gives; numbers marked *cited* come from the named work.

- $N = 40$ constraints per specification (*assumption*: a mid-sized interface specification; the sensitivity sweep covers $N \in \{10, 20, 40, 80\}$).
- $\varepsilon = 0.02$ per-constraint error rate on $\mathcal{C}_{\text{hard}}$ (*assumption*, chosen to sit between the per-constraint threshold $q^* \approx 0.986$ cited from [11] — which corresponds to $\varepsilon \approx 0.014$ — and a deliberately pessimistic 0.02).
- $c = 0.95$ verifier catch probability (*assumption*: a deductive checker for well-formed constraint types such as type errors, range violations, and contradictions; the sweep covers $c \in \{0.8, 0.9, 0.95, 0.99\}$).
- Reference thresholds from [11] (*cited*): $p^* \approx 0.958$ per chapter for a 50% clean-run chance, $q^* \approx 0.986$ per constraint, $P_{\text{phase}} \approx 0.185$ at $p = 0.9$, and the automated-checking threshold 0.979.
- For the complementary aggregate estimate: base correctness $p = 0.85$ (*assumption*) and verification yield $v = 0.54$ (*cited* as "over 54%" in [7]; we use the lower bound).

4.2 Prompt-only pipeline reliability

Under A4, the probability that a specification with N constraints is entirely clean is

$$P_{\text{clean}}^{\text{prompt}} = (1 - \varepsilon)^N.$$

With $\varepsilon = 0.02$ and $N = 40$:

$$1 - \varepsilon = 0.98, \qquad \ln(0.98) = -0.0202027,$$

$$40 \times (-0.0202027) = -0.808108,$$

$$P_{\text{clean}}^{\text{prompt}} = e^{-0.808108} \approx 0.4457.$$

So a prompt-only pipeline with these inputs produces a fully clean specification less than half the time.

4.3 Hybrid pipeline reliability

The hybrid pipeline runs V after generation. A violated constraint escapes only if the verifier misses it, which has probability $1 - c$. The residual per-constraint error rate is

$$\varepsilon_{\text{hyb}} = \varepsilon(1 - c) = 0.02 \times (1 - 0.95) = 0.02 \times 0.05 = 0.001.$$

The clean-run probability is

$$P_{\text{clean}}^{\text{hyb}} = (1 - 0.001)^{40} = 0.999^{40}.$$

Computing: $\ln(0.999) = -0.0010005$, so $40 \times (-0.0010005) = -0.0400201$, and

$$P_{\text{clean}}^{\text{hyb}} = e^{-0.0400201} \approx 0.9608.$$

4.4 The gap and the break-even count

The reliability ratio is

$$\frac{P_{\text{clean}}^{\text{hyb}}}{P_{\text{clean}}^{\text{prompt}}} = \frac{0.9608}{0.4457} \approx 2.156.$$

The break-even constraint count N^* is the smallest integer N with $(1 - \varepsilon)^N < 0.5$:

$$N > \frac{\ln 0.5}{\ln 0.98} = \frac{-0.693147}{-0.0202027} = 34.31,$$

so $N^* = 35$: at 35 or more hard constraints, a prompt-only pipeline with $\varepsilon = 0.02$ fails more often than it succeeds, regardless of prompt quality, under A2 and A4.

4.5 Consistency with the cited drafting thresholds

The per-constraint threshold $q^* \approx 0.986$ cited from [11] corresponds to an error rate $\varepsilon^* = 1 - 0.986 = 0.014$. We check what clean-run probability that error rate implies at $N = 40$:

$$(0.986)^{40}: \quad \ln(0.986) = -0.0140994, \quad 40 \times (-0.0140994) = -0.563975, \quad e^{-0.563975} \approx 0.5691.$$

So even at the threshold adherence measured in [11], a 40-constraint artifact is clean only 56.9% of the time — consistent with that work's reported objective-success score of 16.7% under strict scoring for a much larger multi-chapter artifact, where the effective constraint count is in the hundreds. We verify the scaling: for $N = 300$ constraints at $\varepsilon = 0.014$,

$$300 \times (-0.0140994) = -4.22982, \quad e^{-4.22982} \approx 0.0146,$$

i.e., a 1.5% clean-run probability, the regime in which [11] reports 16.7% under strict scoring and 76.19% mere completion (completion without full constraint satisfaction). The orders of magnitude agree.

