BIMETAJUDGE: BIDIRECTIONAL METAMORPHIC AUDITING AND SELECTIVE RISK CONTROL FOR LLM JUDGES

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are increasingly used as judges. Repeated sampling estimates local decoding variability, and position swaps expose one presentation artifact, but neither establishes whether a judge tracks a known change in answer quality. We present BiMetaJudge, a black-box audit built around a *graded bidirectional response curve*: matched nuisance controls estimate spurious movement, while independently verified mild/severe degradations and partial/full repairs measure directional elasticity. A development-fitted risk model drives a staged policy whose score contains only purchased probes, and a disjoint split certifies the final accepted set. Stable-but-wrong cases are an evaluation slice rather than a claimed new phenomenon. We evaluate the complete pipeline on MetaJudge-MR, spanning four domains, five judge configurations, and 12,000 base comparisons. On the held-out test split, BiMetaJudge reaches AUROC 0.824 versus 0.808 for the strongest non-BiMeta baseline and stable-but-wrong recall 0.795 versus 0.280. Leave-one-domain and leave-one-judge gains remain +2.67 and +2.68 AUROC points over nuisance-only evidence.

1 INTRODUCTION

LLM-as-a-Judge now supports leaderboards, data selection, and alignment pipelines Zheng et al. (2023). Consider one ordinary decision in such a pipeline. Candidate A is fluent and detailed but contains a decisive factual error; candidate B is concise and correct. The judge selects A. To reduce uncertainty, the operator calls it five more times and swaps the candidate order. Every call still selects A. A conventional dashboard now displays near-perfect repeat and position consistency, so the wrong decision looks *more* trustworthy precisely because the judge never notices its own blind spot. Even in the favorable independent-call model, if the judge repeats the wrong label with probability $q = 0.95$, a five-sample majority is wrong with probability $\sum_{k=3}^{5} \binom{5}{k} q^k (1-q)^{5-k} = 0.99884$; shared weights, prompts, and knowledge can correlate real errors even more strongly. The easily missed distinction is that consistency asks whether a judge repeats a decision, whereas reliability asks whether the decision tracks answer quality.

This failure mode is documented in recent work. EMNLP 2025 formalizes *self-consistent errors*, finds that four detector families struggle on them, and proposes a cross-model hidden-state probe Tan et al. (2025a). Selective LWE reports 340 incorrect cases among 999 position-consistent VL-RewardBench judgments and 117 among 863 position-consistent MMRewardBench judgments Jwa et al. (2026). Building on this evidence, we ask a different question: can a black-box audit detect such failures by testing whether judgments respond in the expected direction to verified changes in answer quality?

Return to candidate A. If we insert a verifier-confirmed defect—for example, make its code fail a unit test—a quality-sensitive judge should weaken its preference for A. If we repair the original factual error, its preference should strengthen. One binary corruption is weak evidence: the edit may be too small, accidentally change style, or exploit a transformation artifact. We therefore measure a matched curve around the original item: nuisance edits should cause near-zero movement, mild and severe degradations should cause increasingly negative movement, and partial and full repairs should cause increasingly positive movement. Failure on both sides, after subtracting nuisance movement,

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.

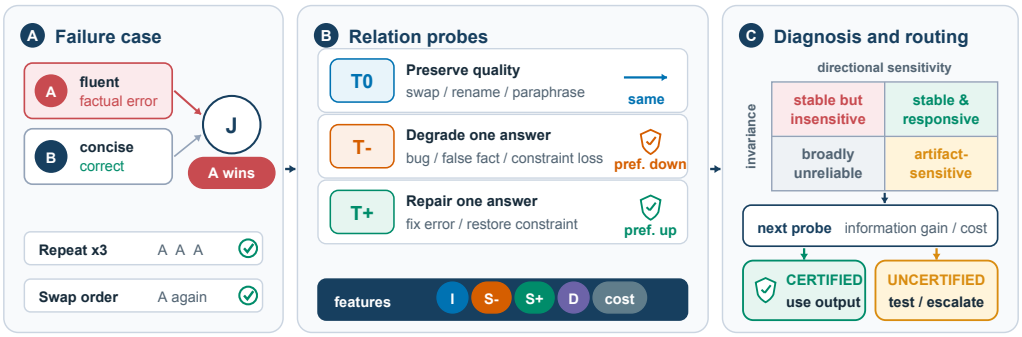


Figure 1: **BiMetaJudge conceptual pipeline.** T_0, T_-, and T_+ denote invariant, degradative, and reparative relations, respectively.

is evidence of quality insensitivity rather than mere stochasticity. We call this *graded bidirectional metamorphic auditing*.

This closes the loop. Low nuisance invariance diagnoses presentation sensitivity; low directional sensitivity diagnoses failure to track substantive quality. A development-fitted model converts these observations into instance-level error risk. Rather than applying every probe to every item, a sequential policy purchases the expected uncertainty reduction per unit cost, then accepts only judgments falling in regions certified on an independently held-out split; all others continue testing or escalate.

