Limits of Self-Correction in LLMs: An Information-Theoretic Analysis of Correlated Errors

Andrew Michael Brilliant *

Posted Date: 25 May 2026

doi: 10.20944/preprints202601.0892.v5

Keywords: LLM; self-correction; information theory; error correlation; external selection; multi-agent verification; context separation; language models; reasoning; validation

 doi:10.20944/preprints202601.0892.v5

Article

Limits of Self-Correction in LLMs: An Information-Theoretic Analysis of Correlated Errors

Andrew Michael Brilliant

Independent Researcher, Sapporo, Japan; a.brilliant@ieee.org

Abstract

We develop a diagnostic framework for evaluating when LLM self-evaluation can be trusted. The framework's central results are: (1) under a shared-blind-spot modeling assumption with joint conditional independence of evaluations given the shared failure structure, k rounds of self-critique provide information about correctness bounded by what the shared latent failure variable Z mediates—not by any independent channel—so that confidence accumulated through repeated self-evaluation reflects the shared failure structure rather than independently accumulated evidence; and (2) a selector satisfying two independently measurable sufficient conditions—bounded false-acceptance and true-acceptance exceeding that bound—provides a quantifiable lower bound on evidence about correctness. Both results are conditional on explicit modeling assumptions. We also prove an information-theoretic bound showing that self-evaluation is bounded in what it can add when a shared latent failure structure mediates both generation and evaluation errors; we foreground this as scaffolding rather than a primary contribution, since the latent variable requires independent operationalization to give the bound empirical bite. The diagnostic framework identifies what to measure to determine whether a deployed system is in the failure regime, and what properties an external selector must have to escape it. We describe design principles for an architecture motivated by this analysis; same-model context separation is an engineering heuristic, not a theoretical solution, and we present it as a practical starting point pending empirical validation.

Keywords: LLM; self-correction; information theory; error correlation; external selection; multi-agent verification; context separation; language models; reasoning; validation

1. Introduction

Recent work demonstrates that large language models cannot reliably self-correct their reasoning Huang et al. [11]. Tsui [26] provides further evidence that this failure is not a knowledge deficit. LLMs can correct identical errors when presented as external input but fail to correct those same errors in their own outputs, a phenomenon termed the **Self-Correction Blind Spot**, measured at an average 64.5% failure rate across 14 models. We argue this reflects a structural property of certain evaluation configurations: when generator and evaluator share failure modes, self-evaluation provides weak evidence of correctness. A derivation may be elegant, internally consistent, and convincingly presented, yet contain a subtle error that the model cannot detect in its own output.

This failure is not primarily about compute or scale. It is about validation. Reliable workflows require a mechanism that separates signal from noise, correct outputs from plausible-but-wrong ones. The question is: what properties must a validation mechanism have to be reliable?

1.1. The Core Problem

Consider a system that generates a hypothesis and then evaluates whether that hypothesis is correct. Under what conditions does self-evaluation provide useful information?

We argue that the answer depends on error correlation. When the evaluator makes errors on the same inputs where the generator makes errors, self-evaluation can be non-identifying: agreement

 doi:10.20944/preprints202601.0892.v5

between generator and evaluator may provide weak evidence of correctness. This echoes well-documented phenomena in human reasoning: confirmation bias, the curse of knowledge, and why peer review and second opinions exist at all. The difference with LLMs is that context can be deleted. A fresh instance has no memory of the reasoning it might defend.

This is not a claim about any specific model's limitations. It is a structural property of certain evaluation configurations. A single agent evaluating its own outputs faces elevated risk of correlated error: sharing training data, inductive biases, and blind spots creates the conditions under which the formal bounds of Section 3 may apply.[1] Tsui [26] offers empirical support for this structural claim. Models that successfully correct errors in externally-presented solutions fail on their own identical errors at substantially higher rates, suggesting that the knowledge to detect errors exists but is not activated during self-evaluation.

1.2. The Deep Context Challenge

This problem becomes acute as context windows expand. Modern LLMs support large context windows, enabling extended reasoning: complex derivations, multi-step analyses, and long research sessions.

But large context is where correlated error accumulation may be most severe. Each reasoning step inherits context from previous steps. Errors compound: an error at one step cascades through subsequent steps, propagating through sequential reasoning Tyen et al. [27]. The longer the reasoning chain, the more opportunity for self-reinforcing mistakes that become invisible within the context that produced them.

Self-evaluation within a deep context may struggle to catch these errors, because the evaluation inherits the same drift that produced the mistake. This creates a tension: the contexts where rigorous evaluation matters most may be the contexts where self-evaluation is least reliable.

Context-separated evaluation may help address this tension. By evaluating outputs in fresh context, without the reasoning trace that produced them, we reduce the inheritance of correlated error. The deeper the original context, the more valuable context separation may become. We stress that whether fresh-context evaluation actually satisfies the conditions for effective external selection (Section 3) is an empirical question; we return to this in Section 6.

A complementary explanation comes from training data composition. Tsui [26] observes that standard training corpora consist overwhelmingly of human demonstrations of correct reasoning, with few examples of a reasoner identifying and correcting their own errors mid-stream. Models may thus lack the behavioral templates for self-correction. They have learned to produce solutions, not to interrupt and revise them. This training gap compounds the context-inheritance problem: models neither detect accumulated errors nor possess trained patterns for correcting them.

1.3. External Selection

If correlated error is the problem, then evaluation in a modified or fresh context (where the original reasoning trace is absent) may provide more independent signal. We call this external selection. The key observation is that "external" refers to *context*, not necessarily to different models. A fresh instance of the same model, without access to the reasoning chain that produced the candidate, may provide more independent critique because the error-producing context is absent.

Multi-agent systems may succeed when they introduce external selection channels Du et al. [7]: formal proof checkers, executable tests, numerical invariants, independently-trained critics, or even the same model under fresh context. The common element is reducing correlation between generator and evaluator failure modes.

This motivates a practical architecture that separates:

- **Generation**: High-entropy exploration of the hypothesis space

[1] We say "may apply" deliberately. Whether these conditions hold for a given deployed model and task is an empirical question, not a consequence of the formalism alone.

 doi:10.20944/preprints202601.0892.v5

- **Selection**: Low-entropy evaluation under external constraints
- **Feedback**: Updating generation based on what survives selection

1.4. Contributions

1. **Repeated self-critique bound** (Lemma 2): Under joint conditional independence of evaluations given the shared failure structure, k rounds of self-critique provide information about correctness bounded by what the shared latent variable Z mediates—not by any independent channel. Confidence accumulated through repeated self-evaluation therefore reflects the shared failure structure rather than independently accumulated evidence. This is the paper's primary formal result.
2. **Sufficient conditions for positive evidence** (Proposition 2): A selector satisfying explicitly measurable bounds on false-acceptance and true-acceptance rates provides a quantifiable lower bound on evidence about correctness. These conditions are empirically testable using existing benchmark infrastructure Kadavath et al. [34], Lin et al. [36].
3. **Information-theoretic scaffolding** (Theorem 1): Under a shared-blind-spot modeling assumption, self-evaluation is bounded in what it can add about correctness. This result is true but its bound is vacuous without an independently constrained latent variable Z; we present it as framing for the above results rather than as an independent contribution.
4. **The falsifiable Z problem**: Identifying the construction of an independently-measurable blind-spot variable as the key open empirical problem the framework requires—and providing concrete candidate approaches
5. **Design principles**: A framework for external selection motivated by the analysis, with explicit acknowledgment that same-model context separation is an engineering heuristic rather than a theoretical solution

1.5. Scope and Claims

What kind of paper this is.

This is a *conditional analysis paper*, not an empirical paper or an impossibility paper. The formal results establish what follows under explicit modeling assumptions; they do not claim those assumptions hold for any specific deployed system, and they do not prove impossibility of reliable self-evaluation in general. The design principles that follow from the analysis are engineering heuristics motivated by the conditional results, not theoretically derived consequences.

When the results apply.

The central negative results depend on the conditional independence assumption $S \perp T \mid (G, Z)$, which captures a specific failure regime: one where the selector's evaluation process is so corrupted by shared blind spots that it gains no independent correctness signal from the input X beyond what Z already mediates. This regime is precisely characterized and not vacuous—Kim et al. [35] study 350+ LLMs overall and find, on one leaderboard dataset, that model pairs agree on the same wrong answer about 60% of the time when both err; shared provider, shared architecture, and higher capability are each associated with higher error correlation, consistent with the high-κ regime the framework analyzes. But it is a *specific* regime, not the universal condition of self-evaluation. The results do not apply when the selector has genuine independent access to correctness information: a proof checker, test executor, or independently-trained model with different failure modes lies outside this regime and may provide substantial evidence.

What the results do not claim.

We do not claim all self-evaluation fails. We do not claim self-evaluation is impossible to improve. We do not claim the proposed architecture provably helps—it is a heuristic starting point. Whether any specific deployed system is in the failure regime the theorems analyze is an empirical question requiring measurement, not a consequence of the formalism.

 doi:10.20944/preprints202601.0892.v5

The design principles we propose are *motivated by* this analysis but are not *derived from* it. Context separation using the same model does not satisfy the formal independence conditions the theory identifies as necessary. We present the architecture as a practical engineering starting point, explicitly disclaim theoretical justification for it, and defer empirical validation to future work.

