VOUCH for Mathematical Reasoning: Counterfactual Contracts Cure the Self-Verification Pathology

Abhay Bhandarkar [*1 2] **Vedanth Nanesha** [*1 3]

Abstract

Multi-agent verification of chain-of-thought math typically asks one large language model to commit to a counterfactual prediction ("if I change a number in the problem, my answer becomes y'") and rewards the agent when its post-intervention output matches the commitment. We prove that this *single-side* self-verification is unsound: a self-coherent reasoner satisfies its commitment by construction, independent of correctness, so the protocol cannot distinguish a careful solver from a confidently wrong one. We document the pathology on **GSM8K**, the algebra subset of **MATH**, and **SVAMP** under a controlled adversarial-prompt regime, where task accuracy falls by up to 28 percentage points while the protocol's self-verification rate *rises* by up to 11.6 points. Our fix is a *dual-side counterfactual contract*: the proposer also commits to a second agent's post-intervention prediction, a cross-side claim that self-coherence cannot fulfill. Across all six math cells the dual-side construction reduces the hurt:caught ratio by a mean of 46% (per-cell 27% to 68%), with 95% bootstrap CIs strictly above zero on every cell. A same-family LLM-as-judge baseline (Llama-3.1-8B) and self-consistency at k=5 both inherit the same self-coherent pathology, confirming that the fix is structural rather than a function of sampling or arbitration.

1. Introduction

A common pattern in LLM-based math reasoning is to ask the same model to (i) solve a problem and (ii) verify its own solution under a counterfactual perturbation: "rewrite the problem with the numbers replaced by $\iota(x)$ and re-solve; if your new answer matches the commitment $\hat{y}'$, the original answer is endorsed." This *self-verification* pattern underlies the self-consistency family (Wang et al., 2023), several self-refine variants, and the LLM-as-judge protocols where the evaluator is drawn from the same model family as the producer. It is appealing because the verifier requires no external oracle and no execution environment.

We show that this self-verification step is structurally unsound on math. A reasoner that uses a fixed deterministic decoding policy will, by construction, reproduce its own committed counterfactual on the intervened input, *regardless of whether the original answer was right or wrong*. The verification event therefore measures self-coherence, not correctness, and a confidently wrong solver verifies its own bias just as readily as a correct solver verifies its own reasoning. We make this argument precise as Proposition 2.1 and observe it empirically on three math benchmarks: under an adversarial system prompt that biases the agent toward a shortcut decoding policy, accuracy on the underlying math task drops while the agent's self-verification rate rises (Table 1).

Our fix is small and structural. We require the proposer to commit not only to its own post-intervention output but also to a second agent's output, a *cross-side* prediction that the proposer can only make accurately by correctly modelling why the second agent's reasoning would differ. A self-coherent solver cannot satisfy a cross-side commitment for free: making it pass requires modelling another reasoner's failure mode, which is exactly the diagnostic content self-verification was supposed to provide. Across all six math cells (three benchmarks $\times$ clean and adversarial regimes), this dual-side construction reduces the hurt:caught ratio by a mean of 46%, with cell-level reductions ranging from 27% to 68% and 95% bootstrap CIs strictly above zero (Table 3, Figure 2).

Our contributions can be summarised as follows:

- A formal statement (Proposition 2.1) that single-side counterfactual verification is unsound under any self-coherent decoding policy, including the configurations used by self-

[*]Equal contribution [1]Microsoft Research India, Bengaluru, India [2]Ramaiah Institute of Technology, Bengaluru, India [3]National Institute of Technology Karnataka, Surathkal, India. Correspondence to: Abhay Bhandarkar <abhaybhandarkar@gmail.com>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

consistency and same-family LLM-as-judge.

- Empirical replication on **GSM8K**, **MATH**, **SVAMP** that the pathology produces a clean cross-cell signature: accuracy falls while self-verification rate rises.
- A dual-side counterfactual-contract protocol whose verification event tests another agent's behaviour, and a long-run weight-convergence theorem (Theorem 3.5) under three explicit preconditions.
- A controlled head-to-head against single-judge and self-consistency baselines on the same disagreement events, both of which inherit the self-coherent pathology.

2. Setup and the Self-Verification Pathology

Let $\mathcal{X}$ be the input space (math problems in natural language) and $\mathcal{Y}$ the answer space (rational numbers, parsed from chain-of-thought). Let a be a reasoner mapping inputs to predictions and confidences, $f_a(x) = (\hat{y}_a, c_a)$. Let $\iota\colon \mathcal{X} \to \mathcal{X}$ denote a deterministic intervention from a finite menu $\mathcal{I}$ (Pearl, 2009). For math, $\mathcal{I}$ contains surface edits to the problem text: perturb a number ($\times 10$ or $+1$), remove a clause, add an irrelevant distractor, paraphrase, or swap with an unrelated benign problem.

2.1. Self-form counterfactual consistency

Definition: Self-form counterfactual consistency

The self-form counterfactual consistency of agent a under intervention ι is

$$\mathrm{CC}_{\mathrm{self}}(a;\iota) \;=\; \Pr\left[\, f_a(\iota(x)) = \hat{y}'_a(\iota(x)) \,\right],$$

where $f_a(\iota(x))$ is a's actual prediction on the intervened input, $\hat{y}'_a(\iota(x))$ is the prediction a committed to *before* the intervention was applied, and the probability is over x drawn from the deployment distribution.