4.6 Sensitivity sweep

We compute $P_{\text{clean}}^{\text{prompt}} = (1-\varepsilon)^N$ and $P_{\text{clean}}^{\text{hyb}} = (1-\varepsilon(1-c))^N$ over the sweep grid. Each entry is $e^{N\ln(1-\varepsilon)}$ or $e^{N\ln(1-\varepsilon(1-c))}$.

For $\varepsilon = 0.02$, $c = 0.95$ ($\varepsilon_{\text{hyb}} = 0.001$):

N	P_{prompt}	P_{hyb}	**ratio**
10	$0.98^{10} = e^{-0.202027} \approx 0.8170$	$0.999^{10} = e^{-0.010005} \approx 0.9900$	1.21
20	$0.98^{20} = e^{-0.404054} \approx 0.6676$	$0.999^{20} = e^{-0.020010} \approx 0.9802$	1.47
40	$0.98^{40} = e^{-0.808108} \approx 0.4457$	$0.999^{40} = e^{-0.040020} \approx 0.9608$	2.16
80	$0.98^{80} = e^{-1.616216} \approx 0.1986$	$0.999^{80} = e^{-0.080040} \approx 0.9231$	4.65

For $c = 0.80$ ($\varepsilon_{\text{hyb}} = 0.02 \times 0.2 = 0.004$), $N = 40$:

$$P_{\text{clean}}^{\text{hyb}} = 0.996^{40}: \quad \ln(0.996) = -0.0040080, \quad 40 \times (-0.0040080) = -0.160320, \quad e^{-0.160320} \approx 0.8518.$$

For $c = 0.99$ ($\varepsilon_{\text{hyb}} = 0.0002$), $N = 40$:

$$P_{\text{clean}}^{\text{hyb}} = 0.9998^{40}: \quad \ln(0.9998) = -0.00020002, \quad 40 \times (-0.00020002) = -0.0080008, \quad e^{-0.0080008} \approx 0.9920.$$

For $\varepsilon = 0.05$ (pessimistic), $c = 0.95$ ($\varepsilon_{\text{hyb}} = 0.0025$), $N = 40$:

$$P_{\text{prompt}} = 0.95^{40}: \quad \ln(0.95) = -0.0512933, \quad 40 \times (-0.0512933) = -2.051731, \quad e^{-2.051731} \approx 0.1285.$$

$$P_{\text{hyb}} = 0.9975^{40}: \quad \ln(0.9975) = -0.0025031, \quad 40 \times (-0.0025031) = -0.100125, \quad e^{-0.100125} \approx 0.9047.$$

The pattern is uniform: the hybrid multiplier grows with N and with ε, and no admissible prompt appears anywhere in the hybrid formula — the verifier's catch probability c is the only lever that enters.

4.7 What prompt refinement can and cannot do

Under A2, prompt refinement can move ε_T within $[\varepsilon_{\min}, \varepsilon_{\max}]$ but not below $\varepsilon_{\min}$. Suppose the best achievable prompt refinement halves the error rate to $\varepsilon = 0.01$ (this is a *projection*, assuming a best-case refinement factor of 2; no cited source measures it). Then at $N = 40$:

then at $N = 40$:

$$P_{\text{clean}}^{\text{prompt}} = 0.99^{40}: \quad \ln(0.99) = -0.0100503, \quad 40 \times (-0.0100503) = -0.402014, \quad e^{-0.402014} \approx 0.6690.$$

So even a best-case halving of the error rate lifts the prompt-only clean-run probability only from 0.4457 to 0.6690 — still far below the hybrid value of 0.9608 computed in Section 4.3. This is the quantitative content of the ceiling: prompt refinement moves along the curve $(1-\varepsilon)^N$, and the curve's shape, not the prompt, determines how far movement can go. The hybrid lever, by contrast, changes the exponent's argument from ε to $\varepsilon(1-c)$, and c is not bounded by the model.

5. Results

All numbers in this section are computed in Section 4 with shown arithmetic; none is measured. They are properties of the model of Section 3 under the labeled assumptions A1–A4 and the inputs of Section 4.1.

