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

Andrew Michael Brilliant *

Posted Date: 27 February 2026

doi: 10.20944/preprints202601.0892.v3

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

 doi:10.20944/preprints202601.0892.v3

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

Recent empirical work shows that large language models struggle to self-correct reasoning without external feedback [11]. We propose a possible explanation: correlated error between generator and evaluator. When both components share failure modes, self-evaluation may provide weak evidence of correctness, and repeated self-critique may amplify confidence without adding information. We formalize this with two information-theoretic bounds. We then describe a practical architecture pairing high-entropy proposal generation with low-entropy external selection. This suggests an alternative to extended chain-of-thought in a single context: separate generation from evaluation using fresh context, restoring the external feedback loop that human reasoning relies on. Importantly, this can be implemented with the same model, reducing error correlation without requiring additional computational cost. The architecture does not replace human judgment; it provides a filter that surfaces candidates surviving external scrutiny for human review.

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 [11]. Tsui [32] 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 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.

 doi:10.20944/preprints202601.0892.v3

This is not a claim about any specific model's limitations. It is a structural property of evaluation systems. A single agent evaluating its own outputs faces correlated error by construction: the same training data, the same inductive biases, the same blind spots. Tsui [32] 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 [24]. 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 can 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.

A complementary explanation comes from training data composition. Tsui [32] 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 [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
- **Selection**: Low-entropy evaluation under external constraints
- **Feedback**: Updating generation based on what survives selection

1.4. Contributions

1. **Information-theoretic bounds**: Formalizing conditions under which self-evaluation may provide weak evidence
2. **Connections to prior work**: A possible explanation for empirical results in self-correction and multi-agent debate

 doi:10.20944/preprints202601.0892.v3

3. **Practical architecture**: A framework for generate-then-judge workflows, including same-model implementations via context separation
4. **Worked examples**: Illustrations of context-separated evaluation

1.5. Scope and Claims

This paper makes a narrow, practical claim: correlated error can make self-evaluation unreliable, and external selection channels can restore reliability. The bounds we present are conditional on explicit assumptions that may not hold in all settings. We do not claim that all self-evaluation fails. We claim the architecture provides a principled filter that can improve the efficiency of human-AI collaboration in settings where the assumptions apply.

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:*

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

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

 doi:10.20944/preprints202601.0892.v3

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 the specific failure modes (e.g., particular reasoning patterns, domain gaps, or edge cases) where both components fail together. The Self-Correction Blind Spot measured by Tsui [32], 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. The selector (same model, same context) fails on the inputs where the generator failed.

3. When Self-Evaluation Fails

3.1. Main Result

The central claim is straightforward: reliable validation benefits from evaluation criteria whose error is not strongly correlated with the generator. When evaluator error is coupled with generator error, 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 Formulation

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.

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. □ □

Interpretation. 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.

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

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

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

 doi:10.20944/preprints202601.0892.v3

3.3. 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*:*

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

For $\varepsilon = 0.01$ *(selector accepts 99% of incorrect hypotheses), acceptance provides at most* 0.014 *bits of evidence, negligible compared to typical prior uncertainty.*

Important caveat. These results are conditional on the stated assumptions. Many successful LLM applications violate these assumptions by incorporating external feedback channels (execution, formal verification, retrieval). When the selector accesses ground truth signals, the conditional independence $S \perp T \mid (G, Z)$ fails in the favorable direction, and self-evaluation can provide substantial information. 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.