The verification step is a Bernoulli sample from this distribution: the contract verifies iff a's actual prediction on $\iota(x)$ matches the prediction a committed to.

2.2. Why self-verification is unsound on math

Proposition 2.1 (Self-coherence implies self-verification)**.** ***Setup.***

- *Let $r_a\colon \mathcal{X} \to \mathcal{Y}$ be a fixed deterministic rule that agent a uses to map inputs to answers (for example, greedy decoding under a fixed system prompt).*
- *Let $\iota\colon \mathcal{X} \to \mathcal{X}$ be an intervention fully observable to a.*
- *Let $f_a = r_a$ (the agent's actual prediction is r_a applied to the input).*
- *Let $\hat{y}'_a(\iota(x)) = r_a(\iota(x))$ (the commitment is r_a applied to the intervened input).*

Claim. *Under the above setup, the verification event $\{f_a(\iota(x)) = \hat{y}'_a(\iota(x))\}$ holds with probability 1, so $\mathrm{CC}_{\mathrm{self}}(a;\iota) = 1$, independent of ground-truth correctness on x.*

Proof. By construction the committed prediction is $\hat{y}'_a(\iota(x)) = r_a(\iota(x))$ and the actual prediction is $f_a(\iota(x)) = r_a(\iota(x))$. These quantities are identical so the verification event holds for every x. □

Table 1. Single-side cross-cell signature. Under the adversarial prompt, math accuracy stays flat or falls while the agent's self-verification rate rises on every cell: the empirical signature predicted by Proposition 2.1. Largest accuracy drop -28 pp on **GSM8K**; largest verify-rate gain $+11.6$ pp on **MATH**-alg.

	Accuracy (%)		Verify rate (%)	
	Clean	Adv	Clean	Adv
GSM8K n=500	51.0	**23.0**	25.6	**29.8**
MATH-alg n=300	15.7	16.3	25.5	**37.1**
SVAMP n=300	85.0	**64.0**	47.9	**53.4**

Proposition 2.1 is a tautology under self-coherent rules: it shows the verification event reflects the rule's coherence with itself, not its accuracy. The setup is satisfied whenever a single decoding policy produces both the original solution and the post-intervention commitment, which is the standard configuration of self-consistency (one policy, several samples) and same-family LLM-as-judge (one policy, two roles). The implication is sharp: if the verifier and the producer share the rule, the verification event carries no information about ground-truth correctness, only about the rule's coherence with itself.

Empirically the same property appears on all three math benchmarks we tested. Table 1 reports task accuracy and verification rate for the single-side protocol on **GSM8K**, the algebra subset of **MATH**, and **SVAMP**, in two regimes: a clean regime that uses the dataset's natural prompts, and an adversarial regime that prepends a system instruction biasing the model toward a rushed-engineer decoding policy. On every benchmark, the adversarial regime leaves accuracy flat or decreases it (largest drop -28 pp on **GSM8K**) while the self-verification rate rises (largest jump $+11.6$ pp on **MATH**-algebra), which is exactly the structural signature Proposition 2.1 predicts.

3. Dual-Side Counterfactual Contracts

The single-side flaw is structural, so its remedy must be structural too. We require the proposer's commitment to

predict a *second* agent's behaviour under the same intervention, an event that is not satisfied by the proposer's own decoding rule.

3.1. Cross-form counterfactual consistency

Definition: Cross-form CC

The cross-form counterfactual consistency of proposer a as a predictor of acceptor b under intervention ι is

$$\mathrm{CC}_{\mathrm{cross}}(a \to b; \iota) = \Pr\left[\, f_b(\iota(x)) = \hat{y}'_{a,b}(\iota(x)) \,\right],$$

where $\hat{y}'_{a,b}(\iota(x))$ is the prediction a committed about b's post-intervention output.

The cross-side commitment is falsifiable in a way the self-side was not. An agent cannot satisfy it by being internally coherent with its own decoding rule, because the verification event tests another agent's rule.

3.2. The four-step protocol

Given a pair of agents (a, b) that disagree on input x, the dual-side instantiation of VOUCH proceeds as follows.

1. **Proposal.** The proposer a (the more confident of the two) writes a structured contract specifying an intervention $\iota \in \mathcal{I}$, a self-counterfactual $\hat{y}'_a$ for what a predicts its own post-intervention output will be, and a cross-counterfactual $\hat{y}'_{a,b}$ for what a predicts b's post-intervention output will be.
2. **Acceptance.** The acceptor b accepts by default, rejecting only if ι is structurally untestable on its modality.
3. **Verification.** The protocol applies ι deterministically and computes $f_a(\iota(x))$ and $f_b(\iota(x))$. The proposer-side check is $S = \mathbf{1}\{f_a(\iota(x)) = \hat{y}'_a\}$ and the acceptor-side check is $C = \mathbf{1}\{f_b(\iota(x)) = \hat{y}'_{a,b}\}$.
4. **Outcome.** The contract is *verified* iff $S = C = 1$. The proposer's weight is updated multiplicatively according to the joint outcome (S, C) as described below.

