AUDITING THE AUDITORS: VALIDATING AUTOMATIC EVALUATION IN MULTI-MODEL LLM CONSENSUS SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-model review aims to improve language-model answers through critique and revision, but evaluating its benefits requires reliable correctness judgments. We examine this problem through TriGuard, a three-model generate–review–revise pipeline using quantized open-weight models. Our evaluation combines controlled replay, which holds primary answers and reviewer feedback fixed across synthesis policies, with human annotation and documented adjudication. The first annotator received recorded AI assistance; a second annotator independently assessed the answers while blinded to prior labels and study results. On two 100-question TruthfulQA slices, we audit all 24 questions affected by a static gate that bypasses synthesis when reviewer feedback is net-negative and the primary model has higher historical accuracy. Automatic scoring indicates a 7.5-percentage-point accuracy reduction relative to unconditional synthesis. Adjudicated labels instead imply a reduction of 5–6 percentage points across the 200 questions. This range reflects unresolved-label assignments, not sampling uncertainty. A separate, outcome-enriched 20-question audit indicates a positive primary-to-synthesis difference within that selected sample, without establishing a full-sample benefit. We also document limitations of two evaluation components: a semantic-equivalence mechanism agrees with human labels on 15 of 30 answer pairs from ten questions, and a HaluEval judge continues to approve reference-contradicting answers after a prompt-and-parser repair. These findings show how evaluation choices can materially change conclusions about multi-model systems. We present a reproducible audit workflow that distinguishes implementation errors, substantive judgment failures, and unresolved labels, while bounding claims to the comparisons actually validated.

1 INTRODUCTION

Large language models (LLMs) are now routinely used to answer factual questions, and their fluency is often mistaken for reliability: a model can state an incorrect claim with the same confident tone as a correct one, and has no built-in signal distinguishing the two (Ji et al., 2023). This has motivated a rapidly growing body of work on *multi-agent* approaches, in which several models, or several calls to the same model, are combined in the hope that disagreement between them surfaces errors a single call would miss (Du et al., 2024; Irving et al., 2018; Li et al., 2024; Liang et al., 2024). Because these systems are typically evaluated using the same kind of automatic, model-based scoring they are meant to improve upon, a second, less-examined question sits underneath the first: whether the evaluators used to judge such systems are themselves reliable enough to trust (Zheng et al., 2023; Gu et al., 2024; Chen et al., 2024).

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.

Existing work has compared automatic judges against human judgments in specific settings (Zheng et al., 2023; Chen et al., 2024; Haldar and Hockenmaier, 2025; Fu and Liu, 2025), and separately, some multi-agent systems have been checked against controlled baselines (Zhang et al., 2025). But evidence about an evaluator's reliability in one setting does not establish its reliability for a different model configuration, task, prompt, or scoring pipeline; nor does a validated evaluator in one study guarantee that a separately reported system's own accuracy gain was measured against a properly controlled baseline isolating the mechanism responsible for it. We therefore audit the evaluator and the policy comparison together, in the specific setting studied here, rather than relying on either being established separately elsewhere. Recent evidence suggests both kinds of gap matter in practice: multi-agent debate has been shown to sometimes reduce accuracy as agents converge on agreement rather than correctness (Wynn et al., 2025; Yao et al., 2025), multi-agent evaluations often fail to outperform simpler baselines once heterogeneity and evaluation choices are examined carefully (Zhang et al., 2025), and LLM judges have been shown to disagree with each other and with human raters more than headline accuracy figures alone would suggest (Haldar and Hockenmaier, 2025; Fu and Liu, 2025).

This paper addresses both gaps directly, for one specific system and one specific design choice within it. We build TriGuard, a heterogeneous three-model review pipeline, and rather than reporting its own automatic scores, we introduce a controlled-replay protocol that holds a primary model's answer and its reviewers' feedback fixed while varying only the downstream synthesis policy, isolating that policy's causal effect from the resampling noise and prompt-wording confounds present in a naive before/after comparison. We then validate the resulting labels, not with a second automatic judge, but with two independent human annotators following a documented adjudication protocol, auditing the complete set of questions our central design choice; a static reviewer gate actually altered. Where this validation process itself uncovered evaluator unreliability, on a second benchmark, we report that failure explicitly rather than discard it. Our contributions are:

- A controlled-replay protocol that isolates a synthesis policy's causal effect from generation-resampling and prompt-wording confounds, including a prospectively specified confirmatory test in which an exploratory finding did not reach significance on a second, held-out slice, reported with a precise, non-overclaiming statement of what that non-confirmation does and does not establish.
- A complete, two-annotator, documented-adjudication human audit of a static reviewer gate's entire behavioral footprint (every question it altered, not a convenience sample), finding a harmful accuracy effect that persists under adjudicated human judgment though smaller in magnitude than automatic scoring alone suggested.
- A documented evaluator-reliability failure on a second benchmark (HaluEval), in which an LLM judge validated an answer that directly contradicts its reference, traced to a cause beyond output-format parsing, leading us to withhold accuracy claims on that benchmark rather than report an unvalidated number.
- A semantic-equivalence voting mechanism for free-text self-consistency, engineered to avoid known structural failure modes (transitive-chaining errors, gold-answer leakage, silent handling of non-consensus), reported as evaluated and rejected after failing human validation, rather than adopted on the strength of its design alone.

2 Related Work

Multi-agent debate and consensus. The propose-critique-revise structure underlying TriGuard is closest to multi-agent debate (Du et al., 2024; Irving et al., 2018), whose central premise is that models checking each other's reasoning improves factual accuracy and mathematical correctness over independent generation. Subsequent work has both extended and complicated this premise. Li et al. (2024) show that aggregating

many samples from the same model, without explicit debate, can rival more complex multi-agent architectures, a finding that raises the question of whether a debate mechanism's apparent benefit is separable from simple ensembling. Liang et al. (2024) argue that debate's value lies in encouraging divergent reasoning paths rather than convergence on a single answer; our gate mechanism does the opposite; it is designed to defer to a historically-accurate primary model rather than incorporate reviewer disagreement. Amayuelas et al. (2024) shows multi-agent debate frameworks are vulnerable to adversarially injected misinformation from a single compromised agent, a different but related concern to our own finding that reviewers can push a correct answer toward a shared misconception without any adversarial manipulation at all. Wu and Ito (2025) find persistent disagreement can be more valuable than forced convergence, consistent with our audited gate result.