3.4. 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* $A_i = \mathbb{I}\{S_i \geq \tau\}$*. If* $S_i \perp T \mid (G, Z)$ *for all* i*, 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 conditional independence assumption, $A_{1:k} \perp T \mid (G, Z)$. The chain rule gives:

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

since the second term is zero. □ □

Proposition 1 (Confidence Amplification)**.** *Under strongly coupled error, repeated self-evaluation that produces consistent acceptance increases subjective confidence while providing no objective evidence.*

 doi:10.20944/preprints202601.0892.v3

Proof. If a system evaluates its hypothesis k times and accepts each time, a naive Bayesian update treats these as independent evidence:

$$\Pr(T = 1 \mid A_1, \ldots, A_k) \propto \Pr(T = 1) \prod_{i=1}^{k} \frac{\Pr(A_i \mid T = 1)}{\Pr(A_i \mid T = 0)} \tag{18}$$

Under strong error coupling, however, $A_1, \ldots, A_k$ are not independent conditional on (T, Z). If $T = 0$ and $Z \in \mathcal{Z}_{\text{bad}}$, all evaluations fail together:

$$\Pr(A_1 = \cdots = A_k = 1 \mid T = 0, Z \in \mathcal{Z}_{\text{bad}}) \approx 1 \tag{19}$$

By Lemma 2, the k acceptances collectively provide no more information than a single evaluation. The apparent evidence of k consistent acceptances is actually a single piece of (non-)information repeated k times. But the subjective experience is k "confirmations," creating false confidence. □ □

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

3.5. When External Selection Works

The results above identify conditions under which self-evaluation may provide weak evidence. The contrapositive suggests when external selection may help:

Corollary 2 (External Selection Criterion)**.** *Evaluation provides 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 false acceptance rate satisfies* $\Pr(A = 1 \mid T = 0) \ll 1$

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 (partially resets Z). The empirical results of Tsui [32] are consistent with this analysis. The "Wait" intervention, which injects a correction marker ("Wait, I think there might be an error") into the model's reasoning trace, reduced the blind spot rate by 89.3% without changing the model's weights. This suggests that even minimal context perturbation can partially break the conditional independence $S \perp T \mid (G, Z)$, consistent with the corollary's prediction that accessing information outside (G, Z) enables more effective evaluation.

3.6. 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 structural in language models.

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. [23] demonstrate that RLHF-trained models systematically exhibit sycophancy (responses matching user beliefs over truthful ones) and that human preference judgments

 doi:10.20944/preprints202601.0892.v3

favor sycophantic responses, creating a training signal toward agreement. This suggests that aligned models predict what a *preferred* human would say, which may favor continuation over contradiction.

A note on optimization targets.

Standard system prompts optimize for "helpful assistant," not "rigorous evaluator" or "truth-seeker." Zheng et al. [29] systematically evaluated social roles in system prompts and found that the "helpful assistant" framing is nearly universal in commercial deployments, yet produces measurably different behavior than alternative framings. These are not equivalent objectives. A helpful assistant that tells a researcher their theory is fundamentally flawed may be accurate but scores poorly on helpfulness. The training signal favors diplomatic balance over harsh accuracy.

Falsifiable prediction.

One observable phenomenon follows from this analysis, which readers can verify directly: *format consistency*. Submit manuscripts of wildly varying quality to any major language model with neutral prompts (e.g., "Tell me your thoughts on this manuscript"). We predict near-uniform response format: balanced positive and negative points regardless of input quality. No human population produces such format consistency on open-ended evaluation tasks. Humans are variable; they have strong reactions, skip sections, write three sentences or three pages depending on mood. The diplomatic balanced structure appearing consistently across queries is itself evidence of optimization toward neutral helpfulness rather than accurate assessment. This prediction is falsifiable. Readers who disagree are invited to test it.

User control through prompting.

Critically, 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 [25]. 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. [30] 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. The diplomatic balanced format emerges from the default "helpful assistant" framing; different framing produces different responses. This is not a limitation but a feature: users control the evaluation stance through prompting. However, the out-of-box configuration with fresh context and neutral prompts yields the default behavior, because that is what the implicit request specified. Expecting rigorous critical evaluation from a system prompted to be a "helpful assistant" is asking for something you did not request.

Implication for context separation.

This returns us to correlated error. When a model generates output under a "helpful assistant" framing, then evaluates that output under the same framing, error correlation is not incidental; it is structurally guaranteed. Both generation and evaluation optimize for the same objective. The problem is compounded by ambiguity in "helpful": for the general public, validation often feels helpful; for researchers, identifying a fatal flaw before publication is the highest form of help. Commercial system prompts optimize for the first definition. Researchers need the second.