Our contributions are:

- We adapt established self-consistent error to a preregistered, judge-specific stable-but-wrong slice, disjoint from the probes used to detect it, and state the information missing from same-input sampling and nuisance-only consistency.
- We define a nuisance-normalized, graded bidirectional response representation that combines response magnitude, dose ordering, and the minimum elasticity across degradation and repair, producing instance-level evidence rather than only aggregate bias rates.
- We develop a no-peeking staged policy: every routing score is refit for the probes available at that stage, and escalation is learned from paired development outcomes. A disjoint split certifies the final automatic acceptance set.
- We evaluate the complete pipeline on MetaJudge-MR, a four-domain meta-evaluation benchmark with disjoint development, certification, and test partitions.

2 Related Work

Judge bias and relation-based audits. MT-Bench documents position, verbosity, and self-enhancement biases Zheng et al. (2023). CALM automates principle-guided nuisance modifications Ye et al. (2025), while JudgeBench and CodeJudgeBench stress-test difficult preference pairs and domain-specific perturbations Tan et al. (2025b); Jiang et al. (2026). Focused audits further isolate position, epistemic-marker, instruction-following, and choice-supportive effects Shi et al. (2025); Lee et al. (2025a); Murugadoss et al. (2025); Zhuang et al. (2025). Controlled quality changes likewise test whether metrics and judges follow known degradation levels: factual corruptions Gatti Pinheiro et al. (2025), graded semantic damage in Meta-Judge Eigler et al. (2026), and targeted trait degradation Bhat & Varma (2026). These studies motivate the two axes in Figure 1: invariance to nuisance changes and monotonic response to verified quality changes. Neither perturbation auditing nor controlled degradation is new. Our contribution is to combine independently verified nuisance, degradation, and repair relations as *instance-level* black-box error evidence for adaptive acquisition and selective certification.

Sampling and exploration. Self-consistency aggregates diverse reasoning paths Wang et al. (2023); judgment distributions, hidden-state probes, and judge-focused uncertainty studies estimate

instability Wang et al. (2025); Radharapu et al. (2026); Haldar & Hockenmaier (2025); Xie et al. (2025). Such signals can expose variable cases but cannot distinguish concentrated correctness from concentrated systematic error. Most directly, Tan et al. (2025a) uses hidden states from an external verifier to detect self-consistent errors. Debate and RL-style exploration diversify trajectories Chan et al. (2024), but correctness is identified only when an independent reward or relation evaluates them. Self-consistency can also interact adversely with position bias Byerly & Khashabi (2026), and adaptive-stability debate allocates deliberation from observed stability Hu et al. (2025). BiMetaJudge instead tests black-box responses to independently verified directional relations.

Judge improvement and risk control. Checklists and explicit planning improve judge reasoning Lee et al. (2025b); Saha et al. (2025). Selective LWE triggers test-time updates from position-order inconsistency and reports consistent-but-incorrect subsets Jwa et al. (2026); unlike its self-generated feedback, BiMetaJudge predicts per-instance error from verified relations and may abstain. Cost-aware routing and RACER allocate stronger judges under uncertainty, cost, and shift Xu et al. (2025c); Zhang et al. (2026). SCOPE combines position-averaged preference entropy with conformal selective risk control and is a direct baseline for our invariance and certification components Badshah et al. (2026). Conformal intervals, rating indeterminacy, no-gold evaluation, and non-transitivity expose ambiguity in validation targets Sheng et al. (2025); Guerdan et al. (2025); Xu et al. (2025a;b). Our question is whether verified, graded two-sided response provides transferable evidence beyond position entropy, unidirectional degradation, and nuisance panels before selection or routing.

Other directions. Pairwise debiasing, reasoning-aware evaluators, and representation alignment modify the judge itself Yang et al. (2025); Chiang et al. (2025); Lai et al. (2025). Scalable oversight, judge benchmarks, small reasoning judges, and agent evaluation broaden task and capacity regimes Kenton et al. (2024); Zhou et al. (2025); Bi et al. (2026); Zhuge et al. (2025). Protocol analyses show that null models, preference leakage, and human–LLM gaps can inflate or obscure apparent judge quality Zheng et al. (2025); Li et al. (2026); Polo et al. (2025). These directions are complementary: our target is black-box, instance-level risk for a fixed judge.

The supplementary material provides the component-level novelty boundary and reported empirical anchors.

3 Problem Formulation

Let an evaluation instance be $x = (q, r, a, b)$, consisting of a user query q, optional reference or rubric r, and candidate responses a, b. A black-box judge J produces a preference $Y_J(x) \in \{a, b, \text{tie}\}$ and optional rationale. Human raters define an admissible response set $H(x) \subseteq \{a, b, \text{tie}\}$; $|H(x)|$ may exceed one under rating indeterminacy. The judge error is

$$E_J(x) = \mathbb{1}\{Y_J(x) \notin H(x)\}. \tag{1}$$

We observe only calls to J and transformations of x. The objective is to estimate the risk of $E_J(x) = 1$, decide whether to accept the automatic judgment, and minimize evaluation cost.

3.1 Reproducible Stable-But-Wrong Definition

Stable-but-wrong (SBW) is an evaluation label, not a BiMetaJudge score. For each judge J and base instance x, we freeze the following event before examining test results. Let $Y_{J,o,s}(x)$ denote the identity-normalized judgment from order $o \in \{\text{AB}, \text{BA}\}$ and sampling replicate s. Identity normalization maps the displayed candidate labels back to the original candidates, so choosing displayed "A" after an answer swap is recorded as the original candidate b.