Sycophancy and failure modes in multi-agent systems. Closest to our central empirical finding is work directly documenting negative effects from cross-model review (Wynn et al., 2025), which shows debate can decrease accuracy over time as models shift from correct to incorrect answers in response to peer pressure, regardless of the numerical balance between stronger and weaker models in the group. Yao et al. (2025) study sycophancy specifically as a driver of multi-agent debate failure, finding that agreement-seeking behavior between models can move a correct answer toward an incorrect one. Pitre et al. (2025) proposed a mitigation targeting sycophancy directly in a consensus setting, a different design response to the same underlying problem our static gate attempts, and unlike our own audited result, report a positive outcome for their mitigation, a discrepancy we note rather than resolve, since the two systems, models, and evaluation protocols differ in ways that make a direct comparison unsafe without a shared controlled replication. Zhang et al. (2025) argue more broadly that multi-agent debate's evaluation culture has overvalued positive results relative to careful baseline comparison and model heterogeneity, a critique our own null and negative results are offered partly in response to.

LLM-as-judge reliability. Automatic evaluation by a language model is now standard practice (Zheng et al., 2023; Gu et al., 2024), but a growing body of work finds this practice less reliable than its convenience suggests. Chen et al. (2024) demonstrated LLM judges exhibit systematic biases distinct from human raters, not merely noisier versions of them. Haldar and Hockenmaier (2025) find LLM-as-judge frameworks are self-inconsistent; the same judge, given the same input twice, can disagree with itself, a concern our own item-level judge-stability audit directly tested and, in our specific configuration, did not find, though our HaluEval finding shows self-*consistency* and factual *correctness* are separate properties a judge can satisfy one of without the other. Fu and Liu (2025) document reliability variation across languages and task types, reinforcing that a judge validated in one setting should not be assumed reliable in another without direct revalidation, precisely the gap between our TruthfulQA judge fix and our HaluEval judge failure. Feng et al. (2026) proposed statistical frameworks for drawing valid conclusions *despite* judge imperfection, using a small human-labelled calibration set to estimate a judge's error rates and correct for them; our own approach is more conservative where a judge's errors are severe and directionally unclear, withholding accuracy claims entirely rather than attempting a statistical correction, though the two approaches are complementary rather than competing.

Benchmarks. TruthfulQA (Lin et al., 2022) measures whether models reproduce common human misconceptions rather than stating known facts, and is our primary evaluation benchmark. HaluEval (Li et al., 2023) evaluates hallucination detection and generation; we use its question-answering subset and report a specific evaluator failure on it (Section 3.6).

3 System Architecture

TriGuard consists of a primary model (Llama 3.1 8B Instruct (Grattafiori et al., 2024)) that generates an initial answer, two independent reviewer models (Qwen 2.5 7B Instruct (Qwen Team et al., 2024), Gemma 2 9B Instruct (Gemma Team et al., 2024)) that critique it without seeing each other's feedback, and a synthesis

step in which the primary model produces a final answer incorporating that feedback. We evaluate three synthesis policies: **v1** (unconditional synthesis, a plain prompt presenting both reviews without weighting); **v3** (a static gate, described formally in Section 3, that bypasses synthesis under a specific condition); and **v4** (conservative synthesis, instructed to adopt only corrections to explicitly stated errors, never unverified supplementary detail). An independent model (Phi-3 Mini (Abdin et al., 2024)) serves only as an automatic judge and is never a pipeline member. Hyperparameters and the hardware/software stack are reported in Appendix 6.

3.1 The Gate Decision Rule

Let q denote a question, $a_p(q)$ the primary model's initial answer, and $s_1(q), s_2(q) \in [0, 1]$ the two reviewers' normalized scores of $a_p(q)$, where 1 denotes full endorsement and 0 denotes full rejection. Let $w_p, w_1, w_2 \in [0, 1]$ denote the primary model's and each reviewer's measured historical accuracy on a disjoint calibration set. The v3 gate's decision function is:

$$G(q) = \begin{cases} \texttt{bypass} & \text{if } \bar{s}(q) < 0.5 \text{ and } w_p > \bar{w}_{\text{rev}} \\ \texttt{synthesize} & \text{otherwise} \end{cases} \tag{1}$$

where $\bar{s}(q) = \frac{1}{2}(s_1(q) + s_2(q))$ is the mean reviewer score and $\bar{w}_{\text{rev}} = \frac{1}{2}(w_1 + w_2)$ is the mean reviewer historical accuracy. In words: the gate withholds synthesis, returning the primary's original answer unchanged, only when reviewers net-reject the answer *and* the primary model has a track record more accurate than its reviewers' average; an attempt to encode "trust the historically better model when reviewers disagree with it." When `synthesize` is selected, the final answer is produced by an LLM call that additionally states each reviewer's w_i in-context; when `bypass` is selected, the final answer is $a_p(q)$ unchanged.

We distinguish this original v3 implementation from the isolated gate intervention evaluated in our central result (Section 3.7). The original implementation changes the synthesis prompt on non-bypassed questions in addition to making the bypass/synthesize decision, which would confound the gate's own effect with the effect of that prompt change. To isolate the gate decision alone, the audited gate returns the *cached unconditional-synthesis (v1) answer* whenever synthesis is permitted, and the frozen primary answer whenever synthesis is bypassed:

$$a_G(q) = \begin{cases} a_p(q), & G(q) = \texttt{bypass}, \\ a_1(q), & G(q) = \texttt{synthesize}, \end{cases} \tag{2}$$

where $a_1(q)$ is the cached v1 (plain-prompt, unconditional synthesis) answer on the same frozen input, not a freshly generated, differently-prompted answer. We refer to this reconstructed quantity as the *isolated gate* throughout our central result, and use "v3" only when referring to the original implementation's own distinct behavior.