When the model evaluates its own output within the same context, it does not automatically shift to adversarial critic. The user's initial prompt also persists: "help me draft this manuscript" combined with the system's "be helpful" creates a trajectory toward supportive collaboration. Asking the same context to then "find the flaws" fights against accumulated framing. The implicit question remains: how would a helpful assistant assess work it just helped produce? The answer favors continuation over correction. Context separation helps because a fresh context with explicit critic framing resets the

 doi:10.20944/preprints202601.0892.v3

prediction target entirely. The goal is simple: better to find fatal flaws in private than to discover them at peer review.

Importantly, the system prompt in commercial deployments (ChatGPT, Claude, Gemini) is not user-editable. Users cannot inspect it, and the models are instructed not to reveal it. User instructions are layered on top of this hidden foundation. A prompt like "evaluate this critically" operates atop "be helpful," not instead of it. Over extended context, the model may drift back toward its base framing. Agentic frameworks built on these APIs inherit the same constraint. From an academic standpoint, relying on undisclosed evaluation criteria is methodologically problematic; the equivalent in peer review would be anonymous reviewers whose instructions and biases are hidden by design.

This interpretation is consistent with the formal analysis but does not depend on it. The information-theoretic bounds hold regardless of the underlying mechanism.

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 [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 [32] draws an analogous parallel to cognitive science, connecting the LLM self-correction blind spot to the bias blind spot documented by Pronin et al. [31]. Humans reliably identify cognitive biases in others while failing to detect the same biases in themselves. The self-referential structure, in which the same system that produces the error attempts to evaluate it, appears to create blind spots across both biological and artificial reasoning systems.

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 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 can repeat indefinitely: fix the issues, return to the original context, declare it ready again, paste into fresh context, find new issues. The fresh context sees what the anchored context cannot.

This is not a claim we can prove formally, but an observation about why context separation may better simulate the iterative, externally-grounded process by which human reasoning converges on reliable conclusions.

 doi:10.20944/preprints202601.0892.v3

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 [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). Errors detected by Lean feed back to the LLM for refinement. 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.

This is consistent with our analysis: external criteria with independent failure modes can enable validation that self-evaluation may struggle to achieve.

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.

This equivalence clarifies why external selection matters. Without independent verification, there is no selection pressure. Without selection pressure, there is no distinction between signal and noise. Without that distinction, no claim becomes "established"; everything remains undifferentiated conjecture.

The context-separated architecture provides what single-context reasoning lacks: an external evaluation that creates selection pressure. One context's output becomes another context's input for critique. The evaluation is external to the generation trace. Signal can be distinguished from noise.

5. Multi-Agent Verification

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

 doi:10.20944/preprints202601.0892.v3

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 (Multi-agent advantage)**.** *If at least one selector has* $\Pr(E_S \mid E_G) < 1 - \epsilon$ *for relevant error classes, then acceptance by that selector provides* $\Omega(\epsilon)$ *bits of evidence about correctness.*

The key 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.

Fresh context evaluation.

The same model under fresh context, without the reasoning chain that produced the candidate, can provide meaningful critique. The error-producing context is absent, so the evaluator cannot "see" the blind spot that caused the error. This is analogous to why a different developer finds bugs that the original author missed: not because of superior skill, but because they lack the mental model that made the bug invisible.

Tsui [32] demonstrates that even partial context separation can be effective. 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%. The marker does not provide new information about the error; it merely interrupts the continuation pattern, creating a minimal form of context separation within a single generation. This suggests that the full fresh-context approach described above may be a strong version of a more general principle: any mechanism that disrupts the evaluator's inheritance of the generator's reasoning trajectory can reduce error correlation.

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 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

 doi:10.20944/preprints202601.0892.v3

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:

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

Even partially independent selectors compound evidence. This explains empirically why diverse multi-agent panels outperform single-agent self-evaluation even when individual agents have similar capability [7,17].

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. The principle is that single-model architectures can achieve meaningful diversity by exploiting the predictive nature of language models rather than fighting it.

