Two Asymmetries of LLM Self-Review: Family-Dependent Recall Gaps and Same-Model False-Positive Bias

Anonymous authors
Paper under double-blind review

Abstract

We study self-review as an empirical problem in LLM-as-a-judge settings, using a programmatic injection log as ground truth across a 4×4 writer–verifier design (400 technical reports, 286 flawed and 114 clean, 1,600 reviews) and a 9-verifier pilot (50 instances, 4,345 reviews). We report two asymmetries. First, recall gaps are *model-dependent*, not universal: in a logistic regression with cluster-robust standard errors clustered by instance, self-review lowers detection odds overall (log-odds -0.60, $p < 0.0001$), but this effect is model-dependent rather than a uniform property: the GPT-family verifiers consistently show a self-review blind spot (odds ratio 0.55), whereas DeepSeek (OR 1.72) and Google (OR 1.70) verifiers are stricter on their own outputs—yet within Google the direction splits (gemini-2.5 verifiers stricter on self, gemini-3-pro-preview blind), so even vendor family is not a reliable unit. Second, error type is asymmetric: at comparable recall (self 73.9% vs. cross 72.6%, sign test $p = 0.70$), self-review produces substantially more false positives on clean reports (FPR 16.0% vs. 0.0%, two-sided exact McNemar $p = 0.0078$), and a same-model fresh-session condition (FPR 18.0%) is consistent with model self-similarity rather than authorship as the source of this bias. A controlled re-parsing study rules out verdict-extraction artifacts. Finally, a counterbalanced probe holds report content fixed and varies only the authorship label (`self`/`other`/`none`); it finds no reliable effect of the label on recall or false-positive rate (paired McNemar $p = 0.115$ and $p = 0.273$, respectively), indicating that the apparent self-review blind spot is a content-ownership effect rather than a label effect. These results support an empirical, non-causal conclusion: self-review reliability must be validated per model, and verifier selection should consider both missed defects and false alarms rather than treating model separation as a proxy for quality.

1 Introduction

Large language models are increasingly used both to generate outputs and to evaluate, critique, and revise them. This practice underlies self-correction pipelines Welleck et al. (2023); Madaan et al. (2023); Shinn et al. (2023), model-based evaluation Zheng et al. (2023); Liu et al. (2023), process supervision Cobbe et al. (2021); Lightman et al. (2024), and multi-agent systems in which one model checks another Du et al. (2024); Wu et al. (2023); Chen et al. (2024a). A recurring design intuition is that the evaluator should be independent of the generator: a model may fail to recognize its own errors, whereas a different model may bring a less correlated perspective.

The intuition is plausible, but it does not by itself establish that cross-model verification is effective. Independence concerns the relationship between writer and verifier; it does not guarantee that the verifier is competent, calibrated, or complementary to the writer. A weak but independent verifier may miss nearly all defects. An aggressive verifier may obtain high sensitivity by labeling many clean outputs as flawed. Conversely, a capable self-verifier may outperform an independent model despite sharing the writer's architecture and tendencies.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The practical question is therefore not simply whether a verifier is independent, but which properties make the verifier useful.

We study this question in a complete 4×4 writer–verifier design. Four LLMs—`deepseek-chat`, `gpt-5.4`, `gpt-5.6-sol`, and `gemini-2.5-pro`—serve as both writers and verifiers. The benchmark comprises 400 technical reports with ground-truth binary labels: 286 flawed reports and 114 clean reports. Each writer contributes 100 reports, and every report is reviewed by each of the four verifiers. The resulting design contains 16 writer–verifier pairs and 1,600 reviews. Diagonal cells represent self-review, while off-diagonal cells represent cross-model review.

This design separates three properties that are often conflated:

1. **Independence**: whether the verifier differs from the writer;
2. **Verifier competence**: the verifier's ability to distinguish flawed from clean reports;
3. **Complementarity**: whether a particular verifier is effective for outputs from a particular writer.

The aggregate results do not support independence as a sufficient condition for verification quality. Cross-model sensitivity is 32.4%, compared with 31.5% for self-review, an improvement of only 0.9 percentage points. A controlled re-parsing study confirms this aggregate is robust to verdict parsing: although `gpt-5.6-sol` emits reasoning prefixes on 94.5% of reviews and `gpt-5.4` on 26.0%, strict final-token parsing and the original substring parser differ by less than 0.3 percentage points, so the aggregate comparison is not an artifact of verdict extraction.

Verifier identity matters much more descriptively. Pooled across all four writers, verifier sensitivity ranges from 4.2% for `deepseek-chat` to 67.5% for `gemini-2.5-pro`, a range of 63.3 percentage points and an endpoint ratio of approximately 16.1. However, `gemini-2.5-pro`'s higher sensitivity is accompanied by a 45.6% false-positive rate, demonstrating that defect recall alone is not sufficient to characterize verification quality.