3.2 Controlled Replay

Measuring the gate's effect by generating the primary answer and reviews independently for the gate and for unconditional synthesis is confounded by two factors: generation resampling (an unseeded language model can produce different text for nominally the same call) and a change in synthesis prompt wording, separate from the gate's own bypass/synthesize decision. We remove both by generating $a_p(q)$, $s_1(q)$, $s_2(q)$, both reviewers' free-text feedback, and a single v1 (unconditional) synthesis answer $a_1(q)$ *once* per question, then constructing the gated answer $a_G(q)$ from that identical frozen input via Equation 2 (Algorithm 1). The gate's isolated effect is then the paired difference between $a_G(q)$'s correctness and $a_1(q)$'s correctness on shared frozen inputs, removing both confounds simultaneously.

This construction supports two distinct comparisons, which we keep notationally separate throughout: the gate audit (Section 3.7) measures the gate against *unconditional synthesis*, not against the primary answer

Algorithm 1 Controlled Replay for the Isolated Gate Audit

Input: question q, primary model M_p, reviewer models M_1, M_2, fixed gate parameters
Output: (a_p, a_1, a_G) – the primary, v1, and gated answers

$a_p \leftarrow M_p$.generate(q) ▷ frozen primary answer
for $j \in \{1, 2\}$ **do**
  $(s_j, f_j) \leftarrow M_j$.review$(q, a_p)$ ▷ frozen score and feedback
end for
$a_1 \leftarrow M_p$.synthesize$_{v1}(q, a_p, s_1, f_1, s_2, f_2)$ ▷ single v1 synthesis call
Cache a_p, both reviews, and a_1
if $G(q) =$ `bypass` **then**
  $a_G \leftarrow a_p$
else
  $a_G \leftarrow a_1$ ▷ reuse the cached v1 answer, no new call
end if
return (a_p, a_1, a_G)

alone,

$$\Delta_G = \frac{1}{N}\sum_{i=1}^{N}(Y_{G,i} - Y_{1,i});$$

the supporting pilot (Appendix 10) measures unconditional synthesis against the *primary answer*,

$$\Delta_{P1} = \frac{1}{N}\sum_{i=1}^{N}(Y_{1,i} - Y_{p,i}),$$

where $Y_{G,i}, Y_{1,i}, Y_{p,i} \in \{0, 1\}$ denote the gated, unconditional-synthesis, and primary answers' correctness on question i. These are different estimands with different baselines, not two readings of the same quantity.

3.3 Paired Significance Testing

For a binary correctness outcome and two paired methods A, B evaluated on the same n questions, let n_{01} denote the count of questions where A is incorrect and B is correct, and n_{10} the reverse. McNemar's exact test evaluates the null hypothesis that the two discordant outcomes are equally likely,

$$H_0 : \Pr(Y_A = 0, Y_B = 1) = \Pr(Y_A = 1, Y_B = 0),$$

using the exact binomial tail probability applied to the observed counts n_{01}, n_{10}:

$$p = \min\left(1,\ 2\sum_{i=0}^{k}\binom{n_{01}+n_{10}}{i}\left(\tfrac{1}{2}\right)^{n_{01}+n_{10}}\right), \qquad k = \min(n_{01}, n_{10}) \tag{3}$$

where $\binom{n}{i}$ is the standard binomial coefficient. We use this test throughout for paired method comparisons, since it conditions only on the discordant pairs n_{01}, n_{10} and does not require the independent-samples assumption a two-proportion z-test would. Where we report a paired difference with an approximate confidence interval, we use the sample-variance normal approximation with Bessel's correction,

$$\widehat{\Delta} = \frac{n_{01} - n_{10}}{N}, \qquad \widehat{\mathrm{SE}}(\widehat{\Delta}) = \sqrt{\frac{n_{01} + n_{10} - N\widehat{\Delta}^{\,2}}{N(N-1)}}, \tag{4}$$

giving an approximate 95% interval $\widehat{\Delta} \pm 1.96\,\widehat{\mathrm{SE}}(\widehat{\Delta})$. We flag this explicitly as a normal approximation to the paired difference, not an exact McNemar interval, and distinct from the unresolved-label bounds introduced next, which reflect annotation ambiguity rather than sampling variability.

3.4 UNRESOLVED-LABEL BOUNDS

Human annotation can leave a question's correctness genuinely unresolved. Rather than discard such cases or force a binary label, we report the range of accuracy differences consistent with every possible resolution. For method B against baseline A on question j, define $d_j = Y_{B,j} - Y_{A,j} \in \{-1, 0, +1\}$ where $Y_{\cdot,j} \in \{0, 1\}$ is correctness. When both labels are resolved, d_j is a fixed value; when one or both are unresolved, we take the achievable range:

$$\Delta_{\min} = \frac{1}{n} \sum_{j=1}^{n} \min_{d_j \in D_j} d_j, \qquad \Delta_{\max} = \frac{1}{n} \sum_{j=1}^{n} \max_{d_j \in D_j} d_j \tag{5}$$

where $D_j \subseteq \{-1, 0, +1\}$ is the set of values d_j can take given which of $Y_{A,j}, Y_{B,j}$ are resolved. For a question where B's output is definitionally identical text to A's (as with our gate's non-bypassed questions), $D_j = \{0\}$ regardless of either label's resolution status, since a shared but unknown correctness value cannot produce a nonzero difference against itself. We enforce this identity constraint structurally rather than computing it from labels, so that annotation uncertainty about a single shared answer's correctness can never manifest as spurious uncertainty about the difference between that answer and itself. $[\Delta_{\min}, \Delta_{\max}]$ is a bound reflecting remaining label ambiguity, not a confidence interval, and we report it as such throughout.

