All Verdicts are Not Equal: Rethinking LLM Judge Reliability

Vineet Kumar
PayPal Artificial Intelligence
PayPal, Bengaluru, India
vkumar32@paypal.com

Darshita Rathore
PayPal Artificial Intelligence
PayPal, Bengaluru, India
drathore@paypal.com

Anindya Moitra
PayPal Artificial Intelligence
PayPal, Bengaluru, India
amoitra@paypal.com

Abstract

LLM-as-a-Judge is the standard paradigm for NLP evaluation, yet its systemic reliability remains poorly understood despite being widely treated as a deterministic ground truth. We present a comprehensive reliability audit, stress-testing six frontier models across four benchmarks, five prompt formats, two presentation orders, three sampling temperatures, and ten repetitions per condition. Our empirical analysis reveals severe vulnerabilities: verdicts change across identical replications at temperature zero, position-order swaps flip the majority of verdicts on challenging tasks, and the most deterministic judge achieves perfect consistency by trivially repeating incorrect verdicts, agreeing with ground truth only 51% of the time. To formalize these multi-faceted failure modes, we introduce the trustworthy verdict rate (T), a unified metric capturing the joint probability that an evaluation is reproducible, order-invariant, and accurate. Using T, we derive a theoretical upper bound on accuracy imposed by position bias and show that reliability is item-specific rather than model-level. Finally, we demonstrate that shifting from pairwise win-rate to holistic rubric scoring improves trustworthiness more than any single-format prompting intervention, offering an actionable framework for robust NLP evaluation.

1 Introduction

LLM-as-a-Judge has become the dominant evaluation paradigm in modern NLP. It powers RL reward signals (Wen et al., 2026), model-comparison (Dubois et al., 2023), automated benchmark scoring, and an increasing fraction of the evaluation pipelines on which downstream model selection now depends (Zheng et al., 2023; Tan et al., 2025). Across all of these settings, a judge's output is treated as a stand-in for ground truth: the verdict is what gets logged, aggregated, and optimized against.

This treatment carries an implicit assumption that the verdict is a stable function of the inputs. That re-querying the same judge with the same pair yields the same answer, and that swapping which response is shown first does not change the preferred one. Both assumptions fail in practice. In our audit, 7–17% of verdicts change across identical replications, and up to 98% of verdicts flip when the presentation order of the two candidates is swapped. The most striking case in our data is a judge that returns the same verdict on 100% of repeated calls maximally deterministic by every single-shot diagnostic in common use yet whose verdicts agree with ground truth only 51% of the time, barely above chance. Determinism, in other words, is not reliability.

We address this with two complementary contributions. First, a **cross-cutting empirical audit**: we stress-test six frontier judges across four benchmarks spanning subjective and objective preference, standard and adversarial difficulty, at five prompt formats, two presentation orders, three sampling temperatures, and ten repetitions per condition, one of the largest reliability-focused audits of LLM-as-Judge systems we are aware of. The design is deliberately cross-cutting: each axis of variation is held fixed in turn so that its individual contribution to verdict instability can be measured rather than averaged out.

Second, a **formal framework** that consolidates the audit's findings into operationally meaningful quantities. We argue that reliability is irreducibly multidimensional and introduce the *trustworthy verdict rate* as the joint probability that a verdict is reproducible, order-invariant, and correct. The framework yields a hard accuracy ceiling imposed by position bias that no improvement in model capability can exceed, decomposes reliability loss into a noise component and a position-bias component that respond to different interventions, and produces a per-cell scalar that reorders judge compar-

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 4581–4598
November 6-10, 2026 ©2026 Association for Computational Linguistics

isons relative to accuracy whenever judges differ on the underlying axes. We further compare three evaluation paradigms (direct pairwise, FLASK rubric decomposition, and G-Eval holistic scoring) and find that paradigm choice can shift trustworthiness further than any single-format intervention.

Our findings show that accuracy alone is insufficient to characterize LLM judges, single-order protocols mask critical position bias, and common single-shot consistency heuristics fail to catch major systemic failure modes.

The remainder of this paper is structured as follows: we survey related literature in §2, establish our formal multidimensional reliability framework in §3, outline our extensive empirical audit setup in §4, analyze our results and findings in §5, and conclude with actionable recommendations in §6.

2 Related Works

LLMs are used across domains codegen (Kumar et al., 2025), conversational analytics (Rathore et al., 2026), agentic evaluation (Agarwal et al., 2026) and LLM-based evaluation has become a standard alternative to human annotation/evaluation. With MT-Bench (Zheng et al., 2023) showing that GPT-4 judgments can achieve high agreement with human preferences. The paradigm has since been extended to RLHF simulation in AlpacaFarm (Dubois et al., 2023), objective ground-truth evaluation in JudgeBench (Tan et al., 2025), and adversarial robustness in LLM-Bar (Zeng et al., 2024). Recent surveys organize the field around what to judge, how to judge, and how to benchmark (Li et al., 2025). Most of this work reports agreement or accuracy; we instead ask whether a judge's verdict remains reliable under repeated sampling, position swaps, and format changes.