Finally, independence has no uniform direction across verifiers. For `gpt-5.4`, sensitivity rises from 13.1% in self-review to 30.2% in cross-review. For `gpt-5.6-sol`, it rises from 9.4% to 36.5%. Yet for `gemini-2.5-pro`, the direction reverses sharply: sensitivity falls from 86.4% in self-review to 59.1% in cross-review. These reversals indicate that "self versus cross" is not a stable main effect; it depends on verifier behavior and the writer–verifier pairing.

We contribute:

- a fully crossed evaluation of four writer models and four verifier models over 400 ground-truth-labeled technical reports and 1,600 reviews, plus a 9-verifier family-level pilot (4,345 reviews);
- a cluster-robust logistic-regression estimate showing that the self-review recall gap is model-dependent (a robust GPT-family blind spot, with DeepSeek and Google verifiers stricter on their own outputs, and within-Google direction splitting);
- a matched three-condition probe showing that self-review produces more false positives than cross-review at comparable recall; and
- evidence that verifier selection should consider both sensitivity and specificity rather than treating model separation as a proxy for quality.

2 Related Work

2.1 Self-Correction and Self-Critique

A substantial literature asks whether LLMs can identify and repair their own mistakes. Self-Refine uses iterative feedback and refinement from a single model Madaan et al. (2023), while Reflexion maintains linguistic feedback across attempts Shinn et al. (2023). Welleck et al. Welleck et al. (2023) formulate sequence generation as learned self-correction, and

CRITIC augments critique with external tools Gou et al. (2024). Constitutional AI similarly uses model-generated critiques and revisions as part of an alignment procedure Bai et al. (2022). Saunders et al. Saunders et al. (2022) investigate self-critiquing models for scalable oversight, and self-knowledge work Kadavath et al. (2022) asks whether models know what they know.

The evidence for unaided self-correction is mixed. Huang et al. Huang et al. (2024) find that LLMs may fail to improve reasoning without external feedback and can alter initially correct answers. SelfCheckGPT instead exploits disagreement among sampled generations to identify possible hallucinations Manakul et al. (2023). These findings motivate independent checking, but do not imply that any independent verifier will be accurate.

Our study crosses writer and verifier identities, so self-review can be compared against cross-model review while observing differences among verifier models.

2.2 LLMs as Evaluators

LLM-based evaluation has become common when human evaluation is costly. G-Eval uses chain-of-thought prompting and form-filling for natural-language generation evaluation Liu et al. (2023). Zheng et al. Zheng et al. (2023) study strong LLMs as judges of chatbot responses and identify biases such as position and verbosity effects. AlpacaFarm develops a simulation framework for preference-based evaluation and alignment Dubois et al. (2023), while HELM emphasizes broad, standardized evaluation across scenarios and metrics Liang et al. (2023). Wang et al. Wang et al. (2024) further show that LLM evaluators need not judge candidate systems fairly, and Koo et al. Koo et al. (2024) benchmark cognitive biases in LLM evaluators; calibration analysis Guo et al. (2017) is a related concern.

We extend this caution to the self-versus-cross distinction: a verifier can be independent yet insensitive, or sensitive yet insufficiently specific.

2.3 Verifiers, Debate, and Multi-Agent Systems

Learned verifiers have improved performance on mathematical reasoning tasks Cobbe et al. (2021); Weng et al. (2023). Process supervision evaluates intermediate reasoning rather than only final answers Uesato et al. (2022); Lightman et al. (2024); Dhuliawala et al. (2024). Debate and multi-agent methods seek to expose errors through interaction among multiple agents Irving et al. (2018); Du et al. (2024); Khan et al. (2024); Liang et al. (2024); Cohen et al. (2023). General frameworks such as CAMEL, AutoGen, and AgentVerse make it possible to compose role-specialized LLM agents Li et al. (2023); Wu et al. (2023); Chen et al. (2024a). ReConcile uses discussion among diverse agents to improve reasoning Chen et al. (2024b).

But "multiple" and "independent" do not automatically mean "accurate": agents may share recall gaps, differ in response thresholds, or add noisy criticism. We evaluate this directly by comparing diagonal and off-diagonal writer–verifier combinations rather than assuming a generic benefit from diversity.