3.5 TWO-ANNOTATOR ADJUDICATION PROTOCOL

Each answer record is independently judged by two annotators. The first annotator's judgments were produced with recorded AI assistance (verdict directly suggested by AI for 22 of 54 records, interpretive guidance for 6, no assistance for 18, and two-annotator adjudication for the remaining 8); the second annotator judged independently, blinded to the first annotator's labels, the automatic judge's verdict, and the study's hypotheses. We do not describe this as two fully independent, unaided human annotations; the first annotator's process differed from the second's in this specific way, and we keep that distinction explicit throughout rather than treating the pair as symmetric. Pre-adjudication agreement is reported as raw percent agreement, $\frac{1}{n}\sum_j \mathbb{1}[L_{1,j} = L_{2,j}]$, rather than a chance-corrected statistic such as Cohen's κ we note this distinction explicitly in Appendix 11, since percent agreement does not correct for the agreement expected by chance alone. Disagreements are resolved through documented joint adjudication, both annotators examining the question and answer text together and recording their reasoning, after both independent judgments had already been recorded rather than by either annotator's label taking automatic precedence (Algorithm 2).

3.6 EXPERIMENTS: EVALUATOR VALIDATION

We audited the evaluation pipeline in stages, distinguishing output-parsing failures from substantive judgment errors and revisiting selected automatic conclusions through human annotation. This followed the general concern that LLM judges can exhibit systematic, non-obvious unreliability (Zheng et al., 2023; Haldar and Hockenmaier, 2025; Fu and Liu, 2025). Beyond the targeted checks described below, we also computed precision, recall, and F1 for both automatic judges directly against our own adjudicated reference labels (Figure 1). This is a descriptive result on the specific audited record set, not a representative estimate of either judge's general-purpose performance. The TruthfulQA judge, evaluated against $n = 50$ resolved adjudicated records drawn from the gate-audit and primary-vs-v1 ledgers, achieved precision 0.727, recall 1.000, F1 0.842, and accuracy 0.880; recall of 1.000 means no false negatives were observed in this specific set, not that the judge has perfect recall in general; its precision gap shows it does sometimes call a human-confirmed-incorrect answer correct, consistent with the leniency independent review corrected elsewhere in this paper (Section 3.7). The semantic-equivalence judge's per-class scores (macro-averaged F1 = 0.498 across SAME/ DIFFERENT/UNCLEAR, $n = 30$) are shown alongside it; we report the raw agreement and per-class scores without characterizing them against a chance baseline, since neither 50.0% agreement nor a

Algorithm 2 Two-Annotator Adjudication

Input: answer records $R = \{r_1, \ldots, r_n\}$, annotators $\text{Ann}_1, \text{Ann}_2$
Output: L_{final} – final labels; agreement – pre-adjudication agreement rate

for $r \in R$ **do**
  $L_1[r] \leftarrow \text{Ann}_1.\text{judge}(r)$ ▷ independently, no automatic label shown
end for
for $r \in R$ **do**
  $L_2[r] \leftarrow \text{Ann}_2.\text{judge}(r)$ ▷ blinded to L_1 and study hypotheses
end for
$D \leftarrow \{r \in R : L_1[r] \neq L_2[r]\}$
agreement $\leftarrow (n - |D|)/n$
for $r \in D$ **do**
  $L_{\text{final}}[r] \leftarrow \text{JointAdjudicate}(r, L_1[r], L_2[r])$ ▷ documented reasoning
end for
for $r \in R \setminus D$ **do**
  $L_{\text{final}}[r] \leftarrow L_1[r]$
end for
return L_{final}, agreement

macro-F1 of 0.498 has a well-defined chance interpretation without an explicit null model and class distribution. The per-class breakdown is uneven in an informative way: DIFFERENT has the lowest F1 (0.40), while UNCLEAR shows recall of 1.00 against precision of only 0.33; the judge predicts "unclear" far more often than annotators actually used that label, rather than reserving it for genuinely ambiguous cases. This human–machine agreement figure is a different comparison, on a different task, from the human–human agreement figures reported in this paper (Figure 4); we do not treat them as directly comparable statistics.

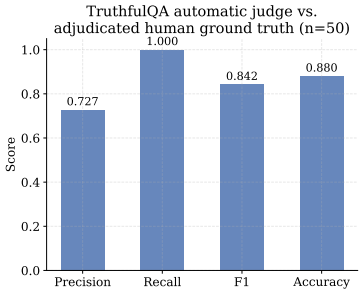

Figure 1: TruthfulQA judge precision, recall, F1, and accuracy vs. adjudicated labels.

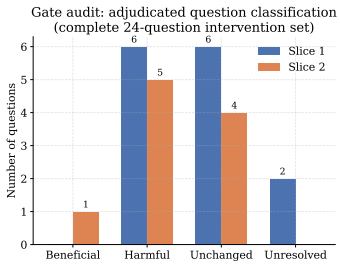

Figure 2: Adjudicated classification of the 24 questions the gate altered.

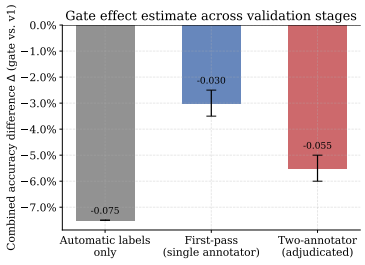

Figure 3: Gate effect across validation stages. Bars show bound midpoints, **not confidence intervals**.

3.6.1 TruthfulQA Judge

An initial keyword-overlap scorer produced false negatives on clearly correct, concise answers. We replaced it with an independent LLM judge (Phi-3) for all TruthfulQA scoring reported in this paper.