6. A Context-Separated Architecture

We now present a practical architecture implementing external 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

 doi:10.20944/preprints202601.0892.v3

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.*

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

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

 doi:10.20944/preprints202601.0892.v3

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

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
```

 doi:10.20944/preprints202601.0892.v3


```
    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:
        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 A: Judge coherence
    judgment = model(context_a,
        "A critic provided steelman and attack arguments.
         Which is more coherent with the problem structure?
         Steelman: " + critique.steelman +
         "Attack: " + critique.attack)

    return judgment
```

This implementation has several advantages:

- **Lower cost**: Fresh context with no history is faster and cheaper than extended chain-of-thought in a single context.

 doi:10.20944/preprints202601.0892.v3

- **Broken correlation**: Context B cannot see Context A's reasoning errors, only the output. The blind spot that caused the error is absent.
- **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).

Empirically, we observe higher disagreement rates between same-model fresh-context critics and same-context self-critique on incorrect solutions. The fresh-context critic catches errors that self-critique misses, consistent with our analysis.

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 strong empirical conclusions. 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 dramatically different evaluations of identical content.
- **In-context persuasion**: Critics who heard the author's defense often 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 frequently disagreed with each other, while self-evaluation produced consistent (but potentially unreliable) agreement.

These observations are indicative, not conclusive. They motivated the formal analysis in Sections 2–3 but do not constitute validation of it.

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 observations as experimental findings.

 doi:10.20944/preprints202601.0892.v3

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. The RLHF-trained structure appears more stable than accuracy. 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.

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.

 doi:10.20944/preprints202601.0892.v3

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

8.1. For AI-Assisted Workflows

This analysis suggests that scaling single models may be insufficient for reliable validation in generate-then-judge workflows. The bottleneck is not generation capability but validation under external constraints. Investment in external selection infrastructure (formal verification tools, test generation, invariant libraries, diverse critic ensembles) may yield returns complementary to larger single models.

8.2. For Reduced Human Oversight

Workflows with reduced human oversight require selection mechanisms with low correlation to generator error. Current approaches using self-critique or same-family judge models may not satisfy this requirement in all settings. Reducing human oversight while maintaining reliability may require:

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

In settings where these mechanisms are unavailable, human review remains an important external criterion.

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 the scalability bottleneck: 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. Can formal verification scale to cover more scientific domains?
2. What is the minimum diversity required for effective multi-agent selection?
3. Can selection criteria themselves be learned without introducing correlation?
4. Do the structural parallels to physics reflect deeper principles?

 doi:10.20944/preprints202601.0892.v3

5. Can commercial API constraints on system prompts be overcome through prompt engineering, or is local deployment necessary for full external selection effectiveness?

8.6. Future Work

The preliminary observations in Section 5.7 motivated this theoretical work but do not constitute rigorous validation. We plan to:

- Conduct controlled experiments comparing self-evaluation to context-separated evaluation across standard benchmarks
- Measure error correlation empirically using established inter-rater reliability metrics
- Test multiple implementations: context separation, persona-based diversity, temperature variation, and multi-model configurations
- Document which approaches work in which settings, rather than advocate for a single method

Until such validation is complete, we explicitly disclaim strong empirical conclusions. The contribution of this paper is the theoretical framework; empirical verification 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. Where Huang et al. documented the phenomenon, we formalize one mechanism by which it can occur. 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 [32] provides what may be the most direct empirical support for our theoretical 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 results suggest this is not a knowledge deficit but an activation failure: the error-correction capability exists but is suppressed during self-evaluation. The "Wait" intervention, which injects correction markers ("Wait," "But," "However") into the model's reasoning trace, reduced the blind spot by 89.3%, activating latent capabilities without changing model weights. The blind spot is attributed to training data composition: standard corpora contain human demonstrations of correct reasoning but lack examples of mid-stream self-correction, so models have not learned the behavioral patterns for interrupting and revising their own outputs. Notably, RL-trained models that encountered correction markers during training did not exhibit the blind spot, further supporting the training data hypothesis. These findings map onto our formalism: the blind spot corresponds to high conditional error coupling κ, the external-vs-self correction gap operationalizes our shared blind spot variable Z, and the "Wait" intervention provides a minimal mechanism for breaking the conditional independence $S \perp T \mid (G, Z)$ that our bounds identify as the source of non-identifying self-evaluation. Where we provide theoretical bounds on when self-evaluation fails, Tsui provides an empirical measurement of how severely it fails and a mechanism for partial remediation.