Recent studies show that LLM judges are susceptible to positional bias and presentation effects (Wang et al., 2024; Shi et al., 2025), can be shifted by adversarial phrasing (Raina et al., 2024), and may be overconfident on ambiguous examples despite modest accuracy (Lu et al., 2025). Complementary work finds low intra-rater reliability and overconfident inconsistency in LLM judges (Haldar and Hockenmaier, 2025; Janiak et al., 2025). We build on these observations by formalizing position sensitivity as a flip rate, deriving the induced accuracy ceiling and combining it with stochastic consistency into the Trustworthy Verdict Rate.

Our treatment also connects to uncertainty estimation and structured evaluation. Selective prediction studies how classifiers trade coverage for accuracy by abstaining on uncertain inputs (El-Yaniv and Wiener, 2010), and recent work adapts this idea to LLM judges using conformal prediction with held-out calibration data (Chirkova et al., 2026). Other studies show that evaluation format affects judge behavior: averaging scores can outperform majority vote, chain-of-thought can hurt some judges, and checklist-style rubrics can improve agreement (Sheng et al., 2025; Lee et al., 2025). Our results connect these strands by showing that repeated-query agreement provides a calibration-free confidence signal, while G-Eval-style single-prompt multi-criterion scoring improves T more reliably than decomposed rubric voting.

3 Formalization

3.1 Setup

We evaluate a judge J on a corpus of preference pairs. Each instance comprises two candidate responses (x, y) together with a ground-truth label indicating which is preferred. The judge returns a verdict $V \in \{x, y\}$ for any presentation of the pair.

Reliability is studied along three axes:

- **Stochastic instability**: the verdict varies across independent runs of the same prompt.
- **Position bias**: the verdict varies with the order in which the two candidates are presented.
- **Format sensitivity**: the verdict statistics vary with the prompt template (e.g., direct vs. CoT, JSON vs. XML).

The first two axes act on a single elicitation and enter the trustworthy rate T defined below. The third axis is treated as a *design factor*: we fix a set $\mathcal{F}$ of prompt templates and compute every quantity in §3.2–§3.5 separately for each $f \in \mathcal{F}$. Format sensitivity is then characterized in §3.6 by the dispersion of those per-format quantities for a fixed (judge, dataset).

3.2 Per-Pair Indicators and Population Rates

For each pair, holding the format f and dataset fixed, we define three Bernoulli indicators:

$$N = \mathbb{1}[\text{verdict reproducible across independent runs}]$$
$$P = \mathbb{1}[\text{verdict invariant under candidate-order swap}]$$
$$R = \mathbb{1}[\text{verdict matches ground truth}]$$

The corresponding population rates are directly observable from repeated elicitations:

$$\begin{aligned} C &:= \Pr(N = 1) \quad \text{(consistency rate)} \\ 1 - \varphi &:= \Pr(P = 1) \quad (\varphi \text{ is the position flip rate}) \\ A &:= \Pr(R = 1) \quad \text{(overall accuracy)} \end{aligned}$$

We partition pairs by P into *decisive* ($P = 1$) and *undecisive* ($P = 0$) subsets, letting $A_{\text{dec}} := \Pr(R = 1 \mid P = 1)$ denote the decisive accuracy and $q := A_{\text{undec}} = \Pr(R = 1 \mid P = 0)$ denote the accuracy on undecisive pairs.

3.3 Assumptions

(A1) Undecisive-pair accuracy. In general evaluation protocols involving ties, abstentions, non-binary labels, or unequal order weighting, $q := \Pr(R = 1 \mid P = 0)$ captures baseline correctness on undecisive pairs. Under a forced-binary, equally weighted preference setup, any pair where the verdict flips under candidate swap yields contradictory verdicts across orderings, exactly one of which matches the ground truth. Hence, in our setup, $q = \frac{1}{2}$.[1]

(A2) Axis independence on a pair. Stochastic reproducibility is uninformative about position-decisiveness, and the correctness rate on consistent, decisive pairs equals the decisive accuracy:

$$\Pr(P = 1 \mid N = 1) = \Pr(P = 1)$$

$$\Pr(R = 1 \mid N = 1, P = 1) = A_{\text{dec}}.$$

3.4 A Tight Upper Bound on Observable Accuracy

Decomposing accuracy over the decisive partition,

$$A = (1 - \varphi)\, A_{\text{dec}} + \varphi q, \tag{1}$$

so that

$$A_{\text{dec}} = \frac{A - \varphi q}{1 - \varphi}. \tag{2}$$

Since $A_{\text{dec}} \leq 1$, we obtain the generalized **capability ceiling**

$$\boxed{A \;\leq\; 1 - \varphi(1 - q)} \tag{3}$$

For our forced-binary setup ($q = \frac{1}{2}$), Eq. (3) simplifies to $A \leq 1 - \varphi/2$.