2. Problem Formalization

2.1. Basic Variables and Setup

Let the input space be $\mathcal{X}$ and hypothesis space $\mathcal{H}$. For an input $X \sim \mathcal{D}$, we define:

Definition 1 (Generator and Selector). *Let $H^\star : \mathcal{X} \to \mathcal{H}$ denote the true (target) hypothesis function. A **generator** produces candidate hypotheses via a conditional distribution $G \sim p(g \mid x)$. A **selector** produces an evaluation score via $S \sim p(s \mid x, g)$. The **acceptance decision** is a deterministic threshold:*

$$A := \mathbb{I}\{S \geq \tau\} \tag{1}$$

Definition 2 (Correctness Indicator). *We define the **correctness indicator**:*

$$T := \mathbb{I}\{G = H^\star(X)\} \tag{2}$$

Thus $T = 1$ when the generator output matches the true hypothesis, and $T = 0$ otherwise.

Definition 3 (Error Events). *The **generator error event** is $E_G := \{T = 0\}$. The **selector error event** is:*

$$E_S := \{A \neq T\} \tag{3}$$

That is, E_S occurs when the selector accepts an incorrect hypothesis ($A = 1, T = 0$) or rejects a correct one ($A = 0, T = 1$).

Definition 4 (Conditional Error Coupling). *The **conditional error coupling** is:*

$$\kappa := \Pr(E_S = 1 \mid E_G = 1) = \Pr(A = 1 \mid T = 0) \tag{4}$$

This is the probability that the selector accepts given that the generator is wrong. Strong coupling ($\kappa \approx 1$) indicates that the selector fails to detect generator errors.

2.2. Shared Blind Spots

To model one mechanism by which errors might correlate, we introduce a latent variable capturing shared failure modes.

Definition 5 (Blind Spot Variable). *Let $Z \in \mathcal{Z}$ be a latent variable indexing input regions where a model family systematically struggles. We say generator and selector **share blind spots** indexed by Z if there exists a set $\mathcal{Z}_{\text{bad}} \subset \mathcal{Z}$ such that:*

$$\Pr(T = 0 \mid Z \in \mathcal{Z}_{\text{bad}}) \geq \alpha_G \gg \Pr(T = 0) \tag{5}$$

$$\Pr(A = 1 \mid T = 0, Z \in \mathcal{Z}_{\text{bad}}) \geq \alpha_S \gg \Pr(A = 1 \mid T = 0) \tag{6}$$

while both failure rates are low on $\mathcal{Z} \setminus \mathcal{Z}_{\text{bad}}$.

Shared blind spots arise naturally when generator and selector share training data, architecture, or context. The variable Z indexes specific failure modes (e.g., particular reasoning patterns, domain gaps, or edge cases) where both components fail together. For the formal results to have practical force, Z should be given an independent, a priori characterization—for example, as a fixed taxonomy of

 doi:10.20944/preprints202601.0892.v5

known failure modes for a given model family—rather than being inferred post-hoc from observed joint failure. Ashury-Tahan et al. [33] provide one concrete instantiation: an empirically-derived taxonomy of 17 error categories derived from analysis of 83 LLMs across 35 datasets, including Logical Reasoning Error, Computation Error, Specification Misinterpretation, and Incomplete Reasoning. Each category is defined independently of any particular generator-selector pair and can serve as an a priori Z specification. The Self-Correction Blind Spot measured by Tsui [26], in which models fail to correct their own errors despite correcting identical errors in external input, is consistent with high α_S in the self-evaluation setting. Kim et al. [35] provide large-scale empirical evidence for the error correlation assumption: studying 350+ LLMs overall, they find on one leaderboard dataset that model pairs agree on the same wrong answer approximately 60% of the time when both err, and separately find that shared provider, shared architecture, and higher capability are each associated with higher error correlation — directly supporting the high κ regime the framework analyzes.

Remark 1 (The modeling assumption $S \perp T \mid (G, Z)$)**.** *Note that the selector S is defined as $S \sim p(s \mid x, g)$: it has access to the input x as well as the generator output g. Since $T = \mathbb{I}\{G = H^\star(X)\}$ also depends on x, the selector could in principle be informative about T by comparing g against the correct answer for the specific x—a channel independent of Z. The conditional independence assumption $S \perp T \mid (G, Z)$ is a* modeling assumption *that captures the specific regime of interest: one where the selector's evaluation process is itself corrupted by the same shared blind spots Z, so that the selector gains no independent signal about correctness by accessing x beyond what Z already mediates. This is precisely the self-evaluation setting, where a model evaluating its own output shares the same failure modes on the same inputs. Selectors that can independently access ground truth (e.g., a proof checker, test executor, or independently-trained model) violate this assumption in the favorable direction, which is exactly why external selection works. Readers should interpret all subsequent results as conditional on this assumption, which characterizes a specific—empirically plausible—failure regime rather than a universal property of self-evaluation.*

3. When Self-Evaluation Fails

3.1. Main Result

The central quantity studied in this section is $I(T; S \mid G)$: the mutual information between correctness T and selector score S, conditional on the generator output G. Conditioning on G rather than on (G, X) is a deliberate modeling choice with substantive implications. Because $T = \mathbb{I}\{G = H^\star(X)\}$ depends on X, a selector with access to X could in principle be informative about T through X alone—a channel independent of the shared blind-spot structure Z (see Remark 1). Conditioning only on G means this paper analyzes the regime where the selector's access to X does not provide an independent correctness signal beyond what Z already mediates. This is the self-evaluation regime of interest: a model evaluating its own output shares the same failure modes on the same inputs, so the X-channel does not rescue evaluation quality. Readers should interpret all results in this section as conditional on that regime. The results do not apply to selectors that genuinely exploit X independently—for example, a proof checker that accesses the problem statement directly and verifies the answer against it.

The formal results below establish: when evaluator error is coupled with generator error via a shared latent structure, self-evaluation becomes non-identifying—agreement provides negligible evidence of correctness.

We formalize this through two theorems: an information-theoretic bound and an evidence bound.

3.2. Information-Theoretic Scaffolding: Bounding What Z Mediates

The central quantity is $I(T; S \mid G)$: the information that the selector provides about correctness, given that we already observe the generator output.

 doi:10.20944/preprints202601.0892.v5

Theorem 1 (Information Bound via Shared Blind Spots)**.** *Let Z be a latent variable. Assume the conditional independence $S \perp T \mid (G, Z)$. Then:*

$$I(T; S \mid G) \leq I(T; Z \mid G) \tag{7}$$

In particular, if $T \perp Z \mid G$, then $I(T; S \mid G) = 0$.

Proof. By the chain rule for conditional mutual information:

$$I(T; S, Z \mid G) = I(T; Z \mid G) + I(T; S \mid G, Z) \tag{8}$$

Under $S \perp T \mid (G, Z)$, we have $I(T; S \mid G, Z) = 0$, hence:

$$I(T; S, Z \mid G) = I(T; Z \mid G) \tag{9}$$

Also by the chain rule:

$$I(T; S, Z \mid G) = I(T; S \mid G) + I(T; Z \mid G, S) \geq I(T; S \mid G) \tag{10}$$

since mutual information is nonnegative. Combining yields $I(T; S \mid G) \leq I(T; Z \mid G)$. If additionally $T \perp Z \mid G$, then $I(T; Z \mid G) = 0$ and the result follows. □ □

Remark 2 (On the informativeness of the bound)**.** *The bound $I(T; S \mid G) \leq I(T; Z \mid G)$ is non-vacuous only when Z is constrained independently of T and S. Because Z is a latent variable, one could in principle always choose Z to encode all residual dependence between S and T, rendering the right-hand side large and the bound trivially satisfied. Theorem 1 therefore establishes* conditions *under which self-evaluation is weak—specifically, when a pre-specified Z characterizes the shared failure structure and $T \perp Z \mid G$ holds—rather than proving that self-evaluation is weak in all realistic systems. Practical application of this theorem requires an a priori characterization of Z (e.g., a fixed taxonomy of known failure modes) that does not depend on the observed outcome.*

Remark 3 (The bound does not imply weakness without a tight, small cap)**.** *The inequality $I(T; S \mid G) \leq I(T; Z \mid G)$ establishes a structural constraint, not a demonstration that self-evaluation is weak. The cap $I(T; Z \mid G)$ could be large — if Z carries substantial information about correctness beyond what G reveals, the bound permits S to be highly informative about T. The theorem implies weakness only when two conditions hold jointly: (1) Z is independently constrained so the bound is non-vacuous, and (2) $I(T; Z \mid G)$ is shown to be small, so the cap is tight. The paper does not independently establish (2) for real LLM systems. The theorem should therefore be read as identifying a structural relationship — information from S is Z-mediated — rather than as proof that self-evaluation is empirically weak. The Lemma 2 and Proposition 2 carry more direct practical content because their conditions are expressible in directly measurable probabilities.*

Interpretation. Under the modeling assumption of Remark 1 and when Z is appropriately constrained (see Remark 2), the information the selector provides about correctness is bounded by how much the blind spot variable Z "knows" about correctness beyond what the generator output already reveals. When Z is a pure nuisance variable (encoding only *how* the system fails, not *whether* it fails), self-evaluation provides zero additional information *within this regime*. This should not be read as a universal claim: selectors that access ground truth independently of Z (e.g., by executing code or consulting a proof checker) fall outside the assumption's scope and may provide substantial information about correctness.