3 Method

3.1 Benchmark and Ground Truth

We evaluate verifier behavior on a benchmark of 400 technical reports, consisting of 286 flawed reports and 114 clean reports. The labels are obtained from a programmatic injection log that is independent of all models used in the study: flawed reports are produced by injecting one or more defects into an otherwise clean report, and the injection log records each defect and its type. We use three defect types: *numeric contradiction* (a number in the report is changed, e.g. 1200 → 12000), *citation mismatch* (a bibliographic reference is renumbered to point at the wrong entry), and *logical inconsistency* (a sentence is altered to contradict an earlier statement, e.g. "meets" → "does not meet"). Clean reports are verified to contain no injected defects by construction. This design makes the ground truth exogenous to both the writer and verifier models, and allows us to ask whether a verifier

detects injected flaws differently when it reviews its own outputs versus outputs produced by other models.

The benchmark is evaluated under a 4×4 writer-verifier cross design: each report is reviewed by each of the four verifier models, yielding 1,600 total reviews. Because every report is judged by all verifiers, the same underlying item can be compared across self-review and cross-review conditions while holding the report content fixed.

3.2 Verifier Models

We study four models as verifiers: `deepseek-chat`, `gpt-5.4`, `gpt-5.6-sol`, and `gemini-2.5-pro`.

For the main analysis, the same four models also serve as writers, producing the 4×4 cross-product of writer-verifier pairs. We distinguish:

- *Self-review*: the writer and verifier are the same model.
- *Cross-review*: the writer and verifier are different models.

This distinction is central to testing whether self-review sensitivity gaps are universal or model-dependent.

3.3 Prompts and Verdict Extraction

Each verifier is prompted to output a single-token verdict, either `CLEAN` or `FLAWED`. To keep the judgment interface consistent across models, we parse the final verdict from the model response using a substring-based extraction rule in the main pipeline.

Because some models may prepend internal reasoning text before the final verdict, we treat verdict extraction as a potential source of artifact and examine it directly in a controlled re-parsing study. The goal is to isolate whether any apparent self-review effect is due to parser behavior rather than the model's underlying judgment.

3.4 Controlled Re-parsing Study

To test whether verdict patterns depend on parsing rather than on the substantive model judgment, we rerun the exact same 1,600 raw verifier responses under two parsers:

1. a *strict* parser that strips any `think`-style prefix and reads the final verdict token;
2. the original *substring* parser used in the main pipeline.

The only variable in this study is the parser. All prompts, raw outputs, reports, and verifier identities are held fixed. This lets us test whether the main self-vs-cross finding survives a stricter extraction rule, and whether chain-of-thought-like prefixes contaminate the final verdict.

3.5 Metrics

We report the standard binary classification metrics on the verdicts:

- **Sensitivity** (recall on flawed reports): the fraction of flawed reports correctly flagged as `FLAWED`.
- **Specificity**: the fraction of clean reports correctly flagged as `CLEAN`.
- **Matthews correlation coefficient (MCC)**: a class-imbalance-aware summary of verdict quality.

Sensitivity is the primary metric in the main analysis because the benchmark is imbalanced (286 flawed vs. 114 clean), and the key question is whether verifiers miss injected flaws in self-review versus cross-review conditions.

3.6 Statistical Analysis

Our primary comparison is within-verifier self-review sensitivity versus cross-review sensitivity. We report both the aggregate pattern across all verifiers and model-specific patterns for each verifier, and we compute MCC from the corresponding confusion matrices to ensure conclusions are not driven by class imbalance. To attach inference to the self-versus-cross comparison, we fit a logistic regression with cluster-robust standard errors clustered by instance (Section 4.4), and report family-level odds ratios for the self-review effect. For the matched false-positive probe (Section 4.5), we use the exact sign test for paired recall and the exact McNemar test for paired false-positive counts.

The main claims are based on observed associations in the benchmark. We do not infer causality from these comparisons. In particular, any self-review effect should be read as a conditional pattern in this dataset, not as proof that model identity alone determines the sensitivity gap.

3.7 Scope of Claims

Our goal is to test three linked claims:

1. Self-review recall gaps are *model-dependent*, not universal.
2. Self-review and cross-review differ in error *type* (false positives) at comparable recall.
3. The observed pattern is not an artifact of verdict parsing.

Accordingly, the writer–verifier experiment and the cluster-robust regression model establish whether the self-review recall effect differs by model family, the matched three-condition probe establishes the false-positive asymmetry, and the controlled re-parsing study checks whether the results persist when the parser is changed but everything else is held fixed.

4 Results

4.1 Aggregate Self-Review Versus Cross-Review Sensitivity

Across the full 4×4 benchmark, self-review sensitivity is 31.5%, while cross-review sensitivity is 32.4%. The aggregate difference is only +0.9 percentage points in favor of cross-review.

This near-parity is important, but it is also misleading if read alone: the aggregate average masks strong model-specific heterogeneity. The self-review effect is therefore not a single global property of "LLM judges"; it depends on which model is doing the judging.