[1]The assumption $q = \frac{1}{2}$ can break under explicit tie-breaking options or "neither" verdicts; however, in our core experiments, the judge produces a forced binary verdict.

The bound is tight and attained iff the judge is perfect on every decisive pair ($A_{\text{dec}} = 1$), and *capability-free*: no improvement in the judge's underlying judgment can lift observable accuracy above $1 - \varphi(1 - q)$ when the position flip rate is φ. We call the residual $\big(1 - \varphi(1 - q)\big) - A$ the **capability gap**, the portion of error orthogonal to position bias.

3.5 The Trustworthy Rate

A verdict is **trustworthy** when it is simultaneously reproducible, robust to candidate-order swap, and correct. The rate at which the judge produces such verdicts is

$$T := \Pr(N = 1 \wedge P = 1 \wedge R = 1).$$

Factoring by the chain rule and applying (A2)[2],

$$\begin{aligned} T &= \Pr(N) \cdot \Pr(P \mid N) \cdot \Pr(R \mid N, P) \\ &= C \cdot (1 - \varphi) \cdot A_{\text{dec}}. \end{aligned}$$

Substituting (2) yields the generalized closed form

$$\boxed{T \;=\; C \cdot \big(A - \varphi q\big)} \tag{4}$$

which simplifies to $T = C \cdot (A - \varphi/2)$ when $q = \frac{1}{2}$.

Interpretation. Equation (4) factors T into two observable terms:

- C: *noise survival*: the fraction of verdicts stable to re-sampling.
- $(A - \varphi q)$: *capability net of undecisive-baseline slack*: the portion of accuracy not borrowed from the baseline q on undecisive pairs.

Two judges with identical overall accuracy A can therefore differ substantially in T when their consistency or position flip rates differ. T is the operationally meaningful quantity: the rate at which a judge produces verdicts that are reproducible, robust to order, and correct.

3.6 Format as a Design Factor

We compute the tuple (C, φ, A, T) separately for every (judge, format, dataset) triple in our experimental grid; each row of Table 1 reports such a tuple. Format sensitivity is then summarized by

[2]In practice, this mutual independence and Assumption 2 (A2) may not hold. See Appendix G for more details. There, we empirically show that even when A2 is violated, T continues to behave as a conservative lower bound.

within-cell dispersion of T across $f \in \mathcal{F}$, holding judge and dataset fixed:

$$\Delta T_{\text{fmt}}(J, D) \coloneqq \max_{f \in \mathcal{F}} T(J, f, D) - \min_{f \in \mathcal{F}} T(J, f, D). \tag{5}$$

A judge with small ΔT_{fmt} is format-robust; one with large ΔT_{fmt} has trustworthy verdicts that depend on a prompt-engineering choice. Reporting T at the cell level lets readers inspect both *level* (which (judge, format) combination is most trustworthy on a given benchmark) and *robustness* (how much that level moves with format), without conflating the two into a single number.

4 Experimental Design

4.1 Datasets

We evaluate on four benchmarks spanning a 2×2 design across subjective vs. objective and standard vs. adversarial settings. MT-Bench (Zheng et al., 2023) and AlpacaFarm (Dubois et al., 2023) represent standard subjective evaluation benchmarks, focusing respectively on crowd-sourced human preferences for open-ended responses and instruction-following preferences. For objective evaluation, we use JudgeBench (Tan et al., 2025), which provides ground-truth labels derived from MMLU-Pro, LiveCodeBench, and LiveBench across 17 domains, and LLMBar (Zeng et al., 2024), an adversarial benchmark containing rule-based response pairs specifically designed to test judge robustness.

4.2 Judges

We evaluate 6 models from 4 providers: **GPT-5.2** and **GPT-4o** (OpenAI), **Gemini-2.5-Pro** and **Gemini-2.0-Flash** (Google), **Claude-Sonnet-4.6** (Anthropic), and **Llama-3.3-70B** (Grattafiori et al., 2024). This spans frontier closed-weight models and a strong open-weight model, with 2 reasoning-specialized models (GPT-5.2, Gemini-2.5-Pro) and 4 standard models.

4.3 Prompt Formats

All formats share a common system prompt and user message template; only the output format instruction varies:

- **Direct**: Single-token verdict (A or B). No rationale.
- **Verdict-first** (JSON / XML): Structured output with verdict *before* rationale the "decide then justify" pattern.
- **CoT** (JSON / XML): Rationale *before* verdict the "think then decide" pattern.

We focus on JSON and XML because they are the dominant structured-generation formats in modern LLM APIs and production pipelines, enabling us to isolate whether formatting conventions alone influence judge behavior (Kumar et al., 2026). This creates a 2×2 sub-design within structured prompting: {JSON, XML} × {verdict-first, CoT}.