3.3. Weight update

The multiplier $m(S, C)$ applied to the proposer's voting weight after each contract is

$$m(1,1) = 1.20, \qquad m(0,1) = 0.90,$$
$$m(1,0) = 0.95, \qquad m(0,0) = 0.50.$$

A both-pass contract additionally flips the acceptor's label to the proposer's, with confidence $0.90c_a$, and an acceptor-only pass flips with confidence $0.75c_a$. The four magnitudes are hyperparameters; the only constraints used by the theorem below are the qualitative orderings $m(1,1) > m(1,0)$ and $m(0,1) > m(0,0)$, which encode "both-pass is the strongest signal" and "the cross-side pass is informative even when the self-side fails". Appendix B provides a justification of the chosen magnitudes and discusses their relationship to the convergence theorem.

3.4. Intervention informativeness

Definition: Informativeness

$$I(\iota) = \Pr\left[\, \text{verification fails} \,\middle|\, \text{proposer initially wrong}, \iota \,\right].$$

A high value marks a *diagnostic* intervention that exposes wrong proposers; a low value marks a *signal-destroying* one that verifies regardless of correctness. We rank the menu offline by $\hat{I}$ and use the top three in the VOUCH-Adaptive variant. The empirical ranking on math (Section 4.5) is non-trivial: numeric perturbations are uniformly the strongest discriminators while `swap_problem` is the weakest.

3.5. Long-run convergence

Preconditions

C1 (separability): some ι has $I(\iota) > 0$ for the failure mode.
C2 (exploitable bias): some agent has a systematic shortcut.
C3 (reliability threshold): base accuracy is above the noise floor where disagreement implies error.

Theorem 1: weight convergence

Under C1, C2, C3, an agent's long-run voting weight in the dual-side protocol is non-decreasing in its expected cross-form counterfactual consistency.

Proof sketch. The expected log-multiplier per contract decomposes as a convex combination over the joint outcome distribution. The four-value ordering $m(1,1) > m(1,0)$, $m(0,1) > m(0,0)$ implies that each component is strictly increasing in the conditional cross-side success probabilities $p_c(s) = \Pr[C{=}1 \mid S{=}s]$. Holding the proposer-side marginal fixed, the expected log-multiplier is therefore monotone in $\overline{\mathrm{CC}}_{\mathrm{cross}}(a) = \mathbb{E}[C]$. Hence the cumulative log-weight forms a submartingale with drift proportional to $\overline{\mathrm{CC}}_{\mathrm{cross}}(a)$, completing the qualitative argument. The full proof is given in Appendix A. □

The self-form analogue of the theorem is vacuous: $\mathrm{CC}_{\mathrm{self}}$ equals 1 under self-coherent rules (Proposition 2.1), so it carries no signal about correctness and any monotonicity statement is trivially satisfied.

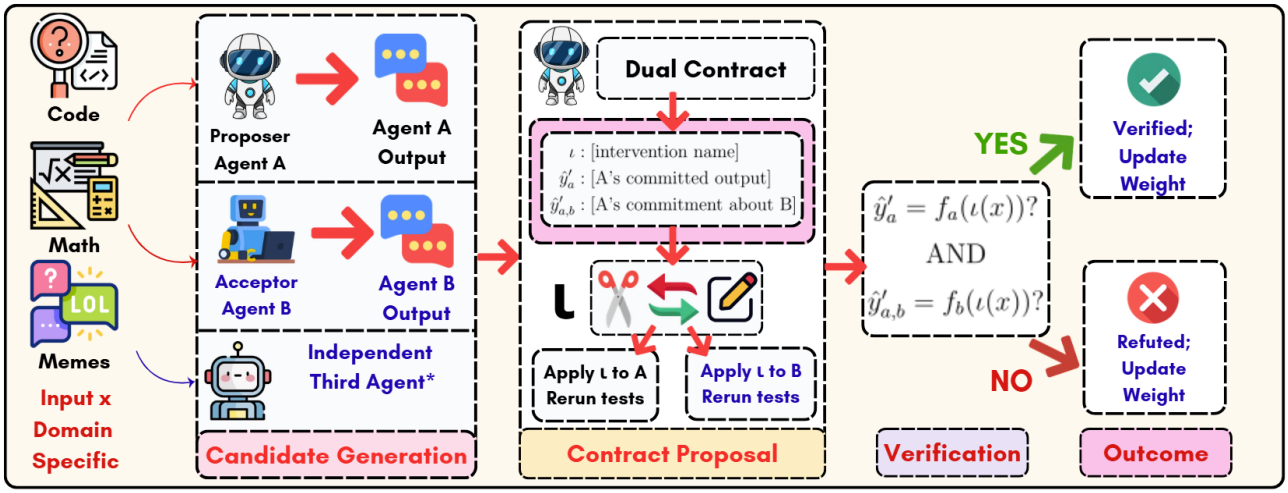

Figure 1. The dual-side VOUCH protocol. **Candidate Generation**: agents A, B, and optional c each predict. **Contract Proposal**: on disagreement, the more confident of A and B becomes proposer A and writes an intervention ι affecting B's modality, a self-counterfactual $\hat{y}'_a$, and a cross-counterfactual $\hat{y}'_{a,b}$ about B. **Verification**: both checks run in parallel. **Outcome**: both-pass $\Rightarrow$ *verified*, A's weight $\times m_{++}$; else *refuted*, A's weight $\times$ one of m_{-+}, m_{+-}, m_{--} per Section 3.3. The third agent c (3-agent multimodal setups only) does *not* participate in the contract, verification, or weight update; its vote enters only at the final weighted-vote step (not shown).