Self-consistency and majority voting.

Wang et al. [26] demonstrated that sampling multiple reasoning paths and taking the majority answer improves accuracy. This works when errors are uncorrelated across samples: some paths succeed while others fail independently. Our analysis clarifies the limitation: if all samples share

 doi:10.20944/preprints202601.0892.v3

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 [27], CAMEL [16], MetaGPT [10], and debate frameworks [7,17] 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.

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 [5] achieved large gains on math problems. Integration with formal provers [1,21,28] provides external selection via mathematical ground truth. The CRITIC framework [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 [19] for iterative text improvement, Reflexion [22] for agent learning, and Constitutional AI [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.

LLM-as-a-Judge biases.

Recent empirical work has documented systematic biases in LLM self-evaluation. Panickssery et al. [20] found significant self-preference bias: LLMs assign higher scores to outputs with lower perplexity, preferring text more familiar to them. Li et al. [15] identified 12 major latent biases in LLM-as-a-Judge systems, including positional bias and self-enhancement bias. 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 [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 [6], which would explain why self-evaluation becomes uninformative in such cases.

AI for science.

AlphaFold [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.v3

10. Conclusion

In generate-then-judge workflows, evaluation can be low-signal when evaluator errors are correlated with generator errors. We formalized this observation: under a shared failure model, the evaluator score need not add information about correctness beyond what is already implied by the candidate and shared latent structure. If false acceptance is high, acceptance yields a small likelihood ratio and weak Bayesian evidence.

Recent empirical work by Tsui [32] is consistent with the practical severity of this problem. Across 14 models, self-correction fails 64.5% of the time on average, even when models demonstrably possess the knowledge to correct the same errors in external input. The finding that a simple "Wait" intervention reduces this blind spot by 89.3% suggests that even minimal disruption of the generation context can substantially reduce error coupling.

Multi-agent systems and tool-use paradigms can help when they introduce external selection channels with independent failure modes: formal verification, executable tests, numerical invariants, and diverse critics. The proposed architecture operationalizes this insight by separating high-entropy generation from low-entropy external selection. Context separation provides a practical implementation requiring only a single model. The "Wait" intervention of Tsui [32] suggests that even lighter-weight approaches, such as injecting correction markers into reasoning traces, may activate latent self-correction capabilities.

The architecture does not replace human judgment. It provides a filter that concentrates human attention on candidates surviving external scrutiny. In workflows where AI generates many candidates, the bottleneck is often reliable validation rather than generation capability. External selection channels can help address this bottleneck.

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

Funding: This research received no external funding.

Acknowledgments: The author thanks anonymous reviewers for helpful feedback.

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

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.

 doi:10.20944/preprints202601.0892.v3

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 Li 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. Arjun Panickssery et al. 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. 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.
23. 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.
24. 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.
25. 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.
26. 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.
27. Qingyun Wu et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.
28. Kaiyu Yang et al. LeanDojo: Theorem proving with retrieval-augmented language models. *Advances in Neural Information Processing Systems*, 36, 2023.
29. Mingqian Zheng, Jiaxin Pei, and David Jurgens. Is "A Helpful Assistant" the Best Role for Large Language Models? A Systematic Evaluation of Social Roles in System Prompts. *arXiv preprint arXiv:2311.10054*, 2023.
30. 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.
31. 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.
32. Ken Tsui. Self-Correction Bench: Uncovering and Addressing the Self-Correction Blind Spot in Large Language Models. *arXiv preprint arXiv:2507.02778*, 2025.