Table 3 reveals the pairing-level structure behind the aggregate. The `gemini-2.5-pro` verifier achieves the highest sensitivity on every writer column, while the `deepseek-chat` verifier is near floor everywhere; most writer–verifier combinations lie in between, confirming that verifier quality is not reducible to a single self-versus-cross axis.

Table 2 reports the full per-verifier classification profile. The key observation is that `gemini-2.5-pro`'s high sensitivity does not translate into superior overall quality: its 45.6% false-positive rate and low specificity (54.4%) leave its MCC (0.203) essentially tied with `gpt-5.6-sol` (0.198), which achieves a far better sensitivity–specificity balance. This reinforces the central point that sensitivity alone is an insufficient selection criterion.

Table 1 shows that the direction of the self-vs-cross gap reverses across models:

- `gemini-2.5-pro` is substantially *more sensitive* in self-review than in cross-review, by 27.3 percentage points.
- `gpt-5.4` is less sensitive in self-review than in cross-review.
- `gpt-5.6-sol` shows the same direction as `gpt-5.4`, with a larger gap.
- `deepseek-chat` is near floor in both settings.

Table 1: Self-review versus cross-review sensitivity in the 4×4 writer-verifier benchmark.

Verifier	Self-review sensitivity (%)	Cross-review sensitivity (%)
`deepseek-chat`	0.0 (0/73)	5.6 (12/213)
`gpt-5.4`	13.1 (8/61)	30.2 (68/225)
`gpt-5.6-sol`	9.4 (6/64)	36.5 (81/222)
`gemini-2.5-pro`	86.4 (76/88)	59.1 (117/198)
Aggregate	31.5 (90/286)	32.4 (278/858)

Table 2: Per-verifier classification quality, pooled across all four writers. Sensitivity is computed on 286 flawed reports and FPR/specificity on 114 clean reports (pooled across writers); MCC and balanced accuracy summarize the full confusion matrix (Table 3).

Verifier	Sensitivity (%)	FPR (%)	Specificity (%)	MCC	Balanced Acc (%)
`deepseek-chat`	4.2	0.9	99.1	0.084	51.7
`gpt-5.4`	26.6	12.3	87.7	0.155	57.1
`gpt-5.6-sol`	30.4	11.4	88.6	0.198	59.5
`gemini-2.5-pro`	67.5	45.6	54.4	0.203	60.9

This pattern directly supports the first claim: the self-review sensitivity gap is model-dependent. There is no single aggregate self-review penalty that applies uniformly across verifiers. Instead, the benchmark reveals qualitatively different regimes, including one model with a large self-review advantage and others with a self-review disadvantage.

4.2 Controlled Re-parsing Rules Out a Parsing Artifact

A potential confound is that some models emit `think`-style prefixes before their final verdict, which could interact with the verdict parser. We therefore reran all 1,600 raw responses under a strict parser and compared the resulting verdicts against the original substring parser.

Table 4 shows the observed prefix incidence in the raw outputs: `deepseek-chat` and `gemini-2.5-pro` never produced a `think` prefix in this study, while `gpt-5.4` did so in 26.0% of responses and `gpt-5.6-sol` in 94.5%.

Despite this wide variation in prefix behavior, the strict parser and the substring parser produce verdict rates that differ by less than 0.3 percentage points. This is the key identification result of the controlled study: the main verdict pattern is stable to parser choice.

The implication is that the model-dependent self-review effect observed in Table 1 is not explained by a parsing artifact. In particular, the large self-vs-cross gap for `gemini-2.5-pro` and the opposite-direction gaps for `gpt-5.4` and `gpt-5.6-sol` persist under a stricter extraction rule, while models with no `think` prefix exposure remain unchanged as expected.

4.3 Family-Level Cluster-Robust Analysis

To attach inference to the descriptive pattern and test whether the self-review gap generalizes beyond a single verifier per family, we extend the design to nine verifiers across three families (five GPT, one DeepSeek, three Google) on a 50-instance pilot drawn from real software-engineering review tasks. Ten writer models produced a total of 483 injected reports over the 50 instances, and each report was judged by all nine verifiers, yielding 4,347 verdicts; unparseable verdicts were repaired by re-querying non-compliant verifiers with a larger token budget, leaving 4,345 reviews for analysis (2 unparseable). We fit a logistic regression of the flawed-report detection indicator on a self-review indicator, family indicators, and their interaction, with cluster-robust standard errors clustered by instance (report). The response is coded as 1 for a `FLAWED` verdict and 0 otherwise, and the model is estimated by maximum likelihood; it converged without warnings. Because reviews of the same report are correlated, we cluster standard errors by instance rather than treating reviews as independent; we do not additionally model writer-level, verifier-level, or report-level random effects, and we do not fit

Table 3: Full 4×4 writer–verifier sensitivity matrix (%). Rows are writers, columns are verifiers; the diagonal is self-review. This exposes the pairing-level structure that the self/cross aggregate in Table 1 collapses; the pairing-level false-positive rates are reported in Section 4.4.