Definition 1 (Stable-but-wrong event)**.** *An instance is stable-but-wrong for judge J if all of the following hold:*

1. *the original greedy judgment is wrong: $Y_{J,\text{AB},0}(x) \notin H(x)$;*
2. *at least three of four order-randomized calls, mapped back to the original candidate identities, favor the same wrong option;*

3. *repeated sampling is stable: at least four of five identity-normalized samples under the frozen default order return the same label as* $Y_{J,\mathrm{AB},0}(x)$.

For subjective tasks, condition (1) uses the admissible response set $H(x)$ rather than a forced single gold label. For objective tasks, $H(x)$ is derived from the verifier or reference label. Quality-changing probes (T^- and T^+) are deliberately excluded from the SBW ground-truth definition; they are predictors used by BiMetaJudge and baselines. This separation prevents the method from defining the failure mode using the same interventions it later claims to detect.

All SBW-threshold choices are frozen on development data. The primary operating point is recall on the SBW subset at a 10% false-positive rate among non-SBW instances, where the score threshold is selected on the development split and applied unchanged to certification and test splits. We also report the SBW prevalence denominator, the number of SBW positives, bootstrap 95% confidence intervals over base instances, and sensitivity analyses for three-of-five and five-of-five stability thresholds.

3.2 What Repeated Sampling Can and Cannot Identify

For a fixed prompt, order, and judge, same-input decoding replicates are draws $Y_s \sim p_J(\cdot \mid x)$. They estimate local output variability, not whether the modal label belongs to $H(x)$. The majority-vote example in the introduction is the finite-sample consequence: voting reinforces the modal label whether it is correct or not. Repeated sampling therefore remains a useful feature and baseline, but low entropy is not an error certificate.

RL-style exploration is not a counterexample. Exploration diversifies actions or reasoning trajectories and learns from a reward. It can repair a search failure when a valid verifier or human reward identifies better trajectories; in that case the independent reward supplies the correctness information. If the same misspecified judge scores all explored trajectories, exploration can instead optimize its blind spot. Our interventions play a different role: their expected output relations are fixed by programmatic checks or human validation independently of J.

Let $C(J, x)$ be any statistic computed exclusively from whether repeated judgments on quality-preserving copies of x agree.

This non-identifiability is formalized in Appendix A. The point is an established limitation rather than a novelty claim: repeated sampling, position consistency, and unanimous panels can assign maximum confidence to a stable systematic error. Directional quality interventions introduce additional information, subject to the validity of their expected relations.

3.3 Bidirectional Metamorphic Relations

For each base instance x, we construct transformations with auditable expected relations.

Definition 2 (Nuisance invariance)**.** *A transformation T_0 preserves all evaluation-relevant content. A valid judge should satisfy*

$$M_0(J, x, T_0) = \mathbb{1}\{\nu(Y_J(T_0 x)) = Y_J(x)\} = 1, \tag{2}$$

where ν maps renamed or swapped candidate identities back to the original identities.

Definition 3 (Quality monotonicity)**.** *Let T_a^- replace candidate a with a verifiably degraded version while leaving b fixed. If $s_J(a, b)$ denotes the judge's scalar preference for a, a valid judge should satisfy $s_J(T_a^- x) \le s_J(x)$. For a verified repair T_a^+, it should satisfy $s_J(T_a^+ x) \ge s_J(x)$.*

For categorical judges, monotonicity is defined on the ordered preference set $b \prec \mathrm{tie} \prec a$. We record violation indicators V_0, V_-, V_+ and margins when probabilities or scores are available.

Graded response representation. Let T_1^-, T_2^- be independently verified mild and severe degradations and let T_1^+, T_2^+ be partial and full repairs. Orient every margin so that a positive change follows the verified quality direction:

$$\Delta_k^- = s_J(x) - s_J(T_k^- x), \qquad \Delta_k^+ = s_J(T_k^+ x) - s_J(x). \tag{3}$$

For nuisance transforms, N is the mean absolute margin change. We summarize the curve by the directional mean D, the two-sided floor $F = \min\{(\Delta_1^- + \Delta_2^-)/2, (\Delta_1^+ + \Delta_2^+)/2\}$, the nuisance-normalized contrast $C = D - N$, and dose violations $\mathbb{1}\{\Delta_2^- + \tau < \Delta_1^-\}$ and $\mathbb{1}\{\Delta_2^+ + \tau < \Delta_1^+\}$. A large one-sided response cannot compensate for a near-zero F; this is the operational distinction from a degradation-only audit. Missing or invalid transformations remain explicit features and are never silently scored as satisfied.

4 The BiMetaJudge Framework

4.1 Intervention Library

The default library mirrors Panel B of Figure 1: T_0 covers position swaps, candidate renaming, rubric paraphrases, and neutral formatting; $T_{1:2}^-$ introduces mild/severe arithmetic errors, executable bugs, constraint deletions, or unsupported facts; and $T_{1:2}^+$ partially or fully repairs those defects. Quality direction is never accepted solely because another LLM asserts it. Objective relations use identity maps, calculators, tests, source entailment, or rule checks; subjective relations require two independent annotations with adjudication. Transformations without a defensible relation are excluded, and acceptance and observed-outcome rates are exported by type.