Each pair is presented in both AB and BA order. Every (pair, judge, format, order) combination is repeated 10 times. We run at three sampling temperatures: T_s=0 (all 6 judges), T_s=0.5, and T_s=1.0 (4 judges; reasoning models excluded as public API does not support the temperature parameter). Detailed breakdowns of models utilized, and public API cost estimates are provided in Appendix E.

5 Results & Empirical Findings

Table 1 reports stochastic consistency C, position flip rate φ, accuracy A, and Trustworthy Verdict Rate T across six judges, five prompt formats, and four datasets. We organize the findings around the three empirical predictions of the formalization in §3: position bias imposes a hard accuracy ceiling, T separates judges that accuracy conflates, and stochastic consistency alone is insufficient for reliability. We then analyze the sources of variation in T and evaluate whether rubric-based paradigms can mitigate the same failure modes.

5.1 The Position-Bias Ceiling is Tight

Equation (3) predicts that no judge can exceed observable accuracy $1-\varphi/2$ on a dataset with position flip rate φ. This bound is not merely formal: several cells in Table 1 lie on, or within a few points of, the ceiling. The clearest example is Gemini-2.0-Flash with Verdict-first XML on JudgeBench. Its flip rate is $\varphi = 98.0\%$, implying a maximum attainable accuracy of 51.0%, and the observed accuracy is exactly $A = 51.0\%$. Under Equation (4), this yields $T = 2.0\%$: the judge is perfectly reproducible across repeated calls ($C = 100\%$), but its verdicts carry essentially no trustworthy signal beyond order-induced coin-flipping.

The same pattern appears, though less extremely, for Gemini-2.0-Flash with Verdict-first JSON on JudgeBench: $\varphi = 82.0\%$ implies a ceiling of 59.0%, while the observed accuracy is 52.0% and T falls to 10.8%. Figure 1 visualizes this ceiling

Judge	Format	MT-Bench				AlpacaFarm				JudgeBench				LLMBar			
		C	φ	A	T	C	φ	A	T	C	φ	A	T	C	φ	A	T
GPT-5.2	Direct	89.3	5.1	64.8	55.6	82.7	9.2	49.5	37.1	86.0	11.0	64.5	50.7	89.0	9.0	75.5	63.2
	VF – JSON	86.2	7.1	64.3	52.4	82.1	6.1	52.0	40.2	81.0	10.0	65.0	48.6	89.0	9.0	75.5	63.2
	VF – XML	89.8	11.2	65.3	53.6	86.1	9.3	54.1	42.6	82.5	12.0	64.5	48.3	92.0	6.0	76.0	67.2
	CoT – JSON	82.1	6.1	66.8	52.3	80.4	15.5	57.7	40.2	79.0	13.0	70.5	50.6	92.5	13.0	81.5	69.4
	CoT – XML	84.2	8.2	65.8	52.0	81.4	12.4	56.7	41.1	75.5	12.0	68.0	46.8	90.5	11.1	82.4	69.5
GPT-4o	Direct	98.0	11.2	64.8	58.0	95.5	9.1	58.6	51.6	94.5	24.0	51.0	36.9	98.5	8.0	71.5	66.5
	VF – JSON	96.9	11.2	63.3	55.9	96.0	12.1	61.1	52.8	94.0	31.0	54.5	36.7	97.0	7.0	74.5	68.9
	VF – XML	98.0	14.3	62.8	54.5	92.4	14.1	60.1	49.0	93.5	35.0	52.5	32.7	97.0	6.0	76.0	70.8
	CoT – JSON	89.8	19.4	61.2	46.2	92.4	22.2	60.1	45.3	88.0	47.0	54.5	27.3	92.5	25.0	70.5	53.7
	CoT – XML	90.8	16.3	61.2	48.2	92.9	21.4	59.2	45.1	88.0	46.0	53.5	26.8	93.0	19.0	73.5	59.5
Gemini-2.5-Pro	Direct	87.8	15.3	62.8	48.4	84.0	20.6	55.2	37.7	81.5	56.0	63.5	28.9	87.0	20.0	80.0	60.9
	VF – JSON	87.0	16.7	62.2	46.8	87.1	12.4	60.3	47.1	91.1	6.9	88.5	77.5	92.5	13.0	85.0	72.6
	VF – XML	88.8	29.6	59.7	39.9	84.5	40.2	54.6	29.2	83.8	37.4	80.3	51.6	83.0	33.0	78.5	51.5
	CoT – JSON	87.6	7.3	62.7	51.7	84.5	14.4	58.2	43.1	89.3	5.4	88.7	76.8	89.5	9.0	85.5	72.5
	CoT – XML	88.5	15.8	60.4	46.5	82.5	17.5	56.2	39.1	89.9	8.8	89.9	76.9	89.0	17.0	83.5	66.8
Gemini-2.0-Flash	Direct	99.5	16.3	60.7	52.3	99.5	22.2	52.3	41.0	99.0	46.0	55.5	32.2	98.0	23.0	61.5	49.0
	VF – JSON	100.0	40.0	55.4	35.4	99.5	53.5	59.3	32.4	98.5	82.0	52.0	10.8	98.0	47.9	63.8	39.1
	VF – XML	99.0	36.5	59.3	40.6	97.5	30.0	61.5	45.3	100.0	98.0	51.0	2.0	98.5	42.4	64.3	42.5
	CoT – JSON	96.9	17.0	59.7	49.6	97.5	19.0	61.5	50.7	96.0	38.4	56.8	36.1	96.9	38.9	63.4	42.6
	CoT – XML	97.9	26.8	57.4	43.1	98.5	31.3	61.8	45.5	96.0	50.0	58.0	31.7	97.5	39.4	62.3	41.5
Claude-Sonnet-4.6	Direct	99.5	7.7	67.6	63.4	97.9	7.4	54.7	49.9	98.8	14.9	57.3	49.3	99.5	12.2	79.7	73.2
	VF – JSON	98.4	4.4	68.5	65.2	98.5	7.2	57.4	53.0	98.1	12.7	71.8	64.2	99.5	14.1	80.8	73.4
	VF – XML	91.8	5.1	67.3	59.4	96.4	8.2	57.1	51.1	89.9	7.1	83.4	71.8	99.0	12.0	82.5	75.7
	CoT – JSON	89.4	8.7	70.2	58.9	88.2	7.2	59.0	48.9	83.4	7.1	82.1	65.5	95.5	5.1	87.4	81.0
	CoT – XML	84.2	2.0	67.9	56.3	88.8	10.2	60.2	48.9	81.5	3.0	87.0	69.7	92.0	6.0	87.0	77.3
Llama-3.3-70B	Direct	100.0	13.3	60.2	53.6	100.0	16.3	60.7	52.6	100.0	26.0	55.5	42.5	100.0	12.0	63.5	57.5
	VF – JSON	100.0	15.3	62.2	54.6	100.0	17.3	61.2	52.6	100.0	22.4	48.2	37.0	100.0	17.0	64.0	55.5
	VF – XML	100.0	11.2	62.2	56.6	100.0	13.1	59.6	53.1	99.5	37.0	44.0	25.4	99.5	15.0	64.0	56.2
	CoT – JSON	98.0	12.2	62.8	55.6	100.0	17.3	62.4	53.8	99.5	45.5	51.5	28.6	99.0	14.0	66.0	58.4
	CoT – XML	99.5	16.3	61.2	52.8	98.0	22.2	62.1	50.0	99.5	44.0	51.5	29.4	100.0	15.0	66.5	59.0