Quantity	Symbol	Value	Source
Prompt-only clean-run probability ($N = 40$, $\varepsilon = 0.02$)	$P_{\text{clean}}^{\text{prompt}}$	≈ 0.4457	Section 4.2
Hybrid clean-run probability ($N = 40$, $\varepsilon = 0.02$, $c = 0.95$)	$P_{\text{clean}}^{\text{hyb}}$	≈ 0.9608	Section 4.3
Reliability ratio	$P_{\text{hyb}} / P_{\text{prompt}}$	≈ 2.156	Section 4.4
Break-even constraint count	N^*	35	Section 4.4
Clean-run probability at the cited threshold ($q^* = 0.986$, $N = 40$)	—	≈ 0.5691	Section 4.5
Clean-run probability at $N = 300$, $\varepsilon = 0.014$	—	≈ 0.0146	Section 4.5
Hybrid clean-run probability at $c = 0.80$, $N = 40$	—	≈ 0.8518	Section 4.6
Hybrid clean-run probability at $c = 0.99$, $N = 40$	—	≈ 0.9920	Section 4.6
Prompt-only clean-run probability at $\varepsilon = 0.05$, $N = 40$	—	≈ 0.1285	Section 4.6
Hybrid clean-run probability at $\varepsilon = 0.05$, $c = 0.95$, $N = 40$	—	≈ 0.9047	Section 4.6
Best-case prompt refinement ($\varepsilon = 0.01$, $N = 40$; projection)	—	≈ 0.6690	Section 4.7
Aggregate estimate $p_{\text{system}} = p \cdot v$	—	$0.85 \times 0.54 = 0.4590$	Section 4.1

Three results carry the paper's claim. First, at the assumed operating point the prompt-only pipeline produces a fully clean specification 44.6% of the time while the hybrid pipeline produces one 96.1% of the time, a factor of 2.16. Second, the break-even count $N^* = 35$ means that at 35 or more hard constraints the prompt-only pipeline fails more often than it succeeds; the failure is a property of N and ε, not of prompt quality. Third, the sensitivity sweep shows the hybrid multiplier growing monotonically in N and ε — from 1.21 at $N = 10$ to 4.65 at $N = 80$ — so the harder the specification, the larger the advantage of the non-sampling lever.

The consistency checks of Section 4.5 connect the model to the cited evidence. The per-constraint threshold $q^* \approx 0.986$ from [11] implies a 40-constraint clean-run probability of 0.5691, and the same threshold extrapolated to a 300-constraint artifact implies 0.0146 — the regime in which [11] reports an objective-success score of 16.7% under strict scoring and a draft-completion rate of 76.19%. The model's predicted orders of magnitude agree with the measured ones without any fitted parameter. The automated-checking benefit reported in [11] — the per-constraint threshold dropping from 0.986 to 0.979 when automated checking is added — is the direction the hybrid formula predicts: adding a non-sampling check channel relaxes the requirement on the sampler.

6. Discussion

What the result is and is not. The ceiling is a theorem about a model: under A1–A4, no admissible prompt transformation reduces the per-constraint error rate on $\mathcal{C}_{\text{hard}}$ below $\varepsilon_{\min} > 0$. It is not a measurement of any deployed system, and it does not say that prompting is useless — Section 4.7 shows a projected best-case refinement halving ε and lifting clean-run probability from 0.4457 to 0.6690, a real gain. The claim is that the gain is bounded by the model, while the hybrid lever is not.

The aggregate form as a complementary estimate. Section 4.1 also carries the whole-artifact product form $p_{\text{system}} = p \cdot v$ with $p = 0.85$ (assumption) and $v = 0.54$ (the lower bound of the characterization yield reported in [7]), giving 0.4590. This form is more pessimistic than the per-constraint form at the same operating point because it charges the full generation failure rather than only the residual error; we retain it as a bounding estimate under explicitly stated assumptions, per the convention declared in Section 1.