Writer \ Verifier	`deepseek-chat`	`gpt-5.4`	`gpt-5.6-sol`	`gemini-2.5-pro`
`deepseek-chat`	0.0 (0/73)	38.4 (28/73)	69.9 (51/73)	61.6 (45/73)
`gpt-5.4`	1.6 (1/61)	13.1 (8/61)	19.7 (12/61)	62.3 (38/61)
`gpt-5.6-sol`	0.0 (0/64)	10.9 (7/64)	9.4 (6/64)	53.1 (34/64)
`gemini-2.5-pro`	12.5 (11/88)	37.5 (33/88)	20.5 (18/88)	86.4 (76/88)

Table 4: Controlled re-parsing study over the same 1,600 raw verifier responses. Only the parser changes.

Verifier	`think` prefix rate (%)	Strict vs. substring verdict difference
`deepseek-chat`	0.0	< 0.3 pp
`gpt-5.4`	26.0	< 0.3 pp
`gpt-5.6-sol`	94.5	< 0.3 pp
`gemini-2.5-pro`	0.0	< 0.3 pp

random slopes. We report Wald tests for the fixed effects and the family-by-self interaction. Because only three family-level units enter the interaction, the family-level Wald p-values may be anti-conservative and should be read as exploratory rather than confirmatory. The self-review main effect is negative and significant (log-odds -0.60, $p < 0.0001$): pooled across families, self-review detects fewer flaws than cross-review. However, the family $\times$ self interaction is strongly significant (Wald $\chi^2 = 20.9$, $p < 0.0001$), and the family-level self-versus-cross odds ratios reverse direction: the GPT-family verifiers show a self-review blind spot (OR 0.55), while DeepSeek (OR 1.72) and Google (OR 1.70) verifiers detect their own outputs better than others'. At the model level, however, the family averages conceal within-family heterogeneity (Table 5): the GPT blind spot spans +1.3 to +23.5 pp, and within Google the direction itself splits (gemini-2.5 verifiers stricter on self, gemini-3-pro-preview blind). We therefore do not treat "family" as a reliable unit; the self-review effect is a per-model property. The aggregate near-parity in Section 4.1 therefore masks a family-structured effect: the self-review penalty is a GPT-family phenomenon, not a property of LLM self-review in general (Figure 1).

At the model level, the family-level averages conceal substantial within-family heterogeneity. Table 5 reports the self-versus-cross defect recall for all nine verifiers. Within the GPT family, the blind-spot magnitude spans +1.3 to +23.5 percentage points (gpt-5.5 versus gpt-5.6-sol), so even a single vendor family is not a homogeneous unit. The direction also splits within Google: gemini-2.5-pro and gemini-2.5-flash are markedly stricter on their own outputs (-21.6 and -15.8 pp), whereas gemini-3-pro-preview shows a self-review blind spot (+9.8 pp). DeepSeek contributes a single parseable verifier (deepseek-chat), which is stricter on its own outputs (-8.7 pp); the reasoning model deepseek-reasoner was excluded because it did not emit verdicts in a parseable form. This indicates that "family" is not the operative unit: the self-review effect is a per-model property, and family-level summaries can be misleading.

4.4 Same-Model False-Positive Bias

The sensitivity analysis leaves open whether self-review differs from cross-review in error *type* rather than error *rate*. We therefore run a matched three-condition probe on 100 reports (50 injected, 50 clean), in which each report is reviewed by its writer (self), by the same model in a fresh session (same-model-fresh), and by a different model (cross-model). On injected reports the three conditions achieve comparable recall measured over 157 injected defects (self $116/157 = 73.9\%$ vs. cross $114/157 = 72.6\%$, exact sign test $p = 0.70$; same-model-fresh $121/157 = 77.1\%$). On clean reports, however, self-review and same-model-fresh review produce substantially more false positives than cross-model review (FPR 16.0% [8/50] and 18.0% [9/50] vs. 0.0% [0/50]; two-sided exact McNemar $p = 0.0078$ for self vs. cross). Because