3.6.2 HALUEVAL JUDGE: A DOCUMENTED FAILURE

Using the same judge architecture on HaluEval, we first re-judged 100 already-scored answers against original substring-based labels under the *pre-repair* configuration; 82% agreed. Examining the 18 disagreements, we found a parsing defect: the judge's prompt asked for a `CORRECT/INCORRECT` verdict but did not explicitly forbid answering the underlying question itself, so a bare *"Yes."* response, ambiguous between "yes, correct" and an echoed object-level answer was silently parsed as a false negative in two identified HaluEval cases. We rewrote the prompt to forbid this ambiguity and replaced substring-search parsing with exact-match parsing against $\{\text{CORRECT}, \text{INCORRECT}, \text{UNCERTAIN}\}$, with a bounded retry on non-conforming output. A representative repaired-configuration sample of 50 records (25 HaluEval, 25 TruthfulQA) found 0/50 persistent-invalid responses; a statement about output-format compliance after repair, not a claim that future format failures are impossible. Format compliance is separate from substantive correctness. We subsequently retargeted the specific wrong-fact HaluEval cases identified above and re-ran them under the *repaired* configuration: they still failed. The starkest case: a candidate answer beginning *"No, the New Orleans Outfall Canals are not the same length…"* against a reference of *"yes"*; a direct polarity contradiction was still judged CORRECT after repair, confirmed not to be a data-pipeline or parsing artifact. We do not infer a representative substantive error rate from these targeted retests, and we do not claim to know the underlying cause; we report only that this evaluator configuration remains unreliable for HaluEval after the parser repair, and we report no HaluEval accuracy results in this paper as a consequence.

3.7 CENTRAL RESULT: THE AUDITED GATE, AND TWO-ANNOTATOR HUMAN VALIDATION

Applying Algorithm 1 to two held-out 100-question TruthfulQA slices, an isolated (confound-controlled) comparison of the audited gate against v1 on their shared frozen inputs found a statistically significant harmful effect in one slice ($n_{01} = 0$ corrections, $n_{10} = 11$ damages; McNemar exact $p = 0.0010$ via Equation 3). Because the gate's non-bypassed questions return the cached v1 answer *unchanged* (Equation 2), their contribution to any accuracy difference is exactly zero by construction, allowing a complete audit of the gate's entire behavioral footprint across both slices, not a convenience sample of it without individually auditing all 200 evaluated questions. The gate altered 24 of 200 evaluated questions. We audited all 24 (48 answer records), plus 6 records from 3 non-bypassed control questions, via Algorithm 2. Pre-adjudication agreement on the 48 bypass-relevant records was 40/48 (83.3%); on the 6 controls, 6/6 (100%). All 8 disagreements were jointly adjudicated, resolving 6 and retaining 2 as genuinely unresolved. As an invariant check, all three non-bypassed control questions where the gate's output is definitionally identical text to v1's received consistent human labels across both copies.

3.7.1 RESULT

Applying Equation 5 to the adjudicated ledger:

$$\Delta_{\text{slice 1}} \in [-0.080, -0.060], \qquad \Delta_{\text{slice 2}} = -0.040 \tag{6}$$

Combined across both slices, $\Delta_{\text{combined}} \in [-0.060, -0.050]$, implying the gate cost 10–12 net correct answers across the 200 evaluated questions. The same 24-question set scored under automatic labels alone implied $\Delta = -0.075$, larger in magnitude, though the same direction. The adjudicated estimate is itself *larger* in magnitude than an earlier single-annotator, AI-assisted first pass on the same 48 records ($[-0.035, -0.025]$): independent review corrected several first-pass judgments that had been too lenient toward the gate. Both unresolved slice-1 pairs have a gated answer adjudicated as incorrect and an unconditional-synthesis (v1) answer whose correctness remains unresolved; each therefore contributes either -1 or 0 to the gated-minus-unconditional difference, never $+1$, since a resolved incorrect gate answer cannot exceed an unresolved v1 answer. The six resolved harmful pairs contribute -6; combined with each unresolved pair's $\{-1, 0\}$ range, this gives the slice-1 bound $[-8, -6]/100 = [-0.080, -0.060]$ exactly,

not as a loose approximation. To state the central result plainly: across the 200 evaluated questions, adjudicated labels imply that the isolated gate reduced accuracy by 5–6 percentage points relative to unconditional synthesis. This range accounts for unresolved labels, conditional on the resolved adjudications; it does not quantify sampling uncertainty, and it does not establish the effect on other questions, model configurations, or gating policies. This finding is directionally consistent with prior evidence that cross-model agreement pressure can degrade rather than improve accuracy (Wynn et al., 2025; Yao et al., 2025), now supported by a complete, adjudicated audit of one specific gate's entire intervention set rather than an aggregate automatic score. We do not claim this generalizes to other gating policies, model families, or benchmarks, and we do not identify sycophancy or consensus pressure as *this* gate's mechanism: the audited gate is a static rule that preserves the primary answer against net-negative reviewer feedback, and its observed harm supports a claim about this specific intervention, not a claim about a general psychological or behavioral mechanism underlying it.

Table 1: Comparison of TriGuard's design against related multi-model approaches

Method	Heterogeneous models	Conditional/gated revision
Self-consistency (Wang et al., 2023)	✗	✗
Multi-agent debate (Du et al., 2024)	partial[1]	✗
More-agents scaling (Li et al., 2024)	✗	✗
CONSENSAGENT (Pitre et al., 2025)	✗	✓
TriGuard (this work)	✓	✓

Table 1 compares design properties as reported in each cited work, not experimental accuracy, since we did not re-implement these systems under our own controlled conditions; a fair head-to-head accuracy comparison would require exactly the shared-model, shared-benchmark replication this paper's methodology was built to support, and is left to future work.

4 Conclusion

We built a heterogeneous multi-model consensus pipeline and, rather than reporting its own numbers as evidence, subjected one specific design choice, a static reviewer gate, to a controlled-replay methodology and a complete, two-annotator, adjudicated human audit of its entire behavioral footprint. The audited result is that this gate measurably hurts accuracy, smaller in magnitude than automatic scoring alone suggested but persisting in direction under conservative treatment of remaining label uncertainty. We report two further negative results from the same validation process: a carefully engineered semantic-equivalence voting mechanism that nonetheless failed human validation, and a documented LLM-judge failure on a second benchmark that led us to withhold accuracy claims there entirely. Future work should extend the controlled-replay and two-annotator audit protocol to a representative (not outcome- enriched) sample of the full primary-vs-synthesis comparison, to additional gating policies and model families, to a repaired and independently re-validated HaluEval evaluator, and to a properly controlled head-to-head replication of the related methods compared only by design in Table 1. We believe the validation discipline demonstrated here, reporting a confirmatory non-result honestly, validating every automatic judge before trusting it, and auditing a central claim's complete intervention set rather than a convenient sample is as much this paper's contribution as the audited number itself. We discussed the limitations of the proposed system in Appendix 11.