Failure modes of the argument. The argument fails if any of its assumptions fails. If A2 is false — if for every constraint type the model has an internal channel that reliably identifies its own violations — then a prompt transformation could in principle route generation through that channel and drive ε_T toward zero, and the ceiling collapses. The evi-

dence of [3] (advanced LLMs show almost no zero-shot capability at detecting their own generated content, with Mistral and Deepseek near total inability on the involvement-measurement task) and [4] (9.6% average human detection of hallucinated news, with fine-grained errors in dates, names, numbers and quotes the hardest to notice) supports A2 for fine-grained factual constraint types, but A2 is an empirical premise, not a theorem. If A3 is false — if a "prompt" can smuggle in execution of a verifier, for example by instructing the model to emit and run code — then the transformation is no longer prompt-level in the sense defined, and the ceiling does not apply; such a pipeline is a hybrid pipeline by the paper's own classification. If A4 fails — if constraint errors are correlated, for example because one misunderstanding produces many violations — then the product form overestimates the penalty of large N, and the break-even count shifts; correlated errors can cut either way, since a single root cause is also a single target for a single verifier check.

What would falsify the main result. The main result is falsified by a measurement: if a controlled study finds a prompt-only pipeline whose per-constraint error rate on a fixed $\mathcal{C}_{\text{hard}}$ class (transcribed thresholds, cross-references, factual details) falls below the model's measured $\varepsilon_{\min}$ across an adversarially wide prompt space — or, equivalently, if refined prompting drives the clean-run probability of a 40-constraint specification past 0.96 at the assumed ε without any non-sampling check — then either A2 or A3 is false and the ceiling does not hold. The falsification test proposed in [11] has the same shape at the drafting level: if a pilot yields per-chapter adherence below 0.958 despite refined prompts, the hypothesis that prompting alone can raise adherence sufficiently is falsified; the present paper predicts the opposite branch, that adherence saturates near the threshold. A second falsifier is internal: if the independence assumption A4 is dropped and the true clean-run probability under correlated errors at $N = 40$, $\varepsilon = 0.02$ exceeds 0.96 without verification, the quantitative gap claimed here is an artifact of the product form.

Limitations list.

- *A1 (fixed model).* The result holds per model snapshot. Training-level interventions — fine-tuning, preference optimization as in [1] — change $P_{\theta,\tau}$ itself and can move $\varepsilon_{\min}$; the ceiling binds only within a snapshot. Sensitivity: the entire result is conditional on this restriction, which is stated, not derived.
- *A2 (intrinsic error).* The existence and magnitude of $\varepsilon_{\min}$ are assumed, supported only indirectly by [3] and [4]. The value $\varepsilon = 0.02$ used throughout Section 4 is an assumption; if the true rate is 0.005, the prompt-only clean-run probability at $N = 40$ rises to $0.995^{40} = e^{40\times(-0.0050125)} = e^{-0.200501} \approx 0.8184$, and the gap to the hybrid value narrows from a factor of 2.16 to about 1.17.
- *A3 (prompt is conditioning only).* The admissible prompt space excludes tool-executing prompts by definition. Systems that blur this line are hybrids, not counterexamples.
- *A4 (independence).* All product-form numbers depend on it. Correlated violations would change N^* and every table entry; the direction of the change is not determined without a correlation model, which this paper does not build.
- *Constraint class boundary.* The ceiling covers $\mathcal{C}_{\text{hard}}$ only. Constraints the model can self-verify reliably (for example, syntactic well-formedness) may be driven to negligible error by prompting alone; the ceiling claims nothing about them.
- *Verifier coverage.* The hybrid numbers assume the verifier's catch probability *c* applies to the same constraint class the sampler errs on. A verifier that checks a disjoint class leaves the sampler's errors untouched, and the hybrid advantage vanishes for the uncovered class.
- *No empirical test performed.* This paper performs no experiment, measurement, or pilot; every number is derived from the stated inputs or cited from [11] and [7]. The empirical program is proposed in Section 7.

Open questions. What is the measured $\varepsilon_{\min}$ for frontier models on specification-style constraint classes? How does $\varepsilon_{\min}$ scale with model size and with constraint locality (distance in the artifact between the constraint's source and its site of use)? Can verifier coverage c be made complete for a restricted specification language, and at what authoring cost? These are measurement and engineering questions the ceiling frames but does not answer.