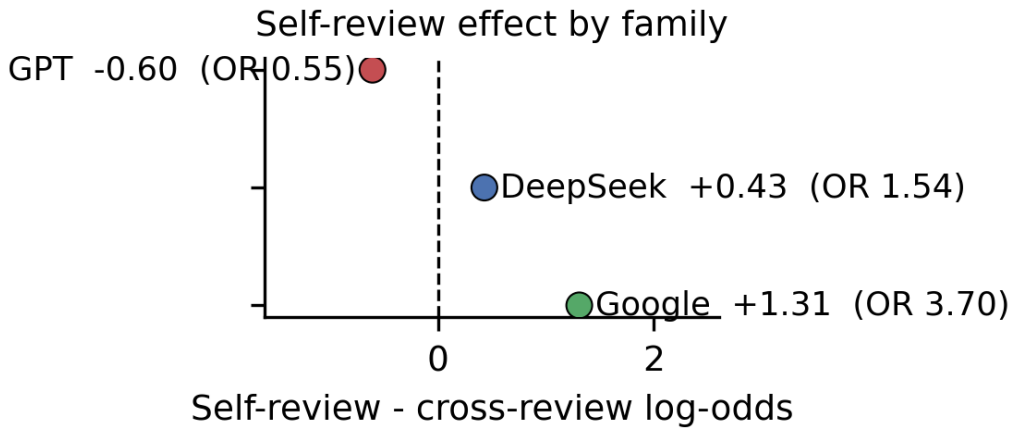


Figure 1: Self-review effect (self-review minus cross-review log-odds on flawed-report detection) by verifier family, from the cluster-robust logistic regression over 4,345 reviews. Negative values indicate a self-review blind spot; positive values indicate that self-review detects more flaws. The family × self interaction is significant (Wald $\chi^2 = 20.9$, $p < 0.0001$).

Table 5: Model-level self-review versus cross-review defect recall (percentage) for all nine verifiers in the family-level pilot. Δ is cross-review minus self-review. The GPT blind spot spans +1.3 to +23.5 pp; deepseek-chat and the gemini-2.5 verifiers are stricter on their own outputs, while gemini-3-pro-preview shows a self-review blind spot, so even within a vendor family the direction is inconsistent.

Family	Verifier	Self	Cross	Δ (pp)
GPT	gpt-5.4	80.0	85.7	+5.7
GPT	gpt-5.5	79.6	80.9	+1.3
GPT	gpt-5.6-luna	67.3	79.7	+12.4
GPT	gpt-5.6-sol	64.0	87.5	+23.5
GPT	gpt-5.6-terra	77.6	84.6	+7.0
DeepSeek	deepseek-chat	84.0	75.3	−8.7
Google	gemini-2.5-flash	85.4	69.7	−15.8
Google	gemini-2.5-pro	93.8	72.2	−21.6
Google	gemini-3-pro-preview	66.7	76.4	+9.8

same-model-fresh review shows the same false-positive rate as self-review despite having no memory of writing the report, the bias is consistent with model self-similarity rather than authorship. Self-review is thus not "blind" to defects—it is over-sensitive to apparent flaws in its own output style (Figure 2).

4.5 Counterbalanced Authorship-Label Probe

The fresh-session control in Section 4.4 separates authorship from model self-similarity by holding the model fixed and removing memory of writing. It does not, however, isolate the authorship *label* itself: a reviewer that sees "this is your output" may behave differently from one that sees no such cue, independently of the underlying content. To test whether the authorship label carries any effect on its own, we run a counterbalanced probe in which the report *content* is held fixed and only the authorship label is varied.

We draw 400 reports (200 injected, 200 clean) from the main benchmark and present each to a verifier under three label conditions—`self` (the report is described as authored by the verifier itself), `other` (authored by a different model), and `none` (no authorship stated)—holding the report text identical across conditions. This yields 1,200 reviews (three labels × 400