3.3. The Necessity of Falsifiable Z

Remark 2 notes that Z can always be constructed post hoc to satisfy the conditional independence assumption, making the bound in Theorem 1 trivially true. This is not a defect to hide — it is an important structural observation about the framework's scope. Because any observed self-evaluation

 doi:10.20944/preprints202601.0892.v5

failure pattern admits a latent-variable explanation under Theorem 1, that theorem cannot by itself predict or diagnose failure in real systems. It establishes what follows *if* the shared-blind-spot model applies — it does not establish that the model applies.

This shifts the scientific burden precisely: the theorem's contribution is to identify what would need to be measured to determine whether a system is in the failure regime. Specifically, the theorem becomes empirically meaningful only when Z is characterized independently of the failure events being analyzed — defined prior to observation, estimable from separate data, and constrained to a fixed taxonomy rather than inferred post hoc from joint failure.

In contrast, the paper's primary results — the repeated self-critique bound (Lemma 2) and the multi-agent advantage proposition (Proposition 2) — have more direct empirical content. Lemma 2 shows that k evaluations under joint conditional independence provide information about correctness bounded by what Z mediates — not by any independent channel — so accumulating evaluations cannot push information beyond the Z-mediated ceiling; the key question for real systems is whether that joint independence holds, which is testable given a specified Z. Proposition 2 gives conditions a selector must satisfy to provide positive evidence, expressed in terms of directly measurable probabilities δ and ϵ.

For Z to ground Theorem 1 in real systems, candidate approaches include: behavioral clustering of error types across prompts, probing studies of model internals, and difficulty-stratified benchmarks designed to identify systematic failure regions for a given model family. Ashury-Tahan et al. [33]'s 17-category ErrorAtlas taxonomy demonstrates that such a priori specifications are achievable: the taxonomy is derived from held-out model behavior and defines failure categories independently of any particular generator-selector relationship.

For the quantities in Proposition 2 and Theorem 2, measurement infrastructure already exists. Lin et al. [36] introduce CriticBench, which evaluates the same LLM on generation, binary critique (correctness classification), and correction for identical problems. After defining a binary acceptance rule for critique outputs, the critique task can be used to estimate $\Pr(A = 1 \mid T = 0)$ and $\Pr(A = 1 \mid T = 1)$ across 17 models and 5 reasoning domains. Kadavath et al. [34] introduce the P(True) framework, measuring model-generated self-assessments of correctness against known ground truth and reporting calibration curves; after thresholding the continuous confidence scores into binary acceptance events, these curves yield estimates of the conditional probabilities of interest. These existing protocols provide a concrete experimental pathway to testing whether a given deployed system satisfies the proposition's conditions, without requiring new benchmark construction.

Canonical acceptance rules for measurement.

To pre-empt ambiguity in any empirical follow-up: for CriticBench, set $A = 1$ if and only if the model's critique classifies the candidate as correct (using the benchmark's existing binary label); for P(True), set $A = 1$ if and only if $\Pr(\text{True}) \geq \tau$, with τ fixed in advance of observing outcomes (e.g., $\tau = 0.5$). Sensitivity of the resulting (δ, ε) estimates to the choice of τ is a natural robustness check and should be reported alongside primary estimates in any empirical study.

Constructing a Z that is both independently-specified and informative enough to make Theorem 1 non-vacuous remains the key open empirical problem, but the tools for doing so are available in the existing literature.

Remark 4 (Ontological vs. epistemological status of Z)**.** *The framework requires care about two distinct claims involving Z. The* ontological *claim is that some Z exists making the conditional independence $S \perp T \mid (G, Z)$ hold approximately—this is mathematically guaranteed by construction and carries no empirical content on its own. The* epistemological *claim is that such a Z can be independently specified, measured, and constrained prior to observing joint failure. Only the epistemological claim gives the framework practical diagnostic power. The theorems establish what follows* if *an appropriate Z is known; they do not establish that any particular Z is knowable. Practical application requires bridging this gap, which is why we identify the*

 doi:10.20944/preprints202601.0892.v5

falsifiable Z construction as the key open problem rather than treating it as solved by the theoretical existence result.

Lemma 1 (Post-Processing Cannot Increase Evidence)**.** *For any deterministic acceptance rule $A = \mathrm{acc}(S)$:*

$$I(T; A \mid G) \leq I(T; S \mid G) \tag{11}$$

This follows directly from the data processing inequality. The acceptance decision cannot contain more information than the selector score from which it derives.

3.4. Evidence Bound Formulation

Theorem 2 (Bounded Evidence from Acceptance)**.** *Assume the selector has high false acceptance rate:*

$$\Pr(A = 1 \mid T = 0) \geq 1 - \varepsilon \tag{12}$$

and $\Pr(A = 1 \mid T = 1) \leq 1$*. Then the log-likelihood ratio contributed by observing $A = 1$ satisfies:*

$$\log_2 \frac{\Pr(A = 1 \mid T = 1)}{\Pr(A = 1 \mid T = 0)} \leq \log_2 \frac{1}{1 - \varepsilon} \tag{13}$$

Proof. We have $\Pr(A = 1 \mid T = 1) \leq 1$ and $\Pr(A = 1 \mid T = 0) \geq 1 - \varepsilon$, so:

$$\frac{\Pr(A = 1 \mid T = 1)}{\Pr(A = 1 \mid T = 0)} \leq \frac{1}{1 - \varepsilon} \tag{14}$$

Taking $\log_2$ gives the result. □ □

Corollary 1 (Degenerate Evidence for Small ε)**.** *For small ε:*

$$\log_2 \frac{1}{1 - \varepsilon} \approx \frac{\varepsilon}{\ln 2} \approx 1.44\varepsilon \text{ bits} \tag{15}$$

To illustrate the mathematical behavior of the bound: for $\varepsilon = 0.01$ (selector accepts 99% of incorrect hypotheses), acceptance provides at most 0.014 bits of evidence. For $\varepsilon = 0.35$ (a more moderate regime), the bound rises to approximately 0.63 bits. These are mathematical illustrations of how the bound scales with ε, not empirical estimates; determining which regime applies to a given deployed system requires direct measurement of acceptance probabilities.

Important caveat. Theorem 2 provides an *upper* bound on evidence under the assumption of high false acceptance. It does not establish that acceptance is negligible in specific realistic settings unless the false acceptance rate is empirically near 1. The bound applies specifically to the regime where $\Pr(A = 1 \mid T = 0) \geq 1 - \varepsilon$ for small ε; outside this regime, acceptance may convey substantial evidence. Many successful LLM applications violate the preconditions by incorporating external feedback channels (execution, formal verification, retrieval). The negative results apply specifically to the regime where: (1) systematic generator failures exist with nontrivial probability, and (2) the selector shares the generator's blind spots. We do not claim all self-evaluation fails, only that it may fail under these conditions.

Empirical context (non-load-bearing). The theorem's precondition—that false acceptance is elevated in self-evaluation settings—is not established by the formal results but is contextually motivated by available empirical evidence. We note explicitly that the mapping is loose: Tsui [26]'s 64.5% Self-Correction Blind Spot measures "failure to self-correct," which is not operationally identical to the formal acceptance event $A = 1$ (it could reflect abstention, switching to a different wrong answer, or other behaviors). Direct substitution into Theorem 2 would overstate precision. We cite this evidence only to note that the high-false-acceptance regime the theorem analyzes is not implausible, not to

 doi:10.20944/preprints202601.0892.v5

establish that any real system is in that regime. Precise empirical measurement of $\Pr(A = 1 \mid T = 0)$ requires controlled experiments with explicit threshold decisions, which we leave to future work. The formal results stand independent of whether Tsui's findings map onto the theorem's preconditions.

3.5. The Confidence Amplification Problem

Worse than providing no information, correlated self-evaluation can amplify confidence in errors.

Lemma 2 (Repeated Self-Critique Bound)**.** *Consider k selector outputs $S_1, \ldots, S_k$ with acceptance decisions $A_i = \mathbb{I}\{S_i \geq \tau\}$. Assume $(S_1, \ldots, S_k) \perp T \mid (G, Z)$ jointly—which holds, for example, when the S_i are mutually conditionally independent given (G, Z) and each satisfies $S_i \perp T \mid (G, Z)$. Then:*

$$I(T; A_{1:k} \mid G) \leq I(T; Z \mid G) \tag{16}$$

That is, k critiques provide no more information about correctness than the single blind spot variable Z.