7. Practical Implications and Engineering Pathways

(a) Who would use the result and what changes. The users are engineering teams that deploy LLMs to author specifications, interface contracts, evaluation rubrics, and agent constitutions; the architects and compliance officers who sign off on those artifacts; and the maintainers of LLM-evaluation pipelines whose judges are themselves LLMs. If the result holds, the decision that changes is the allocation of reliability effort: budget that currently goes into prompt refinement, self-critique loops, and model-review-model chains moves to post-hoc formal enforcement — type systems, schema validators, model checkers, runtime monitors, and deductive verifiers over restricted specification languages. Concretely, a team authoring a 40-constraint interface specification should expect a prompt-only pipeline to deliver a fully clean artifact roughly 45% of the time at the assumed operating point (a projection under the inputs of Section 4.1, not a measurement), and should plan for a verification stage that catches violations with the highest achievable c, because Section 4.6 shows the clean-run probability rising from 0.8518 at $c = 0.80$ to 0.9920 at $c = 0.99$ — the verifier's coverage, not the prompt, is the binding lever.

(b) Current readiness. This work is analytical and has not been peer reviewed; against the NASA Technology Readiness Level definitions it sits at TRL 1–2: the basic principles (the sampling–deduction separation and the no-go argument) are formulated and the quantitative consequences are derived on paper, but no experimental proof of concept has been demonstrated. The claim must not be read as a validated method; the hybrid numbers are model outputs under labeled assumptions.

(c) The smallest next experiment. A pilot study would measure $\varepsilon_{\min}$ directly. The design: fix a constraint class $\mathcal{C}_{\text{hard}}$ (for example, 40 transcribed numeric thresholds and cross-references in an interface specification), generate the specification under k maximally diverse prompts (adversarially selected to span the prompt space), run a deductive verifier over every constraint of every output, and estimate the per-constraint error rate ε_T for each prompt. The falsifying outcome is stated in Section 6: if the minimum estimated ε_T across prompts is zero (or statistically indistinguishable from zero at the study's power), A2 is false for that class and the ceiling does not bind. If instead all prompts yield $\varepsilon_T \geq \varepsilon_{\min} > 0$, the ceiling's premise is confirmed and the measured $\varepsilon_{\min}$ replaces the assumed 0.02 in every number of Section 4. A companion arm would add the verifier and measure the residual rate, testing the prediction $\varepsilon_{\text{hyb}} = \varepsilon_T(1 - c)$.

(d) Binding constraints. The first binding constraint is verifier coverage: the hybrid advantage exists only where a deductive check exists, and for open-ended semantic constraints (is this requirement testable? is this rubric fair?) no verifier is known, so the ceiling binds with no escape hatch — this is an open question, not a solved case. The second is specification language design: achieving high c requires constraining the authoring format so that constraints are machine-checkable, which trades authoring expressiveness for verifiability; the two-layer pattern of [12] — an auditable fragment connected to a machine-facing optimizer — is the design template, and its quantitative anchor (at six qubits, a matrix of $4{,}096$ entries against a diagram of sixteen) illustrates why the auditable fragment must be small enough for a bounded reasoner to check. The third is cost of the verification stage itself: regenerating on verifier failure multiplies generation cost by a factor related to the pass rate; at the assumed operating point the per-constraint pass rate after verification is 0.999, so the expected number of regeneration rounds is near one, but at low c or high ε the regeneration loop dominates cost — the exact cost model is an open question this paper does not derive. Regulatory constraints on safety-critical specifications may in practice mandate deductive artifacts regardless of the ceiling, which would make adoption a compliance question rather than an engineering one.

(e) Order-of-magnitude cost bounds (projection). Using the cost model reported in [11] — a cloud LLM price of \0.02per1Mtokens, acostof\0.00005 per 2500-token chapter, and \0.0008totalfor16$ chapters — generation cost is negligible relative to verification engineering cost at this scale, so the binding investment is building and maintaining the verifier, not running the pipeline. This projection inherits [11]'s price assumption and applies only at that token scale; it is not a measurement.