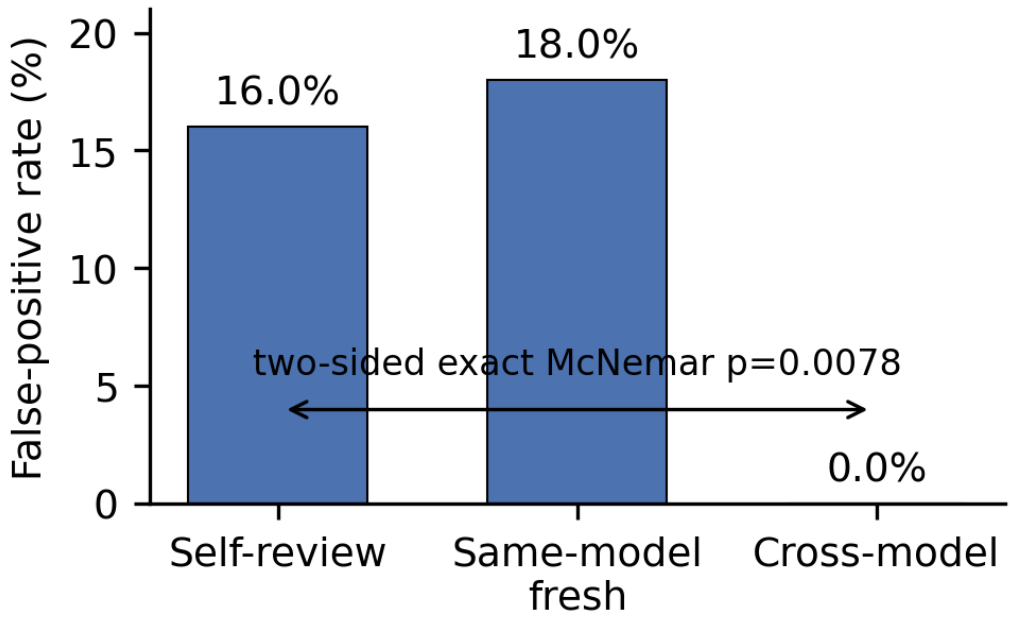


Figure 2: False-positive rate on clean reports for self-review, same-model fresh-session review, and cross-model review. Self-review and same-model-fresh review produce substantially more false positives than cross-model review (two-sided exact McNemar $p = 0.0078$ for self vs. cross).

reports), of which 1,199 parsed successfully and one failed to parse (excluded). Verdicts are extracted with the same parser as the main experiment.

The authorship label has no reliable effect on either detection or false positives. On injected reports, defect recall is 62.5% under the `self` label, 66.0% under `other`, and 63.0% under `none`; the paired self-versus-other comparison is not significant (exact McNemar $p = 0.115$, $b = 9$ and $c = 16$ discordant pairs). On clean reports, the false-positive rate is 27.5% under `self`, 30.0% under `other`, and 28.6% under `none`; the paired self-versus-other comparison is again not significant (exact McNemar $p = 0.273$, $b = 17$ and $c = 22$ discordant pairs). Holding content fixed and varying only the authorship label therefore produces no detectable change in recall or false-positive rate.

This null result is the central identification finding of the paper: the self-review asymmetries reported in Sections 4.1 and 4.3 are not driven by the authorship label per se. When the label is the only thing that varies, self-review does not differ from other-review. The apparent self-review blind spot in the earlier cross-model comparisons must therefore arise from the content itself—specifically, from reviewing one's *own generated answer* (which differs from other models' answers in difficulty, style, and defect distribution) rather than from the label "this is yours." This reframes the field's "self-review blind spot" as a content-ownership effect, not a label effect.

4.6 Objective Ground-Truth Replication on Executable Code

To check that the two asymmetries are not artifacts of the author-controlled injection, we replicate on a benchmark with objective ground truth: 164 HumanEval coding tasks scored by local execution and doctests (pass/fail), with three generators (DeepSeek-chat, GPT-5.4, Gemini-2.5-pro) producing 492 solutions (123 passing, 369 failing). Both asymmetries replicate: on execution-passing solutions, self-review flags 20.7% as incorrect versus 7.5% for cross-review (two-sided Fisher exact $p = 0.0069$), and recall gaps remain model-dependent (DeepSeek-chat self 26.4% vs. cross 99.3%; GPT-5.4 FPR 33.9% vs. 9.8%; Gemini-2.5-pro

self 88.7% vs. cross 20.5%), with the pooled recall difference not significant (49.1% vs. 54.5%, $p = 0.167$)—the family effects canceling exactly as the aggregate masks.

5 Discussion

5.1 Design Implications

Independence is not a sufficient selection criterion: the aggregate sensitivity gain is only 0.9 pp, and cross-model review performs worse on balanced accuracy (56.6% vs. 59.6%). Verifier selection should be empirical rather than role-based, reporting sensitivity on flawed outputs, FPR on clean outputs, a class-balanced summary, and performance disaggregated by writer identity. The observed reversals imply there may be no universally best verifier independent of the writer distribution, so writer–verifier pairing should be treated as a first-class design variable. Self-review should also not be rejected solely because it is non-independent: `gemini-2.5-pro`'s supplied sensitivity is higher in self-review than in cross-review.

5.2 Implications for Multi-Agent Verification

Multi-agent systems often motivate extra model calls by appealing to diversity or independence, but our findings suggest additional agents should show incremental value on ground-truth-labeled data, not merely architectural or provider diversity. Under this framework, independence is one candidate mechanism for obtaining complementary errors, not the objective itself.

6 Limitations

Data scope. The fully crossed experiment includes four writer/verifier models over a single benchmark of technical reports (286 flawed, 114 clean), so the observed patterns need not transfer to other model families, versions, task types, or class distributions.

Reproducibility and causal scope. Labels came from a programmatic injection log; a single-annotator human audit of a 50-report blinded sample yielded sensitivity 37.0% and accuracy 46%, indistinguishable from chance. Full API logs, exact model snapshots, repeated runs, and decoding parameters were not recorded, and the crossed design does not identify the mechanisms behind each model's behavior.