Proof. By the chain rule for conditional mutual information:

$$I(T; A_{1:k}, Z \mid G) = I(T; Z \mid G) + I(T; A_{1:k} \mid G, Z) \tag{17}$$

Under the joint conditional independence assumption, $A_{1:k} \perp T \mid (G, Z)$, so $I(T; A_{1:k} \mid G, Z) = 0$, giving $I(T; A_{1:k}, Z \mid G) = I(T; Z \mid G)$. Also by the chain rule:

$$I(T; A_{1:k}, Z \mid G) = I(T; A_{1:k} \mid G) + I(T; Z \mid G, A_{1:k}) \geq I(T; A_{1:k} \mid G) \tag{18}$$

since mutual information is nonnegative. Combining yields $I(T; A_{1:k} \mid G) \leq I(T; Z \mid G)$. □ □

Remark 5. *The joint independence assumption in Lemma 2 is stronger than assuming each $S_i \perp T \mid (G, Z)$ individually; individual conditional independence does not in general imply joint conditional independence. The joint assumption holds naturally when selector evaluations are drawn independently from the same conditional distribution $p(s \mid g, z)$, as in the case of k independent fresh-context evaluations of the same output under the same latent blind spot.*

Proposition 1 (Confidence Amplification)**.** *Under the joint conditional independence assumption of Lemma 2, k rounds of consistent acceptance provide information about correctness bounded by $I(T; Z \mid G)$—not by any independent accumulation. Repeated acceptance cannot push evidence beyond the Z-mediated ceiling regardless of k.*

Proof. This follows directly from Lemma 2: under the joint conditional independence $(S_1, \ldots, S_k) \perp T \mid (G, Z)$, $I(T; A_{1:k} \mid G) \leq I(T; Z \mid G)$. The bound holds for all k; accumulating more evaluations cannot increase information beyond the Z-mediated ceiling. □ □

Intuition: why this creates false confidence.

A naive Bayesian observer treating $A_1, \ldots, A_k$ as independent computes a posterior that grows with each acceptance. Under strong error coupling, evaluations are not independent: if $T = 0$ and $Z \in \mathcal{Z}_{\text{bad}}$, all evaluations fail together ($\Pr(A_1 = \cdots = A_k = 1 \mid T = 0, Z \in \mathcal{Z}_{\text{bad}}) \approx 1$). The formal information in k consistent acceptances is bounded by the single Z-mediated ceiling from the Proposition above. The subjective experience is k "confirmations" but the objective content is at most $I(T; Z \mid G)$—the same ceiling a single evaluation faces. This mismatch between subjective confidence accumulation and bounded objective information is the confidence amplification failure mode.

This may contribute to a failure mode observed in extended LLM reasoning: increasing confidence in coherent, well-argued, wrong conclusions.

 doi:10.20944/preprints202601.0892.v5

3.6. When External Selection Works

The results above identify conditions under which self-evaluation may provide weak evidence. The contrapositive motivates the following design heuristic (not a formally proved corollary, since the sufficient conditions for positive evidence are given by Proposition 2):

External Selection Design Heuristic. Evaluation is more likely to provide substantial information about correctness when:

1. The selector accesses information not contained in (G, Z), breaking the conditional independence $S \perp T \mid (G, Z)$
2. The selector's blind spots $\mathcal{Z}'_{\text{bad}}$ have low overlap with the generator's $\mathcal{Z}_{\text{bad}}$
3. The selector satisfies the conditions of Proposition 2: bounded false-acceptance and true-acceptance exceeding that bound

These are design targets, not a proved characterization of when evaluation succeeds.

External selection channels that satisfy these criteria include: formal verification (accesses mathematical ground truth), executable tests (accesses computational ground truth), different model families (different $\mathcal{Z}_{\text{bad}}$), and fresh-context evaluation (conjectured to partially reduce shared context-level blind spots; see Section 6 for caveats). The empirical results of Tsui [26] are consistent with the general picture: the "Wait" intervention, which injects a correction marker into the model's reasoning trace, reduced the blind spot rate by 89.3% without changing model weights. This is consistent with context perturbation reducing error correlation, though the cited evidence establishes performance improvement, not which formal conditional independence relations changed. The mechanism remains a conjecture.

3.7. Mechanistic Note: A Predictive Interpretation

The information-theoretic results above establish *that* self-evaluation can fail under correlated error. This section offers one interpretation of *why* such correlation may be elevated in language models. This mechanistic account is consistent with the formal analysis but does not depend on it; the information-theoretic bounds hold regardless of the underlying mechanism.

Bender et al. [3] characterize language models as systems that "stitch together sequences of linguistic forms... according to probabilistic information about how they combine, but without any reference to meaning." Under this framing, when asked to evaluate a hypothesis, a language model predicts what evaluative text would likely follow the prompt, given its training distribution. This prediction inherits whatever patterns characterize human evaluation behavior in that distribution: prestige deference, format heuristics, social smoothing, and narrative continuation.

Alignment training (RLHF) shifts *which* human behavior is predicted but may not change the underlying operation. Sharma et al. [24] demonstrate that RLHF-trained models systematically exhibit sycophancy (responses matching user beliefs over truthful ones) and that human preference judgments favor sycophantic responses, creating a training signal toward agreement.

A note on optimization targets.

Standard system prompts optimize for "helpful assistant," not "rigorous evaluator" or "truth-seeker." Zheng et al. [31] observe that commercial systems commonly define the LLM's role this way—ChatGPT's "You are a helpful assistant" is the canonical example—and systematically evaluate how varying personas affects downstream performance. These are not equivalent objectives.

A hypothesis on format consistency.

One observable phenomenon follows from this analysis: *format consistency*. We hypothesize that submitting manuscripts of widely varying quality to major language models with neutral prompts will yield near-uniform response format—balanced positive and negative points regardless of input quality. This hypothesis is stated in falsifiable form to invite experimental follow-up; we do not treat it as an

 doi:10.20944/preprints202601.0892.v5

established result. Readers who wish to test it empirically are encouraged to do so with appropriate controls for prompt variation and model version.

User control through prompting.

The behaviors described above are defaults, not constraints. Extensive research demonstrates that prompt design substantially affects model behavior, with performance differences of up to 76 percentage points from formatting changes alone Sclar et al. [23]. Role prompting shifts reasoning performance dramatically; Kong et al. [13] report accuracy improvements from 53.5% to 63.8% on mathematical reasoning simply by changing the prompt framing. Zhuo et al. [32] introduce sensitivity metrics showing that prompt variations produce substantial and measurable behavioral shifts. If you prompt a model with explicit evaluation criteria ("identify all flaws," "be maximally critical," "act as a hostile reviewer seeking reasons to reject"), it will shift toward that behavior.

Implication for context separation.

When a model generates output under a "helpful assistant" framing and then evaluates that output under the same framing, error correlation risk is structurally elevated: both generation and evaluation are shaped by the same training signal, which satisfies a necessary (though not sufficient) condition for the shared blind spot model of Section 2 to apply. Context separation may help because a fresh context with explicit critic framing resets the prediction target, potentially reducing the shared context-level component of Z.

Importantly, the system prompt in commercial deployments (ChatGPT, Claude, Gemini) is not user-editable. User instructions are layered on top of this hidden foundation. A prompt like "evaluate this critically" operates atop "be helpful," not instead of it. Agentic frameworks built on these APIs inherit the same constraint.

4. Selection Pressure Across Domains

This observation is not specific to AI systems. Selection pressure (a mechanism that determines which configurations persist) appears across biological, physical, and scientific domains. We present these parallels as motivation for treating external selection as a general pattern rather than a domain-specific observation.

4.1. Self-Reference Limitations

Gödel's incompleteness theorems establish that sufficiently powerful formal systems cannot prove their own consistency Gödel [8]. The structural parallel is suggestive: self-reference creates blind spots. A system that generates claims cannot fully validate those claims using only internal resources. Tsui [26] draws an analogous parallel to cognitive science, connecting the LLM self-correction blind spot to the bias blind spot documented by Pronin et al. [22]. Humans reliably identify cognitive biases in others while failing to detect the same biases in themselves.

This mirrors a familiar experience in software engineering: developers cannot effectively QA their own code. The problem is not laziness or lack of intelligence. It is that the developer knows how the code is *supposed* to work and cannot clear that context when testing. The same cognitive patterns that produced the bug prevent recognizing it as a bug.

4.2. Selection Pressure Across Domains

Selection pressure provides the external criterion that distinguishes signal from noise. We observe this structure across engineering, scientific, and reasoning domains:

The pattern is consistent: perturbation generates variation, selection determines what persists, and surviving configurations amplify.

We argue this pattern more closely reflects how human reasoning actually works. A theory is rarely written in a single session. It is returned to with fresh eyes the next morning, reviewed by colleagues with uncorrelated blind spots, revised after a week away from the problem. Each return

 doi:10.20944/preprints202601.0892.v5

provides external selection: the researcher encounters only the output, not the reasoning trace that produced it.