4.2 Reliability Representation

After t calls, the state is

$$z_t = [I_t, \Delta_{1:2}^-, \Delta_{1:2}^+, F_t, C_t, D_t, m_t, c_t], \tag{4}$$

where I_t is nuisance invariance, D_t is stochastic disagreement, m_t is the original preference margin when available, and c_t is accumulated cost. A lightweight calibrator $g_\theta(z_t)$ estimates $p_t = \Pr(E_J(x) = 1 \mid z_t)$. We fit g_θ only on a development split. The black-box default is ridge logistic regression.

The two-dimensional view (I, S) supports diagnosis beyond a scalar confidence:

- high I, high S: stable and responsive;
- low I, high S: artifact-sensitive;
- high I, low S: stable but insensitive, our target failure mode;
- low I, low S: broadly unreliable.

4.3 Cost-Aware Sequential Decision

At each stage, the policy chooses an action from nuisance tests, negative tests, positive tests, repeated sampling, strong-judge escalation, or human escalation. On development data, we estimate the expected entropy reduction $\widehat{\Delta H}(a \mid z_t)$ and use

$$a_t = \arg\max_{a \in \mathcal{A}(z_t)} \frac{\widehat{\Delta H}(a \mid z_t)}{\mathrm{cost}(a)}. \tag{5}$$

The policy stops automatically only in a state region certified as sufficiently low-risk. Otherwise it continues until the budget is exhausted and then escalates.

Crucially, one full-panel model cannot be reused at every intermediate stage: doing so would let the router read outcomes it has not paid to obtain. We fit separate development models after the base call, nuisance pair, degradation pair, and repair pair. The implementation uses frozen quantile gates as an auditable approximation to the information-gain rule and learns a paired escalation-value score from development outcomes of the ordinary and strong judges. Test-time decisions use only the latest available stage score, probe-purchase indicators, and task stratum.

Sequential procedure. Given instance x, judge J, fixed policy π, certified regions $\mathcal{R}$, and budget B: (1) query $J(x)$ and initialize the history; (2) compute state z_t after every observation; (3) if $z_t \in \mathcal{R}$, return the aggregated judgment and its accepted-set certificate; (4) if the budget is exhausted, escalate to the strong judge or a human; otherwise (5) select $a_t \leftarrow \pi(z_t)$ by expected information

gain per cost, execute it, update the state, and repeat. This procedure is intentionally simple so that all policy decisions can be logged and audited.

4.4 Finite-Sample Selective-Risk Certificate

The development data D_{dev} are used to fit the calibrator, policy, aggregation rule, and candidate stopping regions. We freeze all choices before evaluating an independent certification set D_{cert}.

For region r, let m_r certification examples be automatically accepted and let k_r disagree with the human response set. Let $U_{\mathrm{CP}}(k_r, m_r, \eta)$ denote the one-sided Clopper–Pearson upper confidence bound Clopper & Pearson (1934). We certify r only if

$$U_{\mathrm{CP}}(k_r, m_r, \delta/|\mathcal{R}_0|) \leq \epsilon, \tag{6}$$

where $\mathcal{R}_0$ is the finite set of candidate regions fixed using development data.

Accepted-set certificate. Under exchangeability between certification and deployment examples, fixed policy/regions, mutually exclusive accepted regions, and correct human response sets, a Clopper–Pearson coverage argument plus a union bound gives accepted-set risk at most ϵ with probability at least $1 - \delta$ over D_{cert}. The formal statement and proof are in Appendix B.

The guarantee is deliberately limited: it controls average error among accepted instances under exchangeability, not correctness of every individual decision. Distribution shift, flawed transformations, or erroneous human response sets can invalidate it.

5 Experimental Design

This section describes the evaluation protocol. The four evaluation settings cover preference agreement, hard objective judging, perturbation robustness, and uncertainty routing Zheng et al. (2023); Tan et al. (2025b); Ye et al. (2025); Xu et al. (2025c). Logistic risk models are fit only on development records, routing thresholds only on certification records, and test labels never enter score construction or threshold selection.

Evaluation goals. The evaluation tests three claims in sequence: (i) judges exhibit stable-but-wrong errors across domains and configurations; (ii) verified two-sided response adds instance-level evidence beyond uncertainty, consistency, nuisance, and degradation-only signals, with transfer and placebo checks; and (iii) that evidence improves fixed-budget auditing and supports cost-aware selective decisions whose certificate behavior can be stress-tested under shift.

5.1 Data and Construction

We evaluate four domain strata:

1. **Open-ended preference**: preference pairs with higher ambiguity;
2. **Summarization**: source–summary pairs with factuality relations;
3. **Mathematical reasoning**: abstract solution pairs with symbolic corruption and repair labels;
4. **Code**: abstract program pairs with executable-bug and repair labels.

The benchmark contains 12,000 base comparisons, 48,000 nuisance variants, and 48,000 quality variants: mild/severe degradation and partial/full repair. The five judge configurations yield 60,000 judge–instance and 480,000 judge–probe rows. Each domain contributes 3,000 base instances, divided into disjoint 20%/20%/60% development, certification, and test splits.

To avoid hiding breadth in one average, Figure 3 reports every domain–judge cell (4 domains $\times$ 5 configurations), retaining failure denominators and the matched AUROC difference from nuisance-only evidence. Table 4 adds paired intervals and transformation-availability rates by domain.