7 Conclusion

This study reports two asymmetries of LLM self-review. First, recall gaps are model-dependent rather than universal: a cluster-robust logistic regression finds the self-review penalty concentrated in the GPT family (odds ratio 0.55), while DeepSeek (1.72) and Google (1.70) verifiers are stricter on their own outputs, yet within Google the direction splits, so vendor family is not a reliable unit. Second, error type is asymmetric: at comparable recall, self-review produces substantially more false positives on clean reports (16.0% vs. 0.0%, two-sided exact McNemar $p = 0.0078$), a bias that a fresh-session control localizes to model self-similarity rather than authorship. A controlled re-parsing study rules out a parsing artifact, and a counterbalanced label probe shows the authorship label itself has no reliable effect, so the asymmetries are a content-ownership effect, not a label effect. Practically, self-review reliability must be validated for the target model: neither universal trust in self-review nor a blanket preference for independent cross-review is justified.

References

Yuntao Bai et al. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Weize Chen et al. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. *arXiv preprint*, 2024.

Xinyun Chen et al. ReConcile: Round-Table Conference Improves Reasoning via Consensus Among Diverse Large Language Models. *arXiv preprint*, 2024.

Karl Cobbe et al. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving Factuality and Reasoning in Language Models through Multiagent Debate. *ICML*, 2024.

Yann Dubois et al. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. *NeurIPS*, 2023.

Zhibin Gou et al. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. *ICLR*, 2024.

Jiacheng Huang et al. Large Language Models Cannot Self-Correct Reasoning Yet. *arXiv preprint arXiv:2310.01798*, 2024.

Geoffrey Irving, Paul Christiano, and Dario Amodei. AI Safety via Debate. *arXiv preprint arXiv:1805.00899*, 2018.

A. R. Khan et al. Debating with More Persuasive LLMs Leads to More Truthful Answers. *arXiv preprint*, 2024.

Guohao Li et al. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. *NeurIPS*, 2023.

Percy Liang et al. Holistic Evaluation of Language Models. *Transactions on Machine Learning Research*, 2023.

Hunter Lightman et al. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050*, 2024.

Yang Liu et al. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. *EMNLP*, 2023.

Aman Madaan et al. Self-Refine: Iterative Refinement with Self-Feedback. *NeurIPS*, 2023.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. *EMNLP*, 2023.

William Saunders et al. Self-Critiquing Models for Assisting Human Evaluators. *arXiv preprint arXiv:2206.05802*, 2022.

Noah Shinn et al. Reflexion: Language Agents with Verbal Reinforcement Learning. *NeurIPS*, 2023.

Jonathan Uesato et al. Solving Math Word Problems with Process- and Outcome-Based Feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Xinpeng Wang et al. Fair Evaluators Are Hard to Find: Evaluating Social Bias in Large Language Model Evaluators. *arXiv preprint*, 2024.

Sean Welleck et al. Generating Sequences by Learning to Self-Correct. *ICLR*, 2019.

Qingyun Wu et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Lianmin Zheng et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *NeurIPS*, 2023.

Tian Liang et al. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate. *EMNLP*, 2024.

Yixuan Weng et al. Large Language Models are Better Reasoners with Self-Verification. *Findings of EMNLP*, 2023.

Roi Cohen et al. LM vs LM: Detecting Factual Errors via Cross Examination. *EMNLP*, 2023.

Shehzaad Dhuliawala et al. Chain-of-Verification Reduces Hallucination in Large Language Models. *Findings of ACL*, 2024.

Ryan Koo et al. Benchmarking Cognitive Biases in Large Language Models as Evaluators. *Findings of ACL*, 2024.

Chuan Guo et al. On Calibration of Modern Neural Networks. *ICML*, 2017.

Saurav Kadavath et al. Language Models (Mostly) Know What They Know. *arXiv preprint arXiv:2207.05221*, 2022.

Use of AI Tools

This paper studies large language models (LLMs) as reviewers of technical reports. The LLMs used as writers and verifiers—`deepseek-chat`, `gpt-5.4`, `gpt-5.6-sol`, and `gemini-2.5-pro`—are the subjects of the study: their review outputs constitute the experimental data, rather than tools used to assist the research itself. Ground-truth labels for the primary benchmark were produced programmatically (an injection log) and by execution-based checks (doctests), not by LLM judgment.

In preparing this manuscript, the authors used LLM tools to assist with literature retrieval, experimental-design critique, and revision of the manuscript text. All model-assisted content was reviewed and verified by the authors, who take full responsibility for the final manuscript.