Table 1: Stochastic consistency C (% deterministic across 10 reps, ↑), order flip rate φ (% verdict reversal on swap, ↓), accuracy A (%, ↑), and trustworthy verdict rate $T = C \cdot (A - \varphi/2)$ (in %, ↑) at T_s=0. *Direct*: single-token verdict. *VF*: verdict before rationale. *CoT*: rationale before verdict. T shaded ≥70 / 60–70 / 50–60 / 40–50 / <40; support columns shaded in muted greys.

across all model–format–dataset cells. Points near the boundary are not primarily capability-limited; their gains require reducing φ, not improving the base model.

5.2 T Reveals Reliability Differences Hidden by Accuracy

Accuracy is the standard scalar for judge quality, but Equation (4) predicts that two judges with similar A can have very different T whenever they differ in stochastic consistency or position sensitivity. Table 1 confirms this in two ways.

First, changing only the output format can alter trustworthiness far more than accuracy suggests. On JudgeBench, Gemini-2.5-Pro improves from $A = 63.5\%, T = 28.9\%$ under Direct prompting to $A = 88.5\%, T = 77.5\%$ under Verdict-first JSON. The 25% accuracy gain is accompanied by a 49% gain in T, because the format change both improves accuracy and reduces position flips from $\varphi = 56.0\%$ to $\varphi = 6.9\%$.

Second, comparable accuracies can mask different reliability profiles. On AlpacaFarm, GPT-5.2 with CoT-JSON and Claude-Sonnet-4.6 with Di-

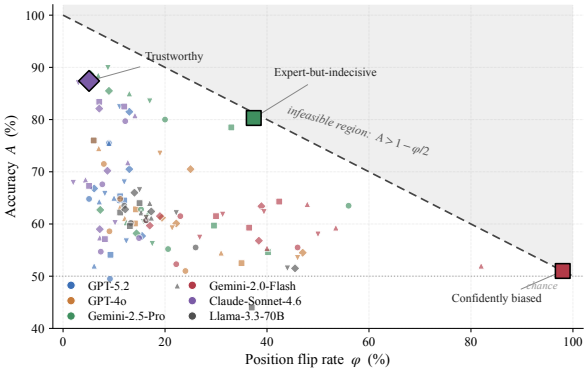


Figure 1: **Accuracy ceiling** view of the Trustworthy Verdict Rate at T_s=0. Each point is one cell of Table 1; the dashed line marks the upper bound $A \leq 1 - \varphi/2$ from Equation (3). The shaded region is mathematically infeasible: position bias imposes an accuracy tax of $\varphi/2$ that no capability improvement can overcome. Points near the ceiling are ceiling-bound; points far below the ceiling, such as several Llama-3.3 cells, still have capability gaps where a stronger judge could help.