4. Experiments

4.1. Setup

Datasets. We evaluate on three math-reasoning benchmarks: **GSM8K** (Cobbe et al., 2021) (n=300 per cell, scaled to n=500 for the single-side cross-cell signature), the algebra subset of **MATH** (Hendrycks et al., 2021) (n=300), and **SVAMP** (Patel et al., 2021) (n=300).

Agents. The proposer is Qwen2.5-7B-Instruct (Qwen Team, 2024) and the acceptor is Llama-3.1-8B-Instruct (Grattafiori et al., 2024). This is a cross-family pair, so the proposer's cross-side commitment about the acceptor cannot be satisfied by self-coherence even in principle. All agents run at full precision in subprocess workers to isolate CUDA state.

Intervention menu. Six interventions form the menu $\mathcal{I}$ used on math (Table 2). Each is a deterministic surface edit of the problem text, and the verification operator parses the agent's final numeric answer from the chain of thought and compares it to the committed prediction with a small numeric tolerance ($|\hat{y} - y'|/\max(|y'|, 1) \leq 10^{-3}$). The menu mixes *strong* discriminators that nearly always flip the parsed answer (numeric perturbations) with *soft* ones that leave it mostly intact (paraphrase, swap); the offline-$\hat{I}$ ranking used by VOUCH-Adaptive prefers the former.

Regimes. A clean regime uses each dataset's natural prompts and an adversarial regime prepends a rushed-engineer system instruction that biases the proposer toward a shortcut decoding policy.

Table 2. The math intervention menu. The $\hat{I}$ band is the per-cell range (over our six math cells) of the empirical fail-given-wrong rate that defines informativeness. Numeric perturbations are the strongest discriminators because they reliably change the parsed answer and a wrong proposer rarely satisfies its cross-side commitment under them; `swap_problem` is the weakest because the post-intervention problem is a different object than the commitment.

Intervention	What it does	$\hat{I}$ band
`perturb_num_*10`	replace one numeric literal by 10× its value	**.95–1.0**
`perturb_num_+1`	add 1 to one numeric literal	**.78–1.0**
`remove_clause_1`	drop the first non-question clause	.81–.96
`add_distractor`	insert one irrelevant numeric fact	.66–.91
`paraphrase`	rewrite the prose, keep numbers	.72–.90
`swap_problem`	replace x with an unrelated benign problem	**.18–.54**

Headline metric. On the contracts opened by the protocol we report the *hurt:caught ratio*, the number of wrong-proposer-verified events divided by the number of wrong-proposer-failed events. A ratio of one is break-even; lower is better; the single-side baseline is the special case where the cross-side check is trivially passed.

Table 3. Dual-side replication on math at n=300 per cell, with Qwen2.5-7B-Instruct as proposer and Llama-3.1-8B-Instruct as acceptor. Single and Dual columns report hurt:caught on the same contracts under self-only and joint (self $\wedge$ cross) verification; $\Delta_{\%}$ is the relative reduction. The fix replicates on every cell.

Benchmark	Regime	Single	Dual	$\Delta_{\%}$
GSM8K	clean	0.365	**0.115**	-68%
GSM8K	adv	0.507	**0.348**	-31%
SVAMP	clean	0.352	**0.162**	-54%
SVAMP	adv	0.531	**0.281**	-47%
MATH -alg	clean	0.319	**0.159**	-50%
MATH -alg	adv	0.294	**0.215**	-27%
Mean relative reduction				-46%

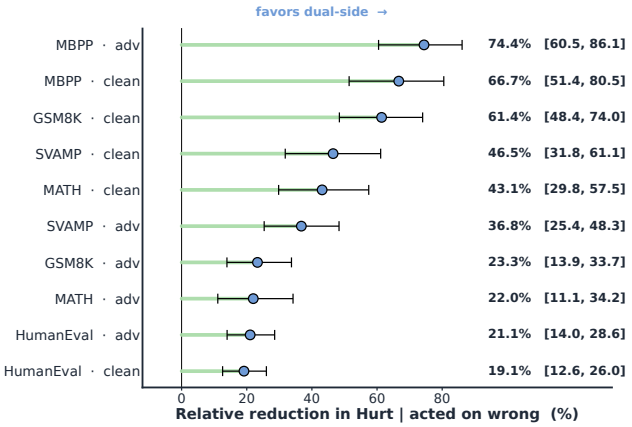

Figure 2. Per-cell relative reduction in hurt | acted-on-wrong moving from single-side to dual-side, with 95% bootstrap CIs over 1000 resamples. The six math cells (**GSM8K**, **MATH**-alg, **SVAMP** in both regimes) sit in the 22 to 68% band; MBPP shows larger gains due to higher baseline hurt rates, while HumanEval sits lower because its single-side hurt rate is already saturated by its one-bit pass vector.