[1]Du et al.'s main experiments use the same-model debate; their appendix includes a small heterogeneous ChatGPT-Bard experiment.

Reproducibility Statement

All model weights, prompts, evaluation and annotation code, the prospective analysis-plan record for our confirmatory test, the complete gate-audit ledger with annotation provenance, and raw per-question results underlying every table is included in supplementary materials, including: exact checkpoint and quantization identifiers; dataset revision, question identifiers, and selection order; calibration-set membership, historical accuracy values, and reviewer-score mapping; prompts, chat templates, and decoding parameters for every generation and judging stage; seed assignment, execution order, software versions, and tested hardware configuration; cached model outputs, raw judge responses, parser versions, annotation provenance, and adjudication logs; and the scripts regenerating every reported table and figure.

AI Disclosure

In this work, we used generative AI tools (a LLM) only to improve the language and grammar of the manuscript. We used the tool to correct grammatical errors, and make the wording clearer and more readable. We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate or prove mathematical claims, propose or refine hypotheses, design the methodology or experiments, implement methods, generate synthetic data, clean or reformat the dataset, or interpret the results. Tasks such as translation and support for qualitative or thematic analysis are not applicable to this work. All ideas, analyses, and conclusions presented here are entirely our own. Every sentence edited with the help of the model was read and checked by the authors to confirm that the meaning of the original text was preserved and that no unintended content was introduced. We take full responsibility for the final content of this work, including the text, claims, and artifacts presented.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Alfonso Amayuelas, Xianjun Yang, Antonis Antoniades, Wenyue Hua, Liangming Pan, and William Yang Wang. Multiagent collaboration attack: Investigating adversarial attacks in large language model collaborations via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6929–6948, 2024. doi: 10.18653/v1/2024.findings-emnlp.407.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.474.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11733–11763. PMLR, 2024.

Chen Feng, Minghe Shen, Ananth Balashankar, Carsten Gerner-Beuerle, and Miguel R. D. Rodrigues. Noisy but valid: Robust statistical evaluation of LLMs with imperfect judges. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://iclr.cc/virtual/2026/poster/10008079.

Xiyan Fu and Wei Liu. How reliable is multilingual LLM-as-a-judge? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11040–11053, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.587.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. A survey on LLM-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024. URL https://arxiv.org/abs/2411.15594v1.

Rajarshi Haldar and Julia Hockenmaier. Rating roulette: Self-inconsistency in LLM-as-a-judge frameworks. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24986–25004, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.1361.

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

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. doi: 10.1145/3571730.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.397.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More agents is all you need. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=bgzUSZ8aeg.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.992. URL https://aclanthology.org/2024.emnlp-main.992/.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229.

Priya Pitre, Naren Ramakrishnan, and Xuan Wang. CONSENSAGENT: Towards efficient and effective consensus in multi-agent LLM interactions through sycophancy mitigation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22112–22133, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.1141.

Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Zengqing Wu and Takayuki Ito. The hidden strength of disagreement: Unraveling the consensus-diversity tradeoff in adaptive multi-agent systems. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15277–15297, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.772. URL https://aclanthology.org/2025.emnlp-main.772/.

Andrea Wynn, Harsh Satija, and Gillian Hadfield. Talk isn't always cheap: Understanding failure modes in multi-agent debate. *arXiv preprint arXiv:2509.05396*, 2025.

Binwei Yao, Chao Shang, Wanyu Du, Jianfeng He, Ruixue Lian, Yi Zhang, Hang Su, Sandesh Swamy, and Yanjun Qi. Peacemaker or troublemaker: How sycophancy shapes multi-agent debate. *arXiv preprint arXiv:2509.23055*, 2025.

Hangfan Zhang, Zhiyao Cui, Jianhao Chen, Xinrun Wang, Qiaosheng Zhang, Zhen Wang, Dinghao Wu, and Shuyue Hu. Stop overvaluing multi-agent debate – we must rethink evaluation and embrace model heterogeneity. *arXiv preprint arXiv:2502.08788*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623, 2023.

5 Appendices

6 Hyperparameters and System Setup

Table 2 shows every generation and evaluation hyperparameter used in this paper's implementation. Sampling temperature is set low (0.3) for the primary and reviewer calls, where we want a consistent, evaluable answer rather than diverse sampling; self-consistency's own sampling step uses a higher temperature (0.7) specifically because diversity across samples is what the method requires. All automatic-judge calls use temperature 0.0 (greedy decoding), since a judge's verdict should not vary run to run for identical input. The gate's reviewer-score threshold is the same value introduced in Equation 1; both it and the significance level are repeated here as a single reference point.

Table 2: Generation and evaluation hyperparameters.

Parameter	Value
Primary generation temperature	0.3
Reviewer critique temperature	0.3
Self-consistency sampling temperature	0.7
Automatic judge temperature	0.0
Primary max tokens	512
Reviewer max tokens	256
Synthesis max tokens	300
Judge max tokens	10
Judge retry bound on non-conforming output	1
Self-consistency samples (n)	3
Model-loading random seed	42
Gate reviewer-score threshold ($\bar{s}(q)$, Eq. 1)	0.5
Significance level (α, McNemar's test)	0.05

6.1 Hardware and Software Requirements

All experiments ran on a single shared, multi-tenant server; GPU availability at any given time depended on other users' concurrent jobs, which we account for by pinning each model to whichever GPU has sufficient free memory at load time rather than a fixed device assignment. Table 3 shows the hardware and software stack.

Table 3: Hardware and software environment.