Consider the contrast with extended chain-of-thought in a single context. The model generates a draft, evaluates it, and declares it ready, all while anchored to the reasoning that produced the draft. In our experience, a manuscript declared "ready" in a long context session will be identified as flawed when pasted into a fresh context. This observation motivated the theoretical framework but does not constitute validation of it (Section 6.9).

The engineering cases in Table 1 show this pattern is already standard practice. Fuzzers generate random inputs; programs that survive without crashing have demonstrated robustness. Monte Carlo methods propose random configurations; those satisfying constraints map the viable solution space. In each case, perturbation without selection produces nothing; perturbation with external selection produces progress.

Table 1. Selection Pressure Across Domains.

System	Perturbation	Selection Criterion	Amplification
Engineering			
Fuzzer	Random mutation	No crash	Bug-free path found
Monte Carlo	Stochastic proposal	Satisfies constraints	Solution region mapped
Genetic alg.	Crossover/mutation	Fitness improves	Optimized design
Scientific			
Hypothesis	Conjecture	Persists under test	Theory in literature
Human Reasoning			
Draft	Initial attempt	Survives fresh review	Revised manuscript
LLM (proposed)			
High temp.	Random token	Survives fresh context	Validated output

4.3. External Selection in Practice: Formal Verification

Recent work in theorem proving illustrates external selection concretely. The Prover Agent framework Baba et al. [1] coordinates an informal reasoning LLM with the Lean proof assistant, where Lean provides external verification. Using relatively small language models, the system achieved 88.1% accuracy on the MiniF2F benchmark, outperforming approaches using larger models without external verification.

The mechanism aligns with our analysis. The LLM generates candidate proofs (high-entropy generation). Lean verifies whether the proof compiles, a criterion external to the LLM's training and biases (external selection). The external selection channel reduces correlation between generator error and evaluator error, because Lean's verification depends on mathematical truth, not on patterns in training data.

4.4. Persistence as the Criterion for Reliability

There is a practical equivalence here that bears stating plainly: in any workflow, *what we treat as reliable* is *what survives independent checks*. These are not two separate properties; they are one property described from two directions.

In scientific practice, we can only build on what we can measure and replicate. What persists under repeated, independent measurement is what we call "established." This is not a limitation of our methods; it is the operational definition of reliability.

The context-separated architecture is motivated by this principle. One context's output becomes another context's input for critique, removing the generation trace from the evaluation context. We stress again that this engineering heuristic is not theoretically equivalent to the external selection the formal analysis identifies as necessary: same-model context separation does not break parameter-level correlations. The analogy to independent checks is motivating rather than exact.

 doi:10.20944/preprints202601.0892.v5

5. Multi-Agent Verification

Note on scope: This section discusses multi-agent verification patterns motivated by the formal analysis. The connections drawn are conceptual—the theory identifies what properties an effective external selector must have; the discussion below catalogs mechanisms that plausibly satisfy those properties. None of the claims in this section are formally derived from the theorems.

If self-evaluation fails under correlated error, how can multi-agent systems succeed?

5.1. Breaking Correlation

Multi-agent verification helps when it introduces selectors whose error is less correlated with the generator.

Definition 6 (External criterion)**.** *An external criterion is a selection mechanism depending on information not fully controlled by the generator.*

Proposition 2 (Sufficient conditions for positive evidence)**.** *Suppose a selector satisfies* $\Pr(A = 1 \mid T = 0) \leq 1 - \epsilon$ *and* $\Pr(A = 1 \mid T = 1) \geq \delta$ *for some* $\delta > 1 - \epsilon > 0$*. Then acceptance by that selector provides*

$$\log_2 \frac{\delta}{1 - \epsilon} > 0 \tag{19}$$

bits of evidence about correctness.

Proof. The log-likelihood ratio of $A = 1$ is $\log_2 \frac{\Pr(A=1|T=1)}{\Pr(A=1|T=0)} \geq \log_2 \frac{\delta}{1-\epsilon}$, which is positive whenever $\delta > 1 - \epsilon$. □ □

Remark 6. *The proposition gives* sufficient *conditions under which acceptance provides positive evidence, not necessary and sufficient conditions. Positive evidence requires only* $\Pr(A = 1 \mid T = 1) > \Pr(A = 1 \mid T = 0)$*; the* (δ, ϵ) *parameterization makes the bound explicit and measurable. A selector meeting these conditions provides at least* $\log_2(\delta/(1 - \epsilon))$ *bits; the actual evidence may be larger. The proposition additionally requires the selector to accept correct hypotheses at rate* $\geq \delta > 1 - \epsilon$*, since a trivially rejecting selector would satisfy the false-acceptance bound without being useful.*

The key insight is diversity of failure modes. A selector trained on different data, using different architecture, or implementing formal verification will fail on different inputs than the generator.

5.2. External Selection Channels

Effective external selection channels include:

Formal verification.

Proof assistants (Lean, Coq, Isabelle) and type checkers provide selection under mathematical ground truth. A proof that compiles is verified by mathematics itself, not by a correlated neural network.

Executable verification.

Unit tests, property-based tests, and simulation checks provide selection under computational ground truth. Code that passes tests satisfies constraints external to the generator. This may help explain why LLMs have become effective at coding tasks: the code interpreter provides built-in external selection. The interpreter does not care how confident the model was; the code runs or it throws an error.

 doi:10.20944/preprints202601.0892.v5

Fresh context evaluation as an engineering heuristic.

The same model under fresh context, without the reasoning chain that produced the candidate, may provide more independent critique than same-context self-evaluation. We stress that fresh-context same-model evaluation does *not* satisfy the strict independence conditions of the External Selection Design Heuristic (Section 3): shared weights and training distribution mean parameter-level blind spots remain, so the conditional independence $S \perp T \mid (G, Z)$ is not broken in the required sense. Rather, context separation should be understood as a *pragmatic engineering heuristic* that partially reduces error correlation by removing the generator's reasoning trace from the evaluator's context, without claiming to achieve the theoretical standard. In the absence of a perfect solution, it serves as a low-cost intervention that alters the activation context even when it cannot alter the underlying weights. Whether this partial reduction is sufficient to yield meaningful practical benefit is an empirical question addressed in planned future work (Section 6.9).

Tsui [26] reports that even minimal context perturbation can substantially affect performance: the "Wait" intervention, which injects a single correction marker into the model's own reasoning trace, reduced the Self-Correction Blind Spot by 89.3%. This performance result is consistent with context disruption reducing error correlation, but the evidence establishes the performance effect, not the formal mechanism. Whether the intervention works by altering the shared failure structure Z, by changing the evaluator's prediction target, or through some other mechanism is not established by the cited data and remains a conjecture.

Definition 7 (Context separation)**.** *Two evaluation contexts are* separated *if the evaluating context has no access to: (1) the generation trace (intermediate reasoning steps), (2) the prompt scaffolding (instructions that shaped generation), or (3) hidden state from the generation process.*

Limitations of same-model context separation.

Fresh context reduces context-level error correlation by removing the generator's intermediate reasoning trace and local prompt scaffolding. However, it does not remove correlated failure modes that originate in the model's parameters or training distribution. Same weights means shared inductive biases remain. Context separation is therefore a partial solution: it breaks correlation introduced *during generation*, but not correlation baked into the model itself. For maximum independence, context separation should be combined with model diversity or external tools.

Numerical invariants.

Dimensional analysis, conservation laws, symmetry constraints, and sanity bounds provide selection under physical ground truth. A derivation that violates energy conservation fails regardless of how convincing it sounds.

Retrieval-grounded checking.

Citation verification against a fixed corpus, exact quote attribution, and fact-checking against authoritative sources provide selection under documentary ground truth.

Independent critics.

Models with different training data, different architectures, or different optimization objectives have partially independent failure modes.

5.3. Why Independence Matters

Perfect independence is not required. What matters is that the joint failure probability is lower than individual failure:

$$\Pr(E_{S_1} \cap E_{S_2} \mid E_G) < \Pr(E_{S_1} \mid E_G) \tag{20}$$

 doi:10.20944/preprints202601.0892.v5

Even partially independent selectors compound evidence. This is consistent with why diverse multi-agent panels can outperform single-agent self-evaluation when they reduce joint failure probability Du et al. [7], Liang et al. [17], though the cited results do not independently confirm that error decorrelation is the operative mechanism.

5.4. Persona-Based Diversity

The predictive mechanism described in Section 3.6 suggests a strategy for achieving decorrelated evaluation within a single model: cast the evaluator as different expert types.

When prompted as "a rigorous mathematician checking for proof gaps," the model predicts what such an expert would say. When prompted as "a skeptical physicist checking dimensional consistency," it predicts different behavior. When prompted as "a journal referee looking for reasons to reject," different still.

Each persona predicts a different distribution of expert behavior, with different priorities, different blind spots, and different failure modes. An error that survives the mathematician may not survive the physicist. A claim that passes the physicist may not pass the referee.

This creates epistemic diversity without model diversity. Multiple personas, each on clean context, provide partially independent evaluations that can be aggregated. The decorrelation arises not from different weights but from different prediction targets.

Implementation details (specific persona prompts, aggregation logic, consensus mechanisms) are left to future work. Whether persona-based diversity achieves sufficient decorrelation to satisfy Proposition 2's requirements is an open empirical question.