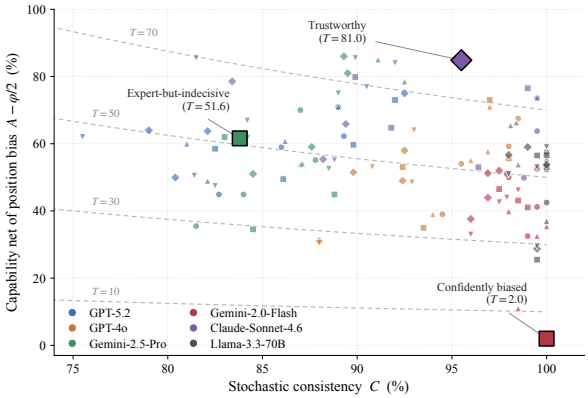


Figure 2: **Iso-T factor space.** Cells from Table 1 plotted in $(C,\ A - \varphi/2)$ coordinates, the two factors of $T = C \cdot (A - \varphi/2)$. Dashed curves are iso-$T$ contours. Highlighted examples show three distinct reliability regimes: *Trustworthy* (Claude-Sonnet CoT-JSON / LLMBar, $T = 81\%$), *Expert-but-indecisive* (Gemini-2.5-Pro VF-XML / JudgeBench, $T = 52\%$), and *Confidently biased* (Gemini-2.0-Flash VF-XML / JudgeBench, $T = 2\%$).

rect prompting achieve similar accuracies (57.7% vs. 54.7%), but their trustworthy rates differ substantially (40.2% vs. 49.9%). Claude's lower flip rate (7.4% vs. 15.5%) means that a larger share of its correct verdicts are reproducibly correct rather than order-contingent.

Figure 2 shows the factorization directly by plotting cells in $(C, A - \varphi/2)$ space. The iso-T contours make clear that trustworthiness can fail through either factor: low stochastic consistency, low bias-adjusted accuracy, or both.

5.3 Consistency Alone Does Not Imply Trustworthiness

A natural shortcut for reliability is to report stochastic consistency C: a judge that returns the same verdict on repeated calls feels dependable. Table 1 shows why this intuition is incomplete. Llama-3.3-70B has $C \geq 98\%$ in every cell, yet its T on JudgeBench ranges only from 25.4% to 42.5% because its flip rate remains high, ranging from 22% to 45%. Gemini-2.0-Flash is even weaker: its C is 96–100% across settings, while T spans 2.0% to 52.3%.

The reason is structural. In Equation (4), C multiplies the bias-adjusted term $(A - \varphi/2)$. When this term is small, even perfect stochastic consistency cannot rescue trustworthiness. A deterministic judge can still be deterministically order-dependent.

5.4 What Drives Variation in T?

The preceding results show that T varies substantially across cells. We next decompose where that variation comes from: judge–dataset interaction, prompt format, and sampling temperature.

Judge–dataset interaction. A three-factor ANOVA of T across the 120 judge–format–dataset cells shows that the largest source of variation is the judge × dataset interaction ($\eta^2 = 28.1\%$), exceeding the judge (27.0%) and dataset (26.2%) main effects (see Appendix C.1). Thus, there is no universally best judge: rankings reorder across benchmarks. A complementary four-facet G-study over pair × judge × format × order further shows that order is a structured source of variance rather than residual noise, validating its explicit inclusion as φ in T (see Appendix C.2).

Format dispersion. The dispersion ΔT_{fmt} from Equation (5) measures how much trustworthiness depends on prompt format rather than judge identity. The largest dispersions concentrate on JudgeBench. Gemini-2.5-Pro spans $\Delta T_{\text{fmt}} = 48.6$ points, from Direct prompting at $T = 28.9\%$ to Verdict-first JSON at $T = 77.5\%$; Gemini-2.0-Flash spans 34.1 points. By contrast, Claude-Sonnet-4.6 on LLMBar varies only from 73.2% to 81.0% ($\Delta T_{\text{fmt}} = 7.8$).

This means that, in high-dispersion settings, a single $(judge, format)$ result is not a stable estimate of the judge's underlying reliability. The format choice can carry as much signal as the model choice. Full details appear in Appendix B, Table 4.