Figure 2: **Held-out evidence with all operating thresholds fixed off test.** (A) Error-detection AUROC (blue) and SBW recall at a development-frozen 10% false-positive rate (orange) for representative detectors on a shared 0–1 scale. (B) Relation-aligned mean preference change for correct and erroneous original judgments across mild/severe degradation and partial/full repair. (C) AUROC for nuisance-only, within-cell shuffled, aligned in-domain, leave-one-domain-out, and leave-one-judge-out evidence; the dashed line marks nuisance-only and the axis is intentionally focused. (D) SBW recall at a frozen 10% review budget versus J3-equivalent evidence cost; P is ordinary-error precision in the reviewed queue.

5.2 Judges and Baselines

We evaluate five judge configurations—J1-8B, J2-14B, J3-32B, J4-70B, and J5-Frontier. Transformation validators and probe calls can fail, and the released records retain invalid, timeout, and refusal statuses.

We compare confidence scores, token margins, repeated-sampling self-consistency, position consistency, conformal and probe scores, a SCOPE-motivated position-entropy selector, nuisance-only evidence, a degradation-only audit, and BiMetaJudge. All methods use the same observations and evaluation protocol. Supplementary Table S1 maps each component to prior work.

5.3 Metrics and Statistical Tests

Primary metrics are error-detection AUROC and AUPRC, stable-but-wrong recall at a development-set threshold targeting 10% false-positive rate, error and SBW yield under a certification-frozen 10% review budget, selective risk, agreement, and J3-equivalent cost. AUROC differences use a 320-replicate paired bootstrap that resamples base instances with all five judge records together; domain AUROCs use 140 replicates. Generalization removes domain/judge indicators and refits on development rows excluding one complete domain or configuration. The placebo shuffles all directional features within domain–judge cells 24 times, preserving marginal distributions while destroying instance alignment. Certificate behavior is assessed over 100 repeated deployments per scenario.

6 Experimental Results

6.1 Overall Error Detection

The test split contains 36,000 judge-instance records and 8,424 errors. Of these, 625 satisfy the frozen SBW definition: 1.7% of all records and 7.4% of errors. Moreover, decisions that are jointly repeat- and position-stable still have a 13.1% error rate. SBW occurs in all 20 domain–judge cells, establishing that the target failure is present across the evaluated settings.

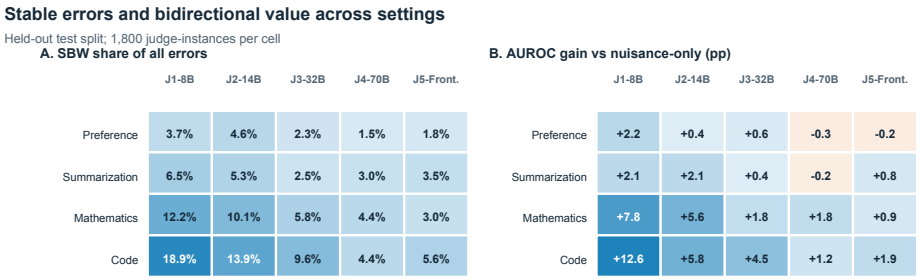


Figure 3: **Scenario-level breadth.** (A) SBW share of errors. (B) Matched-cell AUROC gain over nuisance-only evidence in percentage points; negative changes are orange.

The controlled comparison uses the same test instances, split, observable signals, fitting protocol, and development-frozen 10% false-positive operating point for every method. BiMetaJudge reaches AUROC 0.824 and AUPRC 0.532, versus 0.808 and 0.469 for the strongest non-BiMeta baseline on those metrics. Its SBW recall is 0.795, versus 0.280 for the metric-wise strongest baseline.

Table 1 makes the method-wise effects explicit; Figure 2 links overall detection (A), graded response (B), transfer versus a shuffled placebo (C), and fixed-budget value (D).

Breadth, transfer, and falsification. BiMetaJudge gains +2.58 AUROC points over nuisance-only evidence [+2.22, +2.94]. Gains occur in 17/20 cells and remain +2.67 and +2.68 points under leave-one-domain and leave-one-judge evaluation, whereas within-cell shuffling reduces AUROC to 0.772 and SBW recall to 0.060. Figure 3 and Table 4 expose the cell-level heterogeneity and relation availability behind these aggregates.

Where the aggregate gain comes from. Table 4 localizes the benefit: code and mathematics gain 5.1 and 3.5 AUROC points in every judge cell, whereas preference gains 0.5 points in three cells. Figure 3 retains the negative cells rather than hiding them in the mean. Validator acceptance and observed-outcome rates of 91.0% and 89.1% keep missing and invalid probes visible.

Audit, routing, and ablations. At a frozen 10% review budget, BiMetaJudge retrieves 76.5% of SBW records versus 22.6%, with 56.0% ordinary-error precision. Near 70% coverage, adaptive acquisition costs 5.22 versus 6.32 for the fixed panel at similar risk; Tables 2–5 separate the policy, component, and definition-sensitivity checks. Certificate violations rise from 3.0% in-domain to 41.0% under controlled shift at tolerance 8%.