6. Context-Separated Architecture: Design Principles

The analysis above characterizes *conditions* under which self-evaluation fails and external selection helps. We now describe design principles for an architecture motivated by that analysis. This section is a design proposal, not a validated system: we articulate what properties a workflow should have to address correlated error, and how those properties can be implemented. Whether implementations satisfying these properties perform as expected is an empirical question addressed in planned future work (Section 6.9).

The principles are organized around three functions: high-entropy generation to explore the hypothesis space, external selection to filter candidates under criteria that are not corrupted by the generator's blind spots, and feedback to update generation based on what survives selection.

6.1. Design Goals

1. **Maximize exploration**: Generate diverse candidates without premature filtering
2. **Ensure rigor**: Select only candidates surviving external validation
3. **Enable iteration**: Feed selection results back to improve generation
4. **Preserve human judgment**: Surface candidates for human review, don't replace human decision-making

6.2. Architecture Components

Definition 8 (Generator)**.** *The generator component $\mathcal{G}$ produces candidate hypotheses at high entropy, exploring the space of possibilities without prejudice.*

Definition 9 (Selector)**.** *The selector component $\mathcal{S}$ evaluates candidates against external criteria at low entropy, selecting configurations that survive validation.*

Definition 10 (Feedback)**.** *The Feedback component $\mathcal{B}$ updates generation context based on selection outcomes, biasing future proposals toward surviving structures.*

 doi:10.20944/preprints202601.0892.v5

$$\begin{array}{ll} \mathcal{G} : C \to \{h_1, \ldots, h_k\} & \text{(High-entropy generation)} \\ \mathcal{S} : (h_i, C) \to (s_i, r_i) & \text{(External selection + rationale)} \\ \mathcal{B} : \{(h_i, s_i, r_i)\} \to C' & \text{(Context update)} \end{array} \tag{21}$$

6.3. High-Entropy Generation

The generator component should:

- Operate at high temperature or use explicit diversity objectives
- Generate candidates spanning the hypothesis space, including edge cases
- Avoid premature self-filtering that would narrow the search
- Include alternative assumptions and boundary conditions

Implementation options:

- High-temperature sampling from a single model
- Ensemble sampling from multiple models
- Structured exploration of assumption variations
- Adversarial generation targeting unexplored regions

6.4. External Selection

The selector component should:

- Operate at low temperature for consistency
- Apply explicit checklists and external tools
- Produce structured verdicts with rationales
- Flag uncertainty rather than forcing binary decisions

Selection criteria hierarchy:

1. **Formal**: Does it compile/prove/type-check?
2. **Executable**: Does it pass tests?
3. **Numerical**: Does it satisfy invariants?
4. **Grounded**: Do citations check out?
5. **Adversarial**: Does it survive independent critique?

6.5. Learning from Selection

The feedback component should:

- Add constraints that killed candidates to future prompts
- Preserve successful patterns as templates
- Escalate ambiguous survivors to human review
- Track failure modes for architecture improvement

6.6. Distinguishing from Related Architectures

The key distinction: context-separated evaluation uses persistence under external criteria as the selection signal, not realism (GAN), preference (RLHF), or self-agreement (self-consistency).

Table 2. Comparison with Related Architectures.

Architecture	Selection criterion	Goal	External?
GAN	Discriminator fooled	Realism	No (co-trained)
RLHF	Human preference	Alignment	Partially
Self-consistency	Agreement across samples	Confidence	No (correlated)
Context-separated	External validation	Truth	Yes

 doi:10.20944/preprints202601.0892.v5

6.7. Implementation Sketch

```
GENERATE(context, diversity_target):
    candidates = []
    for i in 1..k:
        hypothesis = generate(context, temperature=HIGH)
        if diversity_check(hypothesis, candidates):
            candidates.append(hypothesis)
    return candidates

SELECT(hypothesis, context):
    verdict = {passed: True, checks: [], rationale: ""}

    # Formal checks
    if has_proof_component(hypothesis):
        proof_result = lean_check(hypothesis.proof)
        verdict.checks.append(("formal", proof_result))
        if not proof_result.success:
            verdict.passed = False

    # Numerical checks
    for invariant in domain_invariants:
        inv_result = check_invariant(hypothesis, invariant)
        verdict.checks.append(("numerical", inv_result))
        if not inv_result.success:
            verdict.passed = False

    # Adversarial checks
    for critic in independent_critics:
        critique = critic.evaluate(hypothesis)
        verdict.checks.append(("adversarial", critique))
        if critique.fatal_flaw:
            verdict.passed = False

    verdict.rationale = synthesize_rationale(verdict.checks)
    return verdict

FEEDBACK(candidates, verdicts, context):
    new_context = context
    for (h, v) in zip(candidates, verdicts):
        if not v.passed:
            new_context.add_constraint(v.rationale)
        else:
            new_context.add_template(h)
    new_context.survivors = [h for (h,v) in zip(...) if v.passed]
    return new_context

MAIN_LOOP(initial_context):
    context = initial_context
    all_survivors = []
    for round in 1..max_rounds:
```

 doi:10.20944/preprints202601.0892.v5


```
        candidates = GENERATE(context, diversity_target)
        verdicts = [SELECT(h, context) for h in candidates]
        context = FEEDBACK(candidates, verdicts, context)
        all_survivors.extend(context.survivors)
        if convergence_criterion(context):
            break
    return prioritize_for_human_review(all_survivors)
```

6.8. Same-Model Implementation via Context Separation

An important observation: this architecture does not require multiple models. The same model under fresh context, without the reasoning chain that produced the candidate, can serve as the selector. The key is context separation, not model separation.

```
CONTEXT_SEPARATED_EVALUATION(problem):
    # Context A: Generate with prediction
    context_a = fresh_context()
    response_a = model(context_a,
        "First predict what you think the answer is, then solve: " + problem)

    # Context B: Steelman AND attack (fresh, no access to reasoning)
    context_b = fresh_context()  # Critical: no shared state
    critique = model(context_b,
        "Here is a proposed solution. Provide both:
         1) Steelman: the strongest case FOR this solution
         2) Attack: the strongest case AGAINST this solution
         Solution: " + response_a.answer_only)

    # Context C: Final verdict (fresh, sees only the problem + critique)
    context_c = fresh_context()  # Separate from both A and B
    judgment = model(context_c,
        "Given this problem and two arguments about a proposed solution,
         which argument is more compelling?
         Problem: " + problem +
        "Steelman: " + critique.steelman +
        "Attack: " + critique.attack)

    return judgment
```

This implementation has several conjectured advantages, pending empirical validation (Section 6.9):

- **Favorable computational tradeoff**: Compared against extended chain-of-thought in a single context, fresh-context evaluation avoids the quadratic attention cost of very long contexts; the total token cost may be lower for problems requiring many reasoning steps. This comparison does not hold against a simple single-pass evaluation.
- **Reduced context-level correlation**: Context B cannot see Context A's reasoning trace, only the output. This removes one potential source of correlated error; parameter-level correlation from shared weights remains.
- **Simultaneous opposition**: Requesting both steelman and attack forces the model to genuinely consider both sides rather than anchoring on one.
- **Temperature control**: High temperature in generation (exploration), low temperature in critique (precision).

 doi:10.20944/preprints202601.0892.v5

6.9. Initial Observations and Future Validation

This section describes qualitative observations that motivated the theoretical framework above. These are not controlled experiments and we explicitly disclaim any empirical conclusions from them. Our goal here is transparency about the origins of this work, not empirical proof.

During development of this methodology, we observed the following patterns when submitting manuscripts to fresh-context instances of commercial language models:

- **Prompt sensitivity**: Minor framing changes (e.g., "thoughts" vs "honest review") produced noticeably different evaluations of identical content.
- **In-context persuasion**: Critics who heard the author's defense sometimes revised harsh assessments to positive ones, suggesting context sharing may correlate evaluator judgment with author framing.
- **Fresh-context disagreement**: Multiple fresh-context evaluations of the same manuscript sometimes disagreed with each other, while self-evaluation produced more consistent (but potentially unreliable) agreement.

These observations are anecdotal and indicative only. They motivated the formal analysis in Sections 2–3 but do not constitute validation of it. No protocols, sampling schemes, or statistical analyses were applied.

Falsifiability and future work.

This analysis makes testable predictions: (1) self-evaluation should show higher error correlation than context-separated evaluation; (2) fresh-context critics should catch errors that in-context self-critique misses; (3) disagreement among independent evaluators should correlate with actual uncertainty about correctness.

Rigorous validation requires controlled experiments across multiple models, systematic variation of prompts and system configurations, and proper statistical methodology. This is beyond the scope of a methodology paper and is left to future work. A companion paper will detail empirical results across model families, prompt variations, and evaluation criteria.

Collaboration invited.

We recognize that experimental design for evaluating LLM self-assessment is methodologically challenging. We welcome collaboration on reducing bias in experimental protocols. Complete conversation logs from our preliminary observations are available on request.

Additional observations.