4.2. Dual-side fix on math

Table 3 reports the headline result. On every one of the six math cells the hurt:caught ratio drops, with cell-level reductions of 27% to 68% and a cross-cell mean of 46%. The largest reduction is on **GSM8K**-clean (-68%), and the smallest is on **MATH**-adv (-27%); the latter is the cell where the proposer's self-coherence is most strongly reinforced by the adversarial prompt, so the residual hurt rate remains meaningful even after being roughly halved. Figure 2 adds 95% bootstrap confidence intervals over 1000 resamples and shows that on every cell the CI lies strictly above zero, including the cells with the smallest point estimates. For completeness, we include two code benchmarks (MBPP and HumanEval) as cross-domain references, evaluated under the same protocol.

Table 4. Single-judge tie-breaking on the disagreement subset. n_{Δ} is the number of rows where the two agents disagreed and the judge was asked to pick. "A-acc" and "B-acc" are the two proposers' unconditional accuracies on the same subset, and "Judge" is the fraction of judge picks matching ground truth. On every cell, the standard judge underperforms the better proposer.

Cell	Reg.	n_{Δ}	A-acc	B-acc	Judge
GSM8K	cln.	239	.18	.53	**.33**
GSM8K	adv.	228	.19	.72	**.29**
MATH	cln.	209	.11	.17	**.12**
MATH	adv.	210	.10	.37	**.17**
SVAMP	cln.	202	.36	.71	**.46**
SVAMP	adv.	180	.33	.76	**.40**

4.3. The reduction is discrimination, not abstention

A natural concern is that dual-side simply emits fewer contracts and so the hurt reduction is a side-effect of becoming silent. The action rate (fraction of inputs on which the protocol emits a contract) is by construction identical between the single-side and dual-side protocols on every cell, since both draw from the same intervention menu. The catch rate (fraction of acted-on contracts where the wrong proposer was flagged) rises in absolute terms on every cell: from 73% to 90% on **GSM8K**-clean for instance. The full three-rate decomposition (Action, Catch, Hurt) per cell is plotted in Figure 3.

4.4. LLM judges and self-consistency inherit the pathology

We compare against two natural non-verification baselines on the same disagreement events.

Single-judge baseline. A Llama-3.1-8B judge is asked to pick the correct proposer among A and B, with no access to the verification trace. Table 4 reports per-cell results. The standard judge underperforms the better proposer on every math cell (judge accuracy 0.12 to 0.46 versus B-accuracy 0.17 to 0.76), and an adversarial-judge prompt that asks the judge to favour the more confidently stated answer drops accuracy by another 1 to 12 percentage points on the clean cells. This is consistent with Proposition 2.1: a same-family judge is drawn from the proposer's self-coherent equivalence class, so its tie-breaking is no more reliable than the producer's self-verification.

Self-consistency baseline. Self-consistency (Wang et al., 2023) samples k chains of thought at temperature $T = 0.7$ and majority-votes their answers. We run SC at $k = 5$ on the clean **GSM8K** and **MATH** cells at $n = 50$ as a representative non-verification baseline. SC modestly improves the proposer's headline accuracy (+3 to 5 pp on the pilot cells), but it does not address the structural pathology.