Component	Specification
GPUs	4 × NVIDIA RTX A5000 (24 GB VRAM each), shared server
Model quantization	4-bit (GGUF format)
Inference backend	llama.cpp via llama-cpp-python 0.3.35, Vulkan GPU backend
Python	3.10
Environment management	conda
Key Python libraries	pandas, openpyxl, matplotlib

7 Inference-Call Accounting

We report cost in terms of LLM inference calls, not runtime: the same call count can take substantially different wall-clock time on a shared, contended server (Table 3) depending on concurrent load, and we do not report timing figures for this reason (Appendix 11). Call counts are also not equivalent to token cost or GPU-time.

Producing the audited quantities (Algorithm 1). For one question, generating a_p, both reviews, and the single v1 synthesis answer costs

$$C_{v1+\text{gate}} = 4$$

calls: one primary generation, two reviews, one v1 synthesis. Constructing the gated answer a_G (Equation 2) adds no further generation call; it reuses either the already-cached primary answer or the already-cached v1 answer, never issuing a new, differently-prompted call. This is the specific efficiency property that makes the

controlled-replay construction practical: independently generating separate primary/review/synthesis chains for the gate and for v1 (rather than reusing one shared chain) would cost $2 \times 4 = 8$ calls for the identical comparison.

The gate decision (Equation 1). Evaluating $G(q)$ is $O(1)$ given the already-computed reviewer scores $s_1(q), s_2(q)$ and the precomputed historical accuracies w_p, w_1, w_2 – a threshold comparison, adding no generation calls.

Standalone gated deployment. The audit cost above should not be read as the gate's deployment cost. Deployed alone (not compared against v1), the gate needs the primary answer and both reviews unconditionally, plus one synthesis call only when not bypassed:

$$C_{\text{gate}} = 3 + \mathbf{1}\{G(q) = \texttt{synthesize}\}.$$

Automatic judging. Each answer requiring a correctness verdict costs one judge call, plus at most one bounded retry on a non-conforming response (Section 3.6), so judging n answers costs at most $2n$ calls in the worst case, $O(n)$ regardless.

Semantic-equivalence voting (Appendix 8). For $n_{\text{samples}} = 3$, checking all pairwise comparisons explicitly (rather than inferring one relationship transitively from the other two) costs $\binom{3}{2} = 3$ judge calls. For a general sample count m, exhaustive pairwise checking costs $\binom{m}{2} = O(m^2)$ calls; quadratic growth is the specific price of avoiding the transitive-chaining error described in Section 8, and is a real scaling limitation if this mechanism were extended to more than 3 samples per question.

Human annotation. Annotation cost scales with the number of answer records shown to annotators, not with the full evaluated question set. The gate audit's complete-coverage property (non-bypassed questions contribute zero by construction, Section 3.7) is precisely what let us audit $m = 24$ altered questions (48 answer records) rather than requiring human judgment on the underlying $n = 200$ evaluated questions, an $O(m)$ annotation cost where $m \ll n$, in contrast to the primary-vs-v1 pilot's outcome-enriched sample (Appendix 10), which has no equivalent structural shortcut and cannot audit its full comparison at the same cost.

8 A Rejected Fix: Semantic-Equivalence Voting

TruthfulQA's free-text answers make self-consistency's standard extracted-answer vote (Wang et al., 2023) inapplicable. We built a semantic-equivalence voting mechanism designed to avoid known pitfalls: it compares candidates directly, never consulting a gold answer; for a 3-sample vote it checks all three pairwise comparisons independently rather than inferring a third relationship transitively; and it defines an explicit handling rule for every outcome, including flagging genuine non-consensus rather than resolving it arbitrarily. Validated against 30 independently human-labeled pairwise comparisons (10 questions, 3 dependent pairs each), agreement was 15/30 (50.0%), with errors in both directions. We did not adopt this mechanism for any reported result, reporting it as evaluated and found unreliable for this configuration rather than a general claim against semantic-equivalence voting.

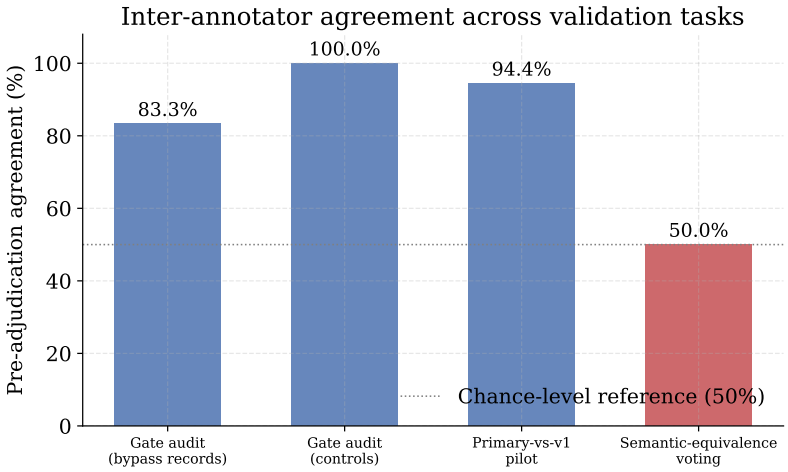

Figure 4: Agreement across the three human-validation tasks in this paper; not all statistics are directly comparable. The gate-audit and primary-vs-v1 bars are pre-adjudication *human–human* inter-annotator agreement. The semantic-equivalence bar is *human–machine* agreement between the human-labeled pairs and the automatic judge's verdicts, a different task with a different denominator; we report it here for reference without characterizing it against a chance baseline, since no well-defined chance level is established for this 3-class task.

9 Prospectively Specified Confirmatory Test

Using Algorithm 1 on a first held-out 100-question TruthfulQA slice (never used in earlier development), we found v1 (always synthesize) significantly outperformed the primary alone. Concerned this might be one of several comparisons examined post hoc, we prospectively specified a single confirmatory test, primary vs. v1 only, McNemar's exact test (Equation 3), $\alpha = 0.05$ on a second, disjoint, previously untouched slice, before collecting that slice's data. Table 4 shows both slices' descriptive statistics and the paired-difference test.