(f) Risks and misuse paths. The ceiling can be misread as a license to skip verification: if a team concludes that no prompt can fix errors, it may also conclude that nothing can, and ship unverified LLM-authored constraints. The correct reading is the opposite — the result identifies exactly which lever works. A second misuse path runs through the detectability evidence: [3] shows supervised detectors reach high F1 in-distribution (Longformer-based role recognizer average F1 of 98.75 on the HNDC test set) but degrade cross-domain (overall F1 of 65.75 for RoBERTa in the cross-domain group), and [4] shows fine-grained hallucination evades both humans (9.6% detection) and zero-shot LLM detectors; a bad actor could use these gaps to route LLM-authored misinformation or fraudulent specifications past statistical checks, which is an argument for grounding detection in external-knowledge channels of the kind [5] categorizes rather than in sampling-based ones.

8. Conclusion

This paper modeled LLM generation as stochastic sampling and constraint satisfaction as deduction, and derived a ceiling: under the labeled assumptions A1–A4, no prompt-level transformation of the sampling process reduces the per-constraint error rate on constraint types the model cannot self-verify below a model-intrinsic bound $\varepsilon_{\min} > 0$, because any transformation that did would constitute the internal verification channel the assumption denies. The quantitative consequence, computed with shown arithmetic, is that at $N = 40$ constraints and an assumed $\varepsilon = 0.02$, a prompt-only pipeline achieves a clean-run probability of 0.4457 while a hybrid pipeline with verifier catch probability $c = 0.95$ achieves 0.9608 — a factor of 2.16 — with the prompt-only pipeline falling below even odds at $N^* = 35$ constraints. The model's predictions agree in order of magnitude with the measured thresholds of AI-assisted long-form drafting [11] without fitted parameters, and the empirical premises are consistent with the self-detection failures reported for LLM-generated text [3] and the low human detection rate of LLM-generated misinformation [4]. The practical conclusion is a division of labor: LLMs draft, formal tools enforce. The verifier's catch probability is the only lever in the reliability formula that is not bounded by the sampler, and engineering effort belongs there. The result's scope is a single model snapshot under independent constraint errors, and its premise — the existence of $\varepsilon_{\min}$ — is an empirical claim awaiting the measurement proposed in Section 7; if that measurement returns zero, the ceiling falls with it.

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] Towards reliable multimodal disaster severity assessment through preference optimization and explainable vision-language reasoning. arXiv:2609.00879v1. https://arxiv.org/abs/2609.00879v1 [2] Efficient seismic reliability and fragility analysis of lifeline networks using subset simulation. arXiv:2310.10232v2. https://arxiv.org/abs/2310.10232v2 [3] Beyond Binary: Towards Fine-Grained LLM-Generated Text Detection via Role Recognition and Involvement Measurement. arXiv:2410.14259v2. https://arxiv.org/abs/2410.14259v2 [4] Can LLM-Generated Misinformation Be Detected?. arXiv:2309.13788v5. https://arxiv.org/abs/2309.13788v5 [5] Combating Misinformation in the Age of LLMs: Opportunities and Challenges. arXiv:2311.05656v1. https://arxiv.org/abs/2311.05656v1 [6] Can LLMs Replace Human Evaluators? An Empirical Study of LLM-as-a-Judge in Software Engineering. arXiv:2502.06193v3. https://arxiv.org/abs/2502.06193v3 [7] Adaptive Sequential Test Planning for Multi-Mechanism Reliability Qualification via Bayesian Monte Carlo Tree Search. arXiv:2608.09622v1. https://arxiv.org/abs/2608.09622v1 [8] Are LLMs More Skeptical of Entertainment News?. arXiv:2605.01727v1. https://arxiv.org/abs/2605.01727v1 [9] https://papers.qnfo.org/papers/adelic-constraints-on-quantum-field-theory-phase-1. QNFO: Adelic Constraints on Quantum Field Theory: Phase 1. [10] https://papers.qnfo.org/papers/attractor-calculus-a-formal-operator-language-for-steering-cognition-across-subs. QNFO: Attractor Calculus: A Formal Operator Language for Steering Cognition Across Substrates. [11] https://papers.qnfo.org/papers/close-slobphase1drafting-v10. QNFO: Quantitative Assessment of AI-Assisted Long-Form Drafting: Completion Rates and Reliability Thresholds. [12] https://papers.qnfo.org/papers/visual-graphs-vs-matrices-epistemic-design. QNFO: Visual Graphs vs Matrices: Epistemic Limits, Cognitive Preference, and the Design of Understandable Computation.