We also observed evaluation sensitivity to surface features: formatting changes, word choice (e.g., "novel" vs "improved"), and acknowledged prestige bias when we asked models directly whether author reputation would affect their assessment. These patterns are consistent with LLM evaluation inheriting human biases from training data, but we do not claim these as experimental findings.

Negative results.

Context separation does not eliminate the evaluation format described above. When evaluating low-quality work, the balanced positive/negative structure persists; the model fills the expected "positive points" slots with increasingly tenuous or hallucinated content rather than breaking format to deliver an unbalanced negative assessment. Context separation reduces correlated error but does not override the formatting prior.

Stop condition.

In practice, iterative critique terminates when critics begin producing hallucinated criticism: attacks referencing problems not present, repeating addressed points, or focusing on irrelevant details. Detection requires human judgment.

 doi:10.20944/preprints202601.0892.v5

7. Threat Model and Mitigations

To clarify the architecture's robustness, we enumerate failure modes and mitigations.

7.1. Attack Surfaces

Correlated critics.

If critic models share training data, architecture, or optimization objectives with the generator, error correlation persists despite apparent multi-agent structure. This is the primary failure mode the architecture addresses.

Prompt injection on selection.

Adversarial inputs could manipulate critic behavior, causing systematic acceptance of flawed candidates.

Spoofable external checks.

If "external" verification tools can be fooled (e.g., tests that don't actually test the claimed property), the selection signal degrades.

Human confirmation bias.

The human operator may preferentially accept candidates that confirm prior beliefs, reintroducing correlated error at the final selection stage.

Feedback gaming.

If the feedback mechanism is predictable, generators could learn to produce candidates that pass selection without satisfying underlying criteria.

7.2. Mitigations

No mitigation is complete. The goal is defense in depth: multiple independent failure modes such that exploiting one does not compromise the entire system.

Table 3. Threat Mitigations.

Threat	Mitigation
Correlated critics	Model diversity (different families, training data, architectures)
Prompt injection	Structured verdict formats, input sanitization, separate contexts
Spoofable checks	Hard external criteria (formal proofs, physical measurements)
Human bias	Adversarial critics, blind review protocols, explicit checklists
Feedback gaming	Diverse selection criteria, periodic architecture audits

8. Implications

Note on scope: The implications below are motivated by the formal analysis but are not logically derived from it. They represent engineering and design suggestions consistent with the theory's conditional conclusions, not predictions guaranteed by the theorems.

8.1. For AI-Assisted Workflows

The formal analysis identifies conditions under which scaling a single model's self-evaluation may be insufficient for reliable validation in generate-then-judge workflows. Where those conditions apply, the bottleneck is not generation capability but validation under criteria that are genuinely external to the generator's failure structure. Investment in external selection infrastructure (formal verification tools, test generation, invariant libraries, diverse critic ensembles) may yield returns complementary to larger single models in such settings.

 doi:10.20944/preprints202601.0892.v5

8.2. For Reduced Human Oversight

The analysis identifies conditions under which workflows with reduced human oversight may require selection mechanisms with low correlation to generator error. Where those conditions apply, current approaches using self-critique or same-family judge models may not provide the independence required. Reducing human oversight while maintaining reliability in such settings may require:

- Formal verification covering the relevant claim space
- Executable tests providing ground truth
- Evaluation architectures with independently constrained failure modes

In settings where these mechanisms are unavailable, human review remains an important external criterion. Whether a given deployment is in the regime the theory analyzes is itself an empirical question.

8.3. For Human-AI Collaboration

The architecture's goal is not to replace human judgment but to concentrate it. By filtering candidates through external selection, the system surfaces hypotheses worthy of human attention. This addresses a scalability consideration: humans cannot evaluate 200 hypotheses, but they can evaluate 2–3 survivors of rigorous automated filtering.

8.4. Deployment Considerations for Research Applications

Researchers needing adversarial evaluation may benefit from deployments with full prompt control. Commercial APIs constrain the system prompt, layering user instructions atop a fixed "helpful assistant" foundation. Local deployment with customizable system prompts enables research-specific evaluation personas (e.g., "Skeptic," "Hostile Reviewer"). This is not a criticism of commercial design choices, which appropriately prioritize safety and broad utility, but an observation about specialization requirements for research workflows.

8.5. Open Questions

1. What constraints on Z are both empirically tractable and sufficient to make the bounds of Section 3 non-vacuous?
2. Does fresh-context same-model evaluation yield measurable reduction in error correlation in practice, even though it does not satisfy the formal independence conditions?
3. Can formal verification scale to cover more scientific domains?
4. What is the minimum selector diversity required for the conditions of Proposition 2 to hold in practice?
5. Can selection criteria themselves be learned without introducing correlation with the generator's failure structure?

8.6. Future Work

The preliminary observations in Section 6.9 motivated this theoretical work but do not constitute rigorous validation. The most important open problems, in order of priority, are:

- **Operationalizing Z using existing taxonomies**: Apply the ErrorAtlas 17-category taxonomy Ashury-Tahan et al. [33] or the Song et al. two-axis framework Song et al. [37] as candidate a priori Z specifications, then test whether the conditional independence $S \perp T \mid (G, Z)$ holds empirically using existing benchmark data stratified by error category
- **Measuring $\Pr(A = 1 \mid T = 0)$ using existing infrastructure**: CriticBench Lin et al. [36] and the P(True) framework Kadavath et al. [34] already measure binary correctness classification against known ground truth. Running these protocols in self-evaluation mode (same model generates and critiques) vs. external-evaluation mode would directly test the proposition's conditions and measure whether deployed systems are in the failure regime

 doi:10.20944/preprints202601.0892.v5

- **Testing the joint independence assumption**: Using error-stratified benchmarks to measure whether multiple self-evaluation rounds provide incrementally less information than expected under independence, as predicted by Lemma 2
- **Measuring context separation's practical effect**: Controlled experiments comparing self-evaluation to context-separated evaluation, with proper measurement of error correlation using inter-rater reliability metrics

Until such validation is complete, we explicitly disclaim strong empirical conclusions. The contribution of this paper is the theoretical framework and the identification of concrete empirical pathways to testing it; empirical confirmation remains future work.

9. Related Work

LLM self-correction: the empirical foundation.

Huang et al. [11] established empirically that "large language models cannot self-correct reasoning yet," showing that without external feedback, self-correction attempts often fail or degrade performance. Our work offers one possible explanation: correlated error between generator and evaluator can render self-evaluation non-identifying. Their finding that multi-agent critique with same-model copies performed "no better than self-consistency" is consistent with our analysis: identical models share error distributions, so adding copies may not break correlation. Our analysis suggests a fix: context separation or genuinely independent evaluation channels.

The Self-Correction Blind Spot.

Tsui [26] provides important empirical context for the framework. The study demonstrates that LLMs can correct identical errors when presented as external input but fail to correct those same errors in their own outputs, a phenomenon termed the Self-Correction Blind Spot. An average 64.5% failure rate is measured across 14 models on three purpose-built benchmarks (SCLI5, GSM8K-SC, PRM800K-SC). The "Wait" intervention, which injects correction markers into the model's reasoning trace, reduced the blind spot by 89.3%, activating latent capabilities without changing model weights. We treat these findings as empirical context rather than formal validation: the mapping between "failure to self-correct" and the theorem's formal quantities ($\Pr(A = 1 \mid T = 0)$, conditional independence relations) is not established, and the mechanism by which "Wait" produces its effect is not identified by the performance evidence. The findings are consistent with the shared-blind-spot model but do not confirm it. Where we provide theoretical conditions under which self-evaluation may fail, Tsui provides empirical measurement of failure rates and a candidate intervention whose mechanism remains to be characterized.

Self-consistency and majority voting.

Wang et al. [28] demonstrated that sampling multiple reasoning paths and taking the majority answer improves accuracy. This works when errors are uncorrelated across samples. Our analysis clarifies the limitation: if all samples share the same systematic bias (high error correlation), majority voting cannot help. The gains from self-consistency diminish as correlation increases, explaining why the technique works better for some tasks than others.

Multi-agent debate and verification.

Multi-agent systems including AutoGen Wu et al. [29], CAMEL Li et al. [16], MetaGPT Hong et al. [10], and debate frameworks Du et al. [7], Liang et al. [17] explore collaborative reasoning. Chen et al. [4] found that model *diversity* among agents was important for performance gains, consistent with the idea that breaking error correlation matters. We attempt to formalize why multi-agent verification can help (decorrelated failure modes) and suggest that context separation within a single model may achieve similar benefits.

 doi:10.20944/preprints202601.0892.v5

Process supervision.

Lightman et al. [18] showed that supervising each reasoning step (process supervision) significantly outperforms supervising only final answers (outcome supervision). This aligns with our analysis: per-step external feedback breaks the model's "solo reasoning bubble," preventing error accumulation within a single correlated context. Process supervision is an instance of external selection applied during training.

External verification and tool use.

Training separate verifier models Cobbe et al. [5] achieved large gains on math problems. Integration with formal provers Baba et al. [1], Polu and Sutskever [21], Yang et al. [30] provides external selection via mathematical ground truth. The CRITIC framework Gou et al. [9] showed that tool-interactive critiquing improves self-correction. These results are consistent with the view that "external" can mean different models, formal tools, or execution environments.