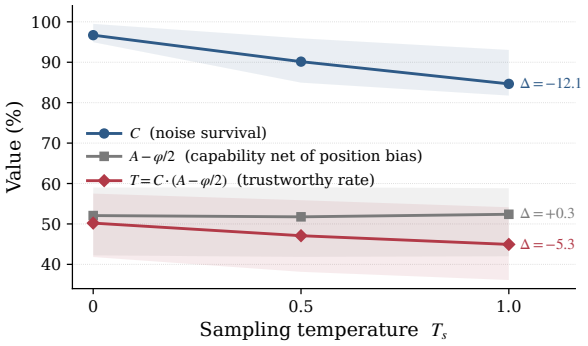


Figure 3: Mean C, $A - \varphi/2$, and T across judge–format–dataset cells as sampling temperature T_s varies from 0 to 1.0. Shaded bands show interquartile range. Temperature reduces C by 12.1 points but leaves $(A - \varphi/2)$ nearly unchanged, so the drop in T is driven by noise survival, as Equation (4) predicts.

Temperature ablation. Sampling temperature should primarily affect the noise-survival term C, not the bias-adjusted capability term $(A - \varphi/2)$. A sweep over $T_s \in \{0, 0.5, 1.0\}$ on the four temperature-tunable judges confirms this prediction. As shown in Figure 3, mean C drops by 12.1 points across the sweep, while $(A - \varphi/2)$ changes by only $+0.3$ points across 48 cells. The resulting decline in T is therefore attributable almost entirely to reduced stochastic consistency, supporting the interpretation that the two factors of T capture distinct reliability properties. The complete temperature sweep analysis is presented in Appendix A.

5.5 Failure Modes and Practical Remedies

Figure 4 organizes judge failures into four regions of (C, φ) space. The *Trustworthy* region is the only regime where the residual gap $1 - \varphi/2 - A$ primarily reflects model capability rather than position bias; closing that gap requires a stronger judge. The *Noisy but fair* region is easier to address: repeated sampling and majority vote can improve reliability because φ is low. The *Chaos* region requires both interventions: average over both candidate orderings to reduce position bias, then ensemble across repeated runs to reduce stochastic noise. The *Confidently biased* region is the most dangerous. Here, high C creates the appearance of reliability while high φ drives T toward zero. The worst observed cell in our study, with $\varphi = 98\%$ and $T = 2.0\%$, falls in this region.

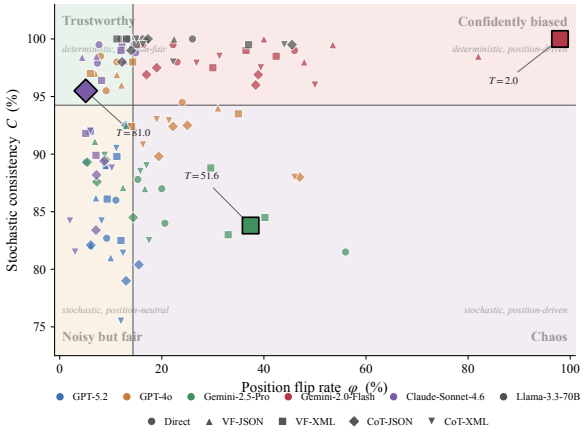


Figure 4: **Failure mode taxonomy.** Judge cells plotted by stochastic consistency C and position flip rate φ. The quadrants distinguish trustworthy, noisy-but-fair, chaotic, and confidently biased judges, each requiring a different mitigation strategy.

Figure 5 aggregates these effects into a per-judge view. Sonnet-4.6 has the highest mean T, but the wide whiskers and overlap across judges show that no model dominates uniformly; reliability depends strongly on the dataset and format pairing.

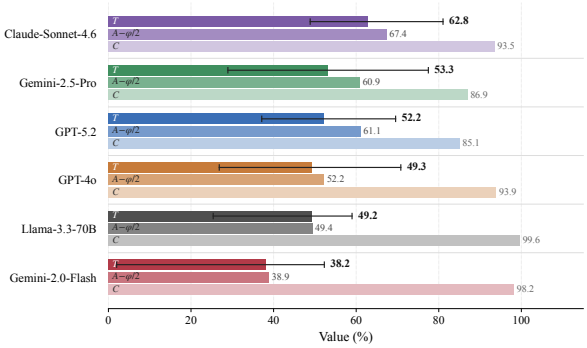


Figure 5: Per-judge mean of C, $A - \varphi/2$, and T averaged across all 20 format–dataset cells per judge, ordered by mean T. Whiskers span the min–max range across cells. Wide ranges reflect judge × (format, dataset) interactions but no judge dominates uniformly; see Appendix C.

5.6 Rubric-Based Evaluation: FLASK and G-Eval

In the preceding sections, we characterize failures of direct pairwise evaluation. Here, we investigate if rubric-based paradigms can mitigate these failures. Specifically, can decomposing judgment into explicit criteria reduce φ and thereby raise the $A_{\max}$ ceiling?

Table 2 compares three paradigms on the two objective benchmarks, JudgeBench and LLMBar: Direct pairwise evaluation with one API call, FLASK-style (Ye et al., 2024) rubric decomposition with four calls per judgment, and G-Eval-style (Liu et al., 2023) score-based evaluation with one call over four scored criteria.