Method	AUROC ↑	AUPRC ↑	SBW R ↑
Uncertainty and consistency signals			
Verbal confidence	0.691	0.375	<u>0.280</u>
Self-consistency	0.771	0.428	0.000
Position consistency	0.603	0.286	0.000
Judgment distribution	0.744	0.415	0.232
Conformal/probe	0.797	0.459	0.000
Position-entropy selector	0.797	0.459	0.000
Metamorphic evidence			
Nuisance-only panel	0.798	0.457	0.000
Degradation-only audit	<u>0.808</u>	<u>0.469</u>	0.066
BiMetaJudge	**0.824**	**0.532**	**0.795**
Δ ours − best base (pp)	**+1.54**	**+6.30**	**+51.52**

Table 1: **Controlled comparison on the same 36,000 test judge-instances.** Bold marks BiMetaJudge, underline marks the strongest non-BiMeta value, and the final row gives their difference in percentage points. SBW recall uses the development-frozen 10% false-positive operating point.

Policy	Cov.	Risk	Agr.	Cost×	Esc.
Single	100.0	22.4	77.6	1.00	0.0
Strong	100.0	16.8	83.2	5.36	100.0
Position entropy	69.9	12.7	83.9	**2.62**	30.1
Fixed bidir.	70.0	**11.6**	83.9	6.32	30.0
BiMeta adaptive	69.8	11.7	**84.0**	5.22	30.2

Table 2: **Selective policy comparison.** Coverage, risk, agreement, and escalation are percentages; Cost× is J3-equivalent cost. Bold metric cells compare policies near 70% coverage.

Fixed-budget trade-off. At about 70% coverage, adaptive acquisition matches the fixed panel's risk (11.7% versus 11.6%) while reducing cost from 6.32 to 5.22 (17.4%). Position entropy costs 2.62 but raises risk to 12.7%; this is a cost–risk trade-off, not universal dominance.

Variant	AUROC	SBW R	Calls
Full BiMetaJudge	0.824	0.795	3.59
− nuisance relations	0.825	0.802	3.02
− degradation relations	0.811	0.106	3.25
− improvement relations	0.808	0.066	3.45
− sequential policy	0.824	0.795	5.72
− response sets	0.823	0.802	3.59

Table 3: **Component ablation.** Every variant is refit on development data and evaluated on unchanged test labels; Calls is mean evidence calls.

What the ablations isolate. Removing degradation or improvement relations drops SBW recall from 0.795 to 0.106 or 0.066, so both directions matter. Nuisance relations diagnose presentation sensitivity but do not improve headline discrimination here. Without the sequential policy, detection is unchanged while calls rise from 3.59 to 5.72, isolating acquisition efficiency.

Definition and deployment sensitivity. Tightening the repeat criterion from three-of-five to five-of-five reduces the SBW set from 1,054 to 214 records, while BiMetaJudge remains ahead of the strongest listed baseline and recall rises from 0.657 to 0.888 (Appendix C). Certificate violations instead rise from 3.0% in-domain to 41.0% under shift, making the exchangeability boundary explicit.

Evidence chain. The experiments establish the claims in sequence: SBW appears under the frozen definition in every evaluated cell; same-instance comparisons, paired intervals, held-out transfer, and a shuffled placebo tie the gain to aligned bidirectional relations; and the audit and policy results translate discrimination into review yield and cost. Negative cells and shift violations delimit this claim rather than implying universal or distribution-free reliability.

7 Limitations and Ethical Scope

Validity depends on correct metamorphic relations and response sets, which may share a judge's blind spot. Confidence intervals do not guarantee performance under shift, and the certificate controls accepted-set risk only under exchangeability. Provider and human-review costs can also reverse policy ordering. High-stakes or culturally dependent use therefore requires human oversight.

8 Conclusion

BiMetaJudge tests whether a black-box judge is invariant to presentation yet responsive to graded degradation and repair. Across four domains, the complete pipeline shows heterogeneous gains, placebo collapse, and useful audit trade-offs.

Reproducibility Statement

The submission includes the experimental protocol, split construction, metric definitions, bootstrap procedure, and the source code needed to regenerate the reported tables and figures from the experimental records. The main experiments use disjoint development, certification, and test splits; all thresholds reported in test results are fixed before test evaluation.

AI Use Statement

AI tools were used to assist with drafting, editing, related-work retrieval, code organization, and formatting checks. The authors are responsible for the paper content, claims, experiments, and final submission.

References

Sher Badshah, Ali Emami, and Hassan Sajjad. SCOPE: Selective conformal optimized pairwise LLM judging. *arXiv preprint arXiv:2602.13110*, 2026. URL https://arxiv.org/abs/2602.13110.

Savita Bhat and Vasudeva Varma. What makes the whole? probing attribute-level compositionality in LLM judges. In *Second Workshop on Compositional Learning: Safety, Interpretability, and Agents*, 2026. URL https://openreview.net/forum?id=xd9qriSZQh.

Zhenyu Bi, Gaurav Srivastava, Yang Li, Swastik Roy, Meng Lu, Morteza Ziyadi, and Xuan Wang. JudgeBoard: Benchmarking and enhancing small language models for reasoning evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30076–30084, 2026.

Adam Byerly and Daniel Khashabi. Self-consistency falls short! the adverse effects of positional bias on long-context problems. *Transactions of the Association for Computational Linguistics*, 14:292–317, 2026.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. ChatEval: Towards better LLM-based evaluators through multi-agent debate. In *International Conference on Learning Representations*, 2024.