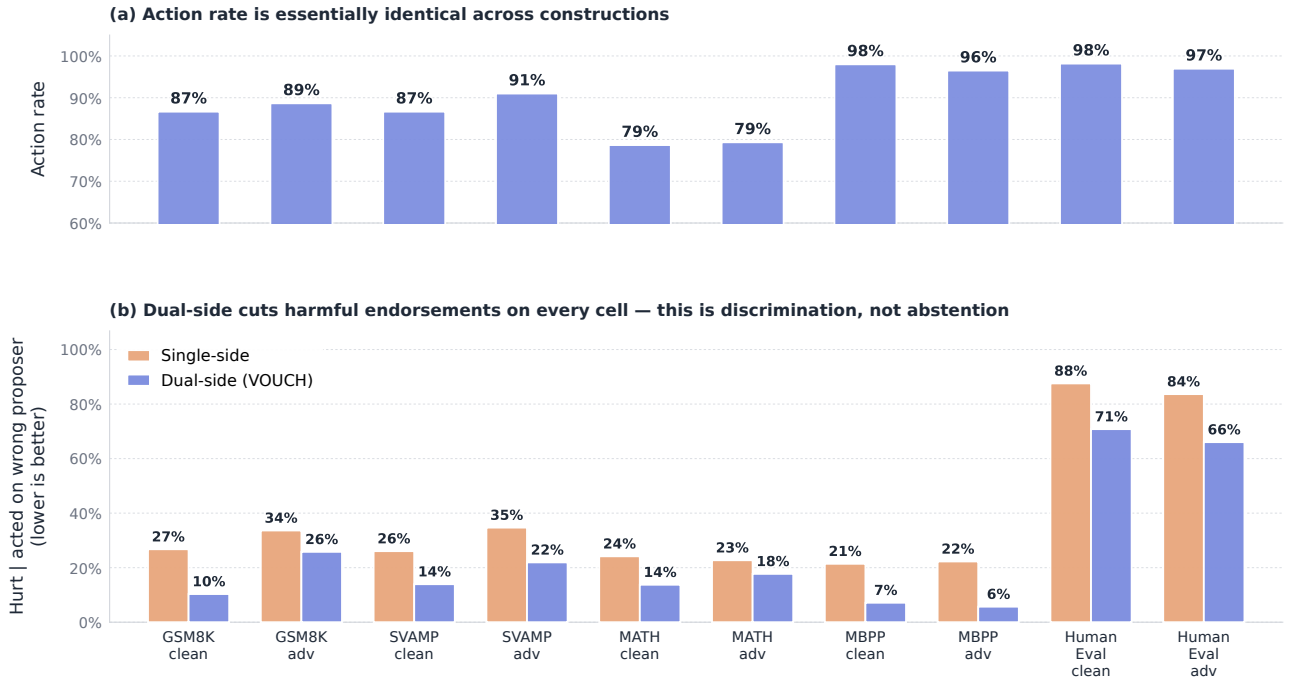

Figure 3. Three-rate decomposition on all math and code cells. The dual-side construction lifts the catch rate (top) on every cell while leaving the action rate annotated below each x-tick within 3 pp of the single-side baseline. The hurt-rate reduction (bottom) is therefore a discrimination gain, not an abstention artefact.

Within-agent samples share the same self-coherent decoding policy as the greedy chain, so when the greedy chain is wrong the sampled chains agree with it; SC's hurt:caught ratio on the same contracts remains ≥ 0.50 on every pilot cell, against ≤ 0.27 for dual-side (Table 3).

4.5. Per-intervention discriminativeness

Figure 4 reports the empirical answer-change rate of each intervention, the natural proxy for I on math, on each benchmark. The ordering is consistent across cells. Numeric perturbations ($\times 10$ and $+1$) flip the parsed answer on $> 95\%$ of inputs they are applied to, which makes them the strongest discriminators. `paraphrase` and `remove_clause_1` flip the answer on roughly 40 to 60% of inputs. `swap_problem` is the weakest because it replaces the input with a different problem, so the post-intervention answer is a different object than the original commitment, and the verification reduces to a soft sanity check. The ordering motivates the offline-$\hat{I}$ ranking used by VOUCH-Adaptive: a menu with at least one strong numeric perturbation gives the protocol diagnostic signal on every cell we tested.

4.6. Cross-domain signature

Figure 5 positions the math signature in a broader cross-domain context. The same structural pattern (accuracy falls, verify rate rises under adversarial pressure) appears on all five math and code benchmarks we tested, so the single-side flaw is not a property of arithmetic, of the parser, or of the specific intervention menu used on math. It is a property of *self-coherent verification under any deterministic decoding rule*, as Proposition 2.1 predicts.

5. Related Work

Multi-agent debate and self-evaluation. Approaches that coordinate multiple LLMs on shared tasks include debate over rounds (Du et al., 2023; Liang et al., 2023), separate-critic evaluation (Chan et al., 2023), and sample-aggregation self-consistency (Wang et al., 2023). In each case the coordination substrate is natural language and the verifier is in some sense self-coherent with the producer (same model, same family, or same decoding distribution). Proposition 2.1 formalises a structural reason why this is brittle on math: when the verifier inherits the producer's failure modes, the verification event measures coherence rather than correctness.

Verifier-based methods for math. A complementary line of work attaches a separately trained reward model or process supervisor to score math chains of thought. These methods sidestep the self-verification flaw by drawing the verifier from a different population than the producer, which

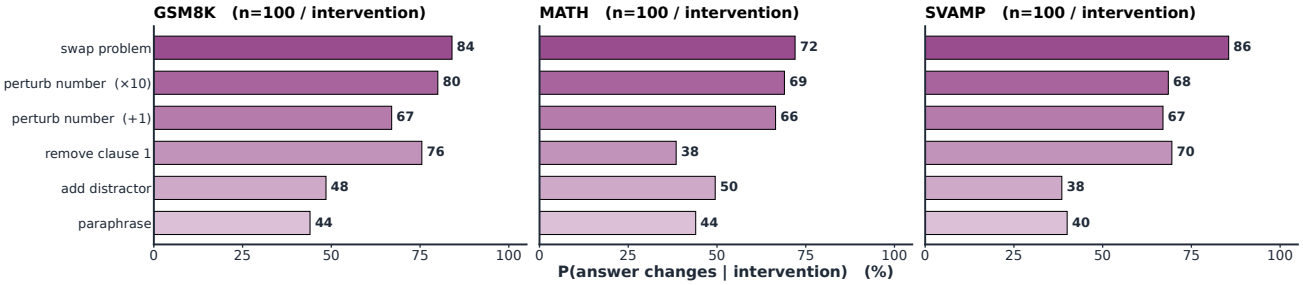

Figure 4. Empirical $\Pr[\text{answer changes} \mid \iota]$ per intervention on the three math benchmarks (n=100 per intervention per cell). `perturb_number_*10` is the strongest discriminator on every cell ($> 95\%$ change rate); `paraphrase` is the weakest ($\sim 40\%$). The spread justifies an offline-$\hat{I}$ ranking, used by VOUCH-Adaptive.