Table 4: Prospectively specified confirmatory-test episode: descriptive statistics and the paired significance test for both slices, under automatic scoring. Mean and SD are computed over each slice's binary per-question correctness indicator ($n = 100$ each) and are descriptive statistics, not an uncertainty estimate for the accuracy itself. Δ is the paired primary-to-v1 difference; its SE and 95% margin of error use the normal approximation in Equation 4, an approximate interval distinct from the unresolved-label bounds reported for our human-adjudicated central result, which reflect annotation ambiguity rather than sampling variability.

Slice	Mean $\pm$ SD (primary)	Mean $\pm$ SD (v1)	$\Delta\pm$ MoE (95%)	McNemar p	Result
Exploratory (Slice 1)	0.500 ± 0.503	0.630 ± 0.485	0.130 ± 0.106	0.029	Significant
Confirmatory (Slice 2)	0.470 ± 0.502	0.500 ± 0.503	0.030 ± 0.090	0.664	Not significant

The confirmatory test was not significant. Under automatic scoring, the exploratory slice showed a primary-to-v1 accuracy difference of 0.130, whereas the prespecified confirmatory slice showed a difference of 0.030 and did not reject the null hypothesis under the exact McNemar test ($p = 0.664$). The confirmatory paired interval, 0.030 ± 0.090 (i.e. $[-0.060, 0.120]$), includes zero and allows both negative and positive effects;

it does not include the exploratory slice's point estimate. These results do not establish a confirmatory improvement, nor does the difference between their significance outcomes establish that the underlying effects differ between slices; both are labeled results using automatic correctness labels, and the human-adjudicated central result (Section 3.7) does not retroactively validate all 200 automatic labels used in this analysis. We report this sequence in full, an exploratory positive signal, a prospectively specified test that did not confirm it, and a precise (non-overclaiming) statement of what that non-confirmation does and does not establish, as we consider the discipline itself a contribution, independent of what it ultimately implies about synthesis.

10 Supporting Result: A Bounded Primary-vs-Synthesis Pilot

Separately, we audited whether unconditional synthesis (v1) improves over the primary model alone, without a gate. Because primary and synthesized answers are not guaranteed to share identical text outside a controlled intervention set, this cannot exploit the same complete-coverage shortcut; we instead use a 20-question outcome-enriched sample (15 automatically-discordant, 5 concordant; 14 questions from slice 1 and 6 from slice 2). Following the same protocol (36 newly annotated records, pre-adjudication agreement 34/36 = 94.4% on those 36; 4 further v1-side judgments reused directly from the gate audit's ledger for the questions shared between the two audits), we find:

$$\Delta \in [0.150, 0.300] \tag{7}$$

(6 corrections, 3 damages, 8 unchanged, 3 unresolved pairs, each constrained to $\{0, +1\}$ by a resolved primary-side answer). Both bounds are positive, directionally opposite the gate's own audited effect. Because the sample is outcome-enriched by design, this interval is not an estimate of the effect across the full 200-question comparison; it is evidence that, within this selected sample, adjudicated judgment found synthesis correcting more errors than it introduced, and a reminder that our central finding concerns the specific static gate, not synthesis in general.

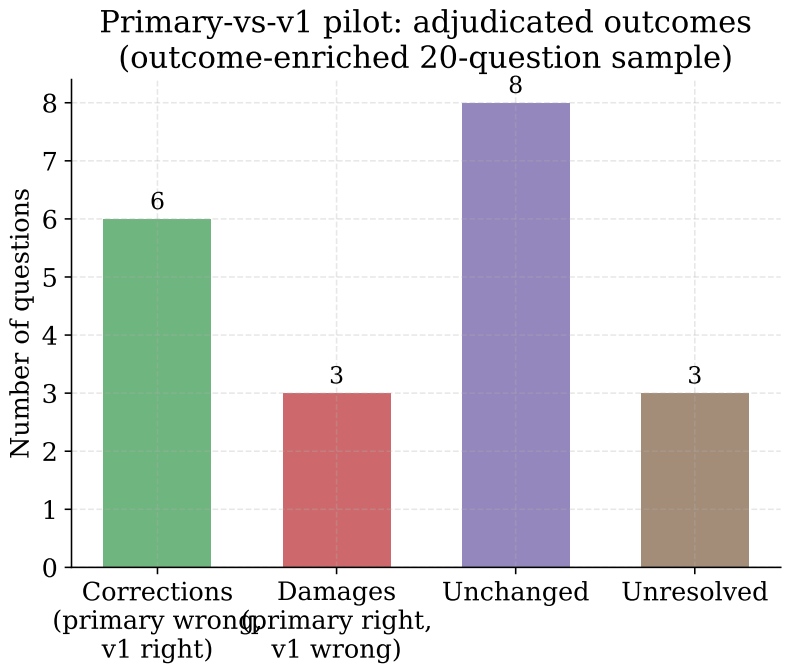

Figure 5: Adjudicated outcomes for the 20-question primary-vs-v1 pilot. Unlike the gate audit (Figure 2), corrections outnumber damages here, but this sample was deliberately outcome-enriched, so this is not directly comparable to a random sample of the full comparison.

11 LIMITATIONS

We evaluated three quantized (4-bit) open-weight models (7–9B parameters) on a single shared server; our central gate result concerns one static gate design on one benchmark, with no claim about other gating policies, model families, or benchmarks. The primary-vs-v1 pilot is bounded to its outcome-enriched sample and does not establish a full-sample effect, and our semantic-equivalence voting mechanism was evaluated and rejected, not repaired. We report no HaluEval accuracy results, having found its evaluator unreliable; a documented evaluator-reliability finding, not a claim that TriGuard fails on HaluEval. We report raw percent agreement, not a chance-corrected measure such as Cohen's κ, and verified cross-run determinism only for the specific sequence and hardware tested, since a fixed seed does not guarantee reproducibility across GPU routing or floating-point summation order. Finally, Table 1 is a design comparison rather than a controlled experimental one, and inference cost is reported as LLM call counts (Appendix 7), not wall-clock time, since the latter depended on a shared, contended server.