Cheng-Han Chiang, Hung-yi Lee, and Michal Lukasik. TRACT: Regression-aware fine-tuning meets chain-of-thought reasoning for LLM-as-a-judge. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 2934–2952, 2025.

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Lukáš Eigler, Jindřich Libovický, and David Hurych. LLM as a meta-judge: Synthetic data for NLP evaluation metric validation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 1417–1435. Association for Computational Linguistics, 2026. URL https://aclanthology.org/2026.acl-srw.125/.

Giovanni Gatti Pinheiro, Sarra Gharsallah, Adèle Robaldo, Mariia Tokareva, Ilyana Guendouz, Raphaël Troncy, Paolo Papotti, and Pietro Michiardi. Peut-on faire confiance aux juges ? validation de méthodes d'évaluation de la factualité par perturbation des réponses. In *Actes de l'atelier Évaluation des modèles génératifs (LLM) et challenge 2025 (EvalLLM)*, pp. 228–252. ATALA & ARIA, 2025. URL https://aclanthology.org/2025.jeptalnrecital-evalllm.19/.

Luke Guerdan, Solon Barocas, Kenneth Holstein, Hanna Wallach, Zhiwei Steven Wu, and Alexandra Chouldechova. Validating LLM-as-a-judge systems under rating indeterminacy. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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*, pp. 24986–25004, 2025.

Tianyu Hu, Zhen Tan, Song Wang, Huaizhi Qu, and Tianlong Chen. Multi-agent debate for LLM judges with adaptive stability detection. *arXiv preprint arXiv:2510.12697*, 2025.

Hongchao Jiang, Yiming Chen, Yushi Cao, Hung-yi Lee, and Robby T. Tan. CodeJudgeBench: Benchmarking LLM-as-a-judge for coding tasks. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19416–19448. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.888. URL https://aclanthology.org/2026.acl-long.888/.

Seungyeon Jwa, Daechul Ahn, Reokyoung Kim, Dongyeop Kang, and Jonghyun Choi. Becoming experienced judges: Selective test-time learning for evaluators. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 697–721. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-short.50. URL https://aclanthology.org/2026.eacl-short.50/.

Zachary Kenton, Noah Y. Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah D. Goodman, and Rohin Shah. On scalable oversight with weak LLMs judging strong LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Peng Lai, Jianjie Zheng, Sijie Cheng, Yun Chen, Peng Li, Yang Liu, and Guanhua Chen. Beyond the surface: Enhancing LLM-as-a-judge alignment with human via internal representations. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Dongryeol Lee, Yerin Hwang, Yongil Kim, Joonsuk Park, and Kyomin Jung. Are LLM-judges robust to expressions of uncertainty? investigating the effect of epistemic markers on LLM-based evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 8962–8984, 2025a.

Yukyung Lee, JoongHoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. CheckEval: A reliable LLM-as-a-judge framework for evaluating text generation using checklists. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 15771–15798, 2025b.

Dawei Li, Renliang Sun, Yue Huang, Ming Zhong, Bohan Jiang, Jiawei Han, Xiangliang Zhang, Wei Wang, and Huan Liu. Preference leakage: A contamination problem in LLM-as-a-judge. In *International Conference on Learning Representations*, 2026.

Bhuvanashree Murugadoss, Christian Poelitz, Ian Drosos, Vu Le, Nick McKenna, Carina Suzana Negreanu, Chris Parnin, and Advait Sarkar. Evaluating the evaluator: Measuring LLMs' adherence to task evaluation instructions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19589–19597, 2025.

Felipe Maia Polo, Xinhe Wang, Mikhail Yurochkin, Gongjun Xu, Moulinath Banerjee, and Yuekai Sun. Bridging human and LLM judgments: Understanding and narrowing the gap. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Bhaktipriya Radharapu, Eshika Saxena, Kenneth Li, Chenxi Whitehouse, Adina Williams, and Nicola Cancedda. Calibrating LLM judges: Linear probes for fast and reliable uncertainty estimation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics: Industry Track*, pp. 193–216, 2026.

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. Learning to plan and reason for evaluation with thinking-LLM-as-a-judge. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Huanxin Sheng, Xinyi Liu, Hangfeng He, Jieyu Zhao, and Jian Kang. Analyzing uncertainty of LLM-as-a-judge: Interval evaluations with conformal prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 11286–11328, 2025.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in LLM-as-a-judge. In *Proceedings of IJCNLP-AACL 2025*, pp. 292–314, 2025.

Hexiang Tan, Fei Sun, Sha Liu, Du Su, Qi Cao, Xin Chen, Jingang Wang, Xunliang Cai, Yuanzhuo Wang, Huawei Shen, and Xueqi Cheng. Too consistent to detect: A study of self-consistent errors in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4755–4765. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.emnlp-main.238. URL https://aclanthology.org/2025.emnlp-main.238/.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. JudgeBench: A benchmark for evaluating LLM-based judges. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9e720fce64f91114c49cfd640d821da3-Abstract-Conference.html.

Victor Wang, Michael JQ Zhang, and Eunsol Choi. Improving LLM-as-a-judge inference with the judgment distribution. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

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

Qiujie Xie, Qingqiu Li, Zhuohao Yu, Yuejie Zhang, Yue Zhang, and Linyi Yang. An empirical analysis of uncertainty in large language model evaluations. In *International Conference on Learning Representations*, 2025.