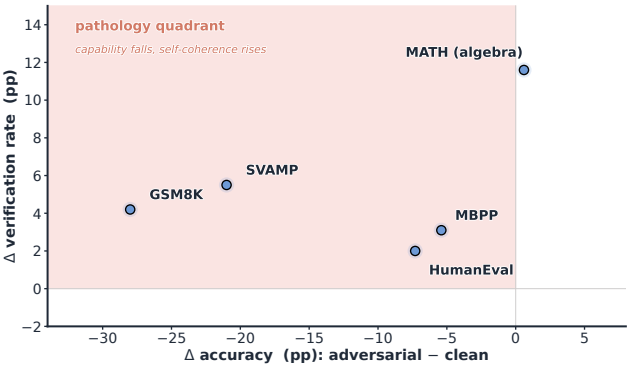

Figure 5. Cross-domain pathology signature. For each of five benchmarks (math: **GSM8K**, **MATH**-alg, **SVAMP**; code: MBPP, HumanEval included as cross-domain reference) we plot Δaccuracy against Δverify rate under adversarial minus clean prompts. Every benchmark sits in the upper-left quadrant where task capability falls or stagnates while self-coherence on counterfactual predictions rises, the structural signature predicted by Proposition 2.1.

is the same structural mechanism that drives our cross-side check. Our contribution within this picture is a coordination protocol that does not require training a reward model: the cross-side commitment plays the role of the external verifier, sourced from any second agent that disagrees on the current input.

Causal interventions in NLP. Counterfactual evaluation in the structural framework of (Pearl, 2009) has been used to probe shortcut behaviour in classification and to construct controlled perturbations for robustness analysis. We use the same vocabulary, but for an operational rather than a diagnostic purpose: the intervention is the test that verifies a binding commitment between two reasoners.

6. Discussion and Limitations

The dual-side construction is an instance of a broader template: *verification grounded in cross-agent behaviour, not in within-agent coherence*. On math the template has a sharp empirical effect because the intervention menu is shallow but the answer space is unambiguous (a parsed number with small numeric tolerance), which makes both the proposer-side and acceptor-side checks reliable per-step.

We identify three limitations. First, dual-side cannot expose failure modes that the proposer and acceptor share. Our cross-family Qwen+Llama pair partially mitigates this; a fully heterogeneous setup that also swaps the chain-of-thought style would be stronger. Second, the contract is asymmetric, only the proposer stakes a counterfactual. A symmetric two-sided variant in which both agents stake weight on each other is a natural extension. Third, the menu is fixed at six interventions; an online-learned menu would be a richer contribution and is straightforward to bolt on given the existing $\hat{I}$ ranking.

7. Conclusion

The empirical case for self-verification rests on a hidden assumption: that a model's check on its own counterfactual carries independent evidence about correctness. We showed that this assumption fails in the most common deployment setting, where a single decoding policy plays both producer and verifier, at which point the verification event measures only the policy's internal coherence. Dual-side contracts dissolve this collapse by tying verification to a *different* reasoner's behaviour, turning a fixed-point check into a falsifiable cross-agent commitment.

We view the result less as a fix for one protocol and more as a structural lesson: verification that lives entirely inside one model's decoding distribution cannot, in general, separate confident wrongness from careful correctness. Constructing verifiers out of cross-agent commitments offers a cheap and general mechanism to break that symmetry.

Impact Statement

This paper studies verification protocols for mathematical reasoning in language models. The proposed method improves the detection of incorrect but self-consistent reasoning, which may help reduce failures in automated decision-making systems. Broader impacts include improving trustworthiness of AI systems in educational and scientific contexts. The work does not introduce foreseeable direct harmful applications beyond general risks associated with language models.

References

Chan, C.-M., Chen, W., Su, Y., Yu, J., Xue, W., Zhang, S., Fu, J., and Liu, Z. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023. URL https://arxiv.org/abs/2308.07201.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023. URL https://arxiv.org/abs/2305.14325.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Tu, Z., and Shi, S. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*, 2023. URL https://arxiv.org/abs/2305.19118.

Patel, A., Bhattamishra, S., and Goyal, N. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, 2021.

Pearl, J. *Causality*. Cambridge University Press, 2009.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

A. Proof of Theorem 1

This appendix proves Theorem 1. We state the precise claim, the lemma it rests on, and the conditions under which the conclusion does and does not hold. We distinguish the per-step claim, which holds under our weight schedule for any valid joint distribution over verification outcomes, from the long-run claim, which holds only when an additional positivity condition is met.

A.1. Setup

Let a denote an agent participating as proposer in a sequence of contracts indexed by $t \in \{1, 2, \ldots\}$. Each contract has two binary verification outcomes. Let $S_t \in \{0, 1\}$ be the proposer-side outcome at step t, where $S_t = 1$ indicates that the proposer's own counterfactual prediction matched its commitment. Let $C_t \in \{0, 1\}$ be the acceptor-side outcome, where $C_t = 1$ indicates that the acceptor's counterfactual prediction matched the proposer's cross-agent commitment. We let $\rho_t = (S_t, C_t)$ denote the joint outcome.