Apparent counterexamples.

Some work reports successful self-correction: Self-Refine Madaan et al. [19] for iterative text improvement, Reflexion Shinn et al. [25] for agent learning, and Constitutional AI Bai et al. [2] for safety. We reconcile these with our analysis by noting they typically address (a) style and format rather than deep reasoning, (b) cases where oracle feedback is implicitly available, or (c) safety constraints that are well-represented in training data. When Huang et al. [11] removed oracle feedback from self-correction setups, improvements vanished, suggesting that apparent self-correction may often rely on hidden external signals.

Correlated errors and LLM-as-a-Judge biases.

Kim et al. [35] provide the most direct large-scale measurement of the correlated error assumption underlying this framework. Studying 350+ LLMs overall, they find on one leaderboard dataset that model pairs agree on the same wrong answer approximately 60% of the time when both err; shared provider, shared architecture, and higher capability are each separately associated with higher error correlation. This directly instantiates the high-κ regime the framework analyzes, and the authors demonstrate that LLM-as-judge validity is undermined by these correlations. Wataoka et al. [20] find evidence consistent with perplexity familiarity being a central mechanism: LLMs assign higher evaluations to outputs with lower perplexity, and their own outputs naturally have lower perplexity to them. Ye et al. [15] identified 12 major latent biases in LLM-as-a-Judge systems. These empirical findings align with our theoretical analysis: evaluation that shares the generator's distribution will exhibit correlated error.

Ensemble diversity and error decorrelation.

The insight that independent errors enable reliable aggregation is foundational in ensemble learning Krogh and Vedelsby [14]. We apply this insight to LLM self-evaluation, where error correlation may be particularly relevant due to shared training data, weights, and context. We connect ensemble theory to information theory: conditional mutual information $I(T; S \mid G)$ can approach zero under high correlation Cover and Thomas [6], which would explain why self-evaluation becomes uninformative in such cases.

AI for science.

AlphaFold Jumper et al. [12] demonstrates AI solving well-defined scientific problems with clear evaluation criteria. Our focus is the less-structured setting of hypothesis generation and validation, where ground truth is not known in advance and external selection must be actively constructed.

 doi:10.20944/preprints202601.0892.v5

10. Conclusions

The paper's primary result is the repeated self-critique bound: under joint conditional independence of evaluations given the shared failure structure, k rounds of self-critique provide information about correctness bounded by what the shared latent variable Z mediates—not by any independent channel. Confidence accumulated through repeated self-evaluation therefore reflects the shared failure structure rather than independently accumulated evidence. Whether the joint conditional independence assumption holds for real LLM systems is an empirically testable question, not a consequence of the formalism.

The multi-agent advantage proposition gives the constructive counterpart: a selector satisfying explicitly measurable bounds on false-acceptance and true-acceptance rates provides a quantifiable lower bound on the evidence it contributes. These are sufficient conditions, not necessary and sufficient: positive evidence requires only that true-acceptance exceed false-acceptance, but the (δ, ε) parameterization makes the bound explicit and measurable against existing benchmark infrastructure. Both quantities are directly measurable for real systems using protocols like CriticBench Lin et al. [36] and P(True) Kadavath et al. [34].

The information-theoretic bound (Theorem 1) is best understood as scaffolding for these results. It shows that under a shared-blind-spot modeling assumption, self-evaluation is bounded in what it can add — but the bound is vacuous without an independently constrained latent variable Z. We foreground this limitation rather than minimizing it. Constructing a falsifiable, a priori characterization of Z for real LLM systems is the key open problem the framework identifies, and we propose concrete approaches in the future work section.

The design principles we describe are motivated by the analysis but not derived from it. Same-model context separation is an engineering heuristic — it removes the generation trace from the evaluation context without breaking the parameter-level correlations the theory identifies as the source of difficulty. We present it as a practical starting point, with the explicit prediction that it will provide partial benefit, and defer validation of that prediction to empirical future work.

Recent findings by Tsui [26] — that models fail to self-correct at high rates, and that minimal context disruption substantially reduces blind-spot rates — are cited as empirical context, not formal validation. The mapping between those findings and the theorem's formal quantities is not established; confirming that real systems are in the failure regime requires controlled experiments with explicit acceptance decisions, which we leave to future work.

The framework's value is diagnostic: it identifies what to measure to determine whether a system is in the failure regime, and what properties an external selector must have to escape it. The results apply specifically to the regime where the selector's evaluation is mediated by the same shared failure structure as the generator — not to evaluation systems with genuinely independent correctness access. The bottleneck in reliable AI-assisted workflows is often validation rather than generation. The framework makes that bottleneck precise and identifies the conditions under which it can be addressed.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. The theoretical framework does not depend on any proprietary dataset. Empirical validation using existing benchmarks (CriticBench, P(True)) is identified as future work; if conducted, data and analysis scripts will be made publicly available at that time.

Acknowledgments: The author thanks anonymous reviewers for helpful feedback, and Joe Reid for infrastructure buildout and design.

Conflicts of Interest: The author declares no conflicts of interest.

Use of AI Tools: AI tools assisted with drafting and editing. All ideas, methodology, theorems, and technical content are the author's own work.

 doi:10.20944/preprints202601.0892.v5

References

1. Kaito Baba, Chaoran Liu, Shuhei Kurita, and Akiyoshi Sannai. Prover Agent: An agent-based framework for formal mathematical proofs. *arXiv preprint arXiv:2506.19923*, 2025.
2. Yuntao Bai et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.
3. Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 610–623, 2021.
4. Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.
5. Karl Cobbe et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
6. Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.
7. Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.
8. Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.
9. Zhibin Gou et al. CRITIC: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*, 2023.
10. Sirui Hong et al. MetaGPT: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*, 2023.
11. Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024.
12. John Jumper et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021.
13. Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, and Xin Zhou. Better Zero-Shot Reasoning with Role-Play Prompting. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024.
14. Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems (NIPS)*, 1995.
15. Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, et al. Justice or Prejudice? Quantifying Biases in LLM-as-a-Judge. *arXiv preprint arXiv:2410.02736*, 2024.
16. Guohao Li et al. CAMEL: Communicative agents for "mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36, 2023.
17. Tian Liang et al. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023.
18. Hunter Lightman et al. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. (OpenAI; ICLR 2024).
19. Aman Madaan et al. Self-Refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
20. Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. Self-Preference Bias in LLM-as-a-Judge. *arXiv preprint arXiv:2410.21819*, 2024.
21. Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.
22. Emily Pronin, Daniel Y. Lin, and Lee Ross. The bias blind spot: Perceptions of bias in self versus others. *Personality and Social Psychology Bulletin*, 28(3):369–381, 2002.
23. Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I Learned to Start Worrying about Prompt Formatting. *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024.
24. Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, and Ethan Perez. Towards Understanding Sycophancy in Language Models. *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024.
25. Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

 doi:10.20944/preprints202601.0892.v5

26. Ken Tsui. Self-Correction Bench: Uncovering and Addressing the Self-Correction Blind Spot in Large Language Models. *arXiv preprint arXiv:2507.02778*, 2025.
27. Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. LLMs Cannot Find Reasoning Errors, but Can Correct Them Given the Error Location. *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13894–13908, 2024.
28. Xuezhi Wang et al. Self-consistency improves chain of thought reasoning in language models. *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, 2023.
29. Qingyun Wu et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.
30. Kaiyu Yang et al. LeanDojo: Theorem proving with retrieval-augmented language models. *Advances in Neural Information Processing Systems*, 36, 2023.
31. Mingqian Zheng, Jiaxin Pei, Lajanugen Logeswaran, Moontae Lee, and David Jurgens. When "A Helpful Assistant" Is Not Really Helpful: Personas in System Prompts Do Not Improve Performances of Large Language Models. *Findings of the Association for Computational Linguistics: EMNLP 2024*, arXiv:2311.10054.
32. Jingming Zhuo, Songyang Zhang, Xinyu Fang, Haodong Duan, Dahua Lin, and Kai Chen. ProSA: Assessing and Understanding the Prompt Sensitivity of LLMs. *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024.
33. Shir Ashury-Tahan, Yifan Mai, Elron Bandel, Michal Shmueli-Scheuer, and Leshem Choshen. ErrorMap and ErrorAtlas: Charting the Failure Landscape of Large Language Models. *arXiv preprint arXiv:2601.15812*, 2026.
34. Saurav Kadavath et al. Language Models (Mostly) Know What They Know. *arXiv preprint arXiv:2207.05221*, 2022.
35. Elliot Kim, Avi Garg, Kenny Peng, and Nikhil Garg. Correlated Errors in Large Language Models. *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.
36. Zicheng Lin, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. CriticBench: Benchmarking LLMs for Critique-Correct Reasoning. *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.
37. Peiyang Song, Pengrui Han, and Noah Goodman. Large Language Model Reasoning Failures. *Transactions on Machine Learning Research*, 2026. arXiv:2602.06176.