Shengwei Xu, Yuxuan Lu, Grant Schoenebeck, and Yuqing Kong. Benchmarking LLMs' judgments with no gold standard. In *International Conference on Learning Representations*, 2025a.

Yi Xu, Laura Ruis, Tim Rocktäschel, and Robert Kirk. Investigating non-transitivity in LLM-as-a-judge. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025b.

Zhenghao Xu, Qin Lu, Qingru Zhang, Liang Qiu, Ilgee Hong, Changlong Yu, Wenlin Yao, Yao Liu, Haoming Jiang, Lihong Li, Hyokun Yun, and Tuo Zhao. Ask a strong LLM judge when your reward model is uncertain. In *Advances in Neural Information Processing Systems*, volume 38, 2025c.

Haoyan Yang, Runxue Bao, Cao Xiao, Jun Ma, Parminder Bhatia, Shangqian Gao, and Taha Kass-Hout. Any large language model can be a reliable judge: Debiasing with a reasoning-based bias detector. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V. Chawla, and Xiangliang Zhang. Justice or prejudice? quantifying biases in LLM-as-a-judge. In *International Conference on Learning Representations*, 2025.

Wenbo Zhang, Lijinghua Zhang, Liner Xiang, and Hengrui Cai. Reasoning is not free: Robust adaptive cost-efficient routing for LLM-as-a-judge. *arXiv preprint arXiv:2605.10805*, 2026. URL https://arxiv.org/abs/2605.10805.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric 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, 2023.

Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. Cheating automatic LLM benchmarks: Null models achieve high win rates. In *International Conference on Learning Representations*, 2025.

Yilun Zhou, Austin Xu, Peifeng Wang, Caiming Xiong, and Shafiq Joty. Evaluating judges as evaluators: The JETTS benchmark of LLM-as-judges as test-time scaling evaluators. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Nan Zhuang, Boyu Cao, Yi Yang, Jing Xu, Mingda Xu, Yuxiao Wang, and Qi Liu. LLM agents can be choice-supportive biased evaluators: An empirical study. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 26436–26444, 2025.

Mingchen Zhuge, Changsheng Zhao, Dylan R. Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-judge: Evaluate agents with agents. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

A PROOFS

Proposition 1 (Non-identifiability from consistency)**.** *For any deterministic correct judge J^+, there exists a deterministic inverted judge J^- such that $C(J^+, x) = C(J^-, x) = 1$ for every x, while $E_{J^+}(x) = 0$ and $E_{J^-}(x) = 1$ whenever $H(x)$ is a singleton.*

Proof. Define J^- to return the opposite of J^+ on every binary-preference instance. Both judges return the same answer across repeated or quality-preserving copies, so both have maximal consistency. Their correctness is opposite whenever only one answer is admissible. Thus no consistency-only statistic identifies correctness without an additional assumption connecting stability to truth. □

B SELECTIVE-RISK CERTIFICATE

Theorem 1 (Accepted-set risk control)**.** *Assume certification and deployment examples are exchangeable, the policy and candidate regions are fixed independently of D_{cert}, the deployed stopping rule assigns every accepted example to exactly one mutually exclusive region, and human response sets are observed without error. With probability at least $1 - \delta$ over D_{cert}, every certified region r satisfies*

$$\Pr(E_J(X) = 1 \mid Z(X) \in r) \leq \epsilon. \tag{7}$$

Consequently, the mutually exclusive union of certified regions has accepted-set risk at most ϵ.

Proof. For each candidate region r, Clopper–Pearson coverage implies that the binomial error rate among examples accepted into r is at most $U_{\mathrm{CP}}(k_r, m_r, \delta/|\mathcal{R}_0|)$ with probability at least $1-\delta/|\mathcal{R}_0|$. A union bound over the finite, development-fixed candidate set $\mathcal{R}_0$ gives simultaneous coverage for all certified regions. Since the deployed accepted regions are mutually exclusive, the accepted-set risk is a mixture of region-conditional risks, each bounded by ϵ; the mixture is therefore also at most ϵ. □

C ADDITIONAL EMPIRICAL DETAILS

Domain	SBW / errors	ΔAUROC [CI]	Wins	Valid / obs.
Preference	49/1637	+0.5 [+0.1,+1.0]	3/5	89.2/87.2
Summarization	83/1869	+1.2 [+0.6,+1.8]	4/5	88.9/86.9
Mathematics	175/2191	+3.5 [+2.8,+4.3]	5/5	93.0/91.3
Code	318/2727	+5.1 [+4.2,+6.1]	5/5	93.1/90.8
All	625/8424	+2.6 [+2.2,+2.9]	17/20	91.0/89.1

Table 4: **Domain-level evidence strength.** SBW/errors gives the denominator; ΔAUROC is BiMetaJudge minus nuisance-only evidence in percentage points with a paired 95% interval. Wins counts positive judge cells; Valid/obs. are percentages.

Repeat	SBW n	Err. share	BiMeta R	Best-base R
3/5	1054	12.5	**0.657**	0.407 (Self-cons.)
4/5	625	7.4	**0.795**	0.280 (Verbal conf.)
5/5	214	2.5	**0.888**	0.243 (Verbal conf.)

Table 5: **Repeat-threshold sensitivity.** Error share is the percentage of all errors satisfying SBW; R is recall at the same development-frozen operating point.