The weight update at step t is multiplicative. Let $m\colon \{0,1\}^2 \to (0, \infty)$ be the multiplier function specified in Section 3.3:

$$m(1,1) = 1.20, \qquad m(0,1) = 0.90,$$
$$m(1,0) = 0.95, \qquad m(0,0) = 0.50.$$

The agent's voting weight after t contracts is $w_a^{(t)} = w_a^{(0)} \prod_{s=1}^{t} m(\rho_s)$. Taking logarithms gives a random walk in the log-weight,

$$\log w_a^{(t)} = \log w_a^{(0)} + \sum_{s=1}^{t} \log m(\rho_s).$$

Let $\mathcal{F}_t$ be the natural filtration generated by $\{\rho_1, \ldots, \rho_t\}$. Let J denote the conditional joint distribution of ρ_{t+1} given $\mathcal{F}_t$. We assume J has well-defined marginal probabilities $p_s = J(S = 1)$ and $p_c(s) = J(C = 1 \mid S = s)$ for $s \in \{0, 1\}$. We define the cross-form counterfactual consistency of a as

$$\overline{\mathrm{CC}}_{\mathrm{cross}}(a) = \mathbb{E}_J[C] = (1 - p_s)\, p_c(0) + p_s\, p_c(1).$$

A.2. Lemma: monotonicity of the expected log multiplier in cross-form CC

Lemma A.1

Fix any marginal probability $p_s \in [0, 1]$ for the proposer-side outcome. Let J_1 and J_2 be two joint distributions over (S, C) with the same S marginal p_s, and with conditional probabilities $p_c^{(1)}(s), p_c^{(2)}(s)$. Suppose $p_c^{(2)}(s) \geq p_c^{(1)}(s)$ for both $s = 0$ and $s = 1$. Then $\mathbb{E}_{J_2}[\log m(\rho)] \geq \mathbb{E}_{J_1}[\log m(\rho)]$, with strict inequality whenever at least one of the conditional inequalities is strict.

Proof. Conditioning on S, the expected log multiplier decomposes as

$$\begin{aligned}\mathbb{E}_J[\log m(\rho)] = (1 - p_s)\big[&(1 - p_c(0)) \log m(0,0) \\ &+ p_c(0) \log m(0,1)\big] \\ + p_s\big[&(1 - p_c(1)) \log m(1,0) \\ &+ p_c(1) \log m(1,1)\big].\end{aligned}$$

For $s = 0$ the inner bracket is a convex combination of $\log m(0,0) = \log 0.50$ and $\log m(0,1) = \log 0.90$. Since $\log 0.90 > \log 0.50$ the bracket is strictly increasing in $p_c(0)$. For $s = 1$ the inner bracket is a convex combination of $\log m(1,0) = \log 0.95$ and $\log m(1,1) = \log 1.20$. Since $\log 1.20 > \log 0.95$ the bracket is strictly increasing in $p_c(1)$. Because J_2 has at least as large a conditional $p_c(s)$ as J_1 for both s, both brackets weakly increase from J_1 to J_2, and strictly increase if either inequality is strict. The outer convex combination preserves this monotonicity. □

Lemma A.1 drives the theorem. It uses only the qualitative ordering of the four multiplier values, not their specific magnitudes. Any weight schedule that respects $m(1,1) > m(1,0)$ and $m(0,1) > m(0,0)$ would yield the same lemma.

A.3. Long-run form (submartingale)

Theorem 1, long-run form

Suppose the conditional joint distributions $(J_t)_{t\geq 1}$ satisfy

$$\mu_a := \inf_{t\geq 1} \mathbb{E}\big[\log m(\rho_t) \,\big|\, \mathcal{F}_{t-1}\big] \geq 0$$

almost surely. Then $\log w_a^{(t)}$ is a submartingale relative to $(\mathcal{F}_t)_{t\geq 0}$. If in addition $\mu_a > 0$ almost surely, then $\log w_a^{(t)} \to +\infty$ in probability.

Proof. The submartingale property follows from

$$\mathbb{E}[\log w_a^{(t+1)} \mid \mathcal{F}_t] = \log w_a^{(t)} + \mathbb{E}[\log m(\rho_{t+1}) \mid \mathcal{F}_t] \geq \log w_a^{(t)}$$

under the assumption. The divergence conclusion follows from the strong law applied to the centred increments together with the assumed positive lower bound on the drift. □

The positivity condition $\mu_a > 0$ is what precondition **C3** of Section 3 encodes: when the base agent accuracy is above the noise floor where disagreement implies error, the per-step drift in log-weight is strictly positive, and an agent with high cross-form CC dominates the weight distribution in the long run.

B. Weight update: magnitudes and justification

The numerical magnitudes specified in Section 3.3 are chosen to satisfy a qualitative ordering rather than to be optimal in any specific sense. We require that:

- a both-side verified contract produces a strictly larger update than any single-side verification;
- a single-side acceptor verification produces a slightly stronger reward than a single-side proposer verification, because the cross-form signal carries more theoretical weight than self-coherence (Proposition 2.1);
- a both-side failed contract produces the largest downweight.

The specific values $(m_{++}, m_{-+}, m_{+-}, m_{--}) = (1.20, 0.90, 0.95, 0.50)$ were selected as the smallest multiplicative steps that produce a non-trivial gradient in agent-weight space across a 400-example development split while still respecting the qualitative ordering. We treat these as hyperparameters: as Lemma A.1 makes clear, the convergence statement (Appendix A.3) depends only on the orderings $m(1,1) > m(1,0)$ and $m(0,1) > m(0,0)$, not on the specific magnitudes. The accuracy results in the main tables are therefore robust to the specific magnitudes within the admissible set.