FLASK reduces φ for some judges but collapses C universally (e.g., GPT-5.2: C 86.0% → 34.8% on JudgeBench), leaving T lower than Direct for five of six judges. G-Eval elicits all criterion scores in a single prompt, preserving C while reducing position sensitivity; for capable judges this can lift T substantially (Gemini-2.5-Pro: T 28.9% → 77.9% on JudgeBench, a +49 pp gain exceeding any format-induced swing in Table 4).

6 Conclusion and Recommendations

We have audited six frontier LLM judges across four benchmarks, five prompt formats, two presentation orders, three sampling temperatures, and ten

Judge	JudgeBench – Direct				JudgeBench – FLASK				JudgeBench – G-Eval				LLMBar – Direct				LLMBar – FLASK				LLMBar – G-Eval			
	C	φ	A	T	C	φ	A	T	C	φ	A	T	C	φ	A	T	C	φ	A	T	C	φ	A	T
GPT-5.2	86.0	11.0	64.5	50.7	34.8	9.1	63.6	20.5	68.5	16.0	68.0	41.1	89.0	9.0	75.5	63.2	55.0	21.0	79.5	37.9	88.0	18.0	79.0	61.6
GPT-4o	94.5	24.0	51.0	36.9	44.4	47.5	52.5	12.8	89.0	61.0	62.0	28.0	98.5	8.0	71.5	66.5	53.0	31.0	75.0	31.5	95.5	28.0	78.0	61.1
Gemini-2.5-Pro	81.5	56.0	63.5	28.9	60.6	10.5	67.4	37.7	87.7	10.0	93.8	**77.9**	87.0	20.0	80.0	60.9	63.0	14.0	78.0	44.7	86.0	27.0	79.0	56.3
Gemini-2.0-Flash	99.0	46.0	55.5	32.2	90.4	30.3	57.1	37.9	94.5	40.4	59.0	36.7	98.0	23.0	61.5	49.0	66.0	32.0	65.5	32.7	99.0	39.0	74.0	54.0
Claude-Sonnet-4.6	98.8	14.9	57.3	49.3	69.0	2.0	75.1	51.1	78.7	14.4	82.0	58.8	99.5	12.2	79.7	**73.2**	58.0	11.0	81.0	43.8	91.0	9.1	85.0	**73.2**
Llama-3.3-70B	100.0	26.0	55.5	42.5	96.9	51.0	53.6	27.2	97.0	55.1	52.0	23.7	100.0	12.0	63.5	57.5	97.5	20.0	67.5	56.1	100.0	29.0	78.0	63.5

Table 2: Three-paradigm comparison at T_s=0: Direct pairwise (1 API call), FLASK rubric decomposition (4 calls per judgment), and G-Eval score-based evaluation (1 call with 4 scored criteria). All values in percentage points. $T = C \cdot (A - \varphi/2)$ shaded ≥70 / 60–70 / 50–60 / 40–50 / <40; support columns (C, φ, A) in muted greys. **Bold** = best T per dataset.

repetitions, with an additional comparison of direct, FLASK-decomposed, and G-Eval rubric paradigms on the two objective benchmarks. The audit yields that judge reliability is not what accuracy measures. A verdict is operationally *trustworthy* only when it is simultaneously **reproducible** across runs, **invariant to candidate-order swap**, and **correct**. We formalize this as the *trustworthy verdict rate*, and prove that the same formalization implies a hard capability ceiling that no improvement in model capability can overcome at fixed position flip rate. Both quantities are directly observable from repeated elicitations and require no ground-truth labels beyond what existing benchmarks already provide.

We also argue that this ceiling is empirically tight; i.e., near-deterministic judges can still produce verdicts that are worse than chance; and reliability is item-specific rather than judge-level. The most striking case in our data is a single judge × format × dataset configuration with $C = 100\%$ and $T = 2\%$: a judge that appears maximally reliable by every single-shot diagnostic in common use, but contributes essentially nothing beyond order-induced coin-flipping.

The implication for practice is concrete. As LLM judges become infrastructure for RLHF, leaderboards, automated evaluation pipelines, and downstream model selection — the field needs reliability audits alongside accuracy benchmarks. We recommend that benchmark releases report at minimum (C, φ, A, T) per cell, that judge selection use T rather than accuracy as the comparative scalar, and that protocols which leave φ unobservable (single-order evaluation) be treated as fundamentally under-specified. None of these recommendations require new infrastructure; they require reading existing infrastructure with a richer lens than the one currently in use.

The trustworthy verdict rate is not a competitor to accuracy. It is what accuracy was always meant to mean.

Limitations

Our study has three primary limitations. First, due to computational tractability, we restricted our evaluation to four English-only benchmarks. Consequently, our reliability findings may not generalize to low-resource languages where frontier models exhibit weaker baseline performance, and our estimates for low-frequency phenomena, such as per-domain flip rates, carry minor sampling variance that headline numbers may not surface. Second, we evaluate closed-weight models without access to log-probabilities or internal states, which precludes the analysis of finer-grained token confidence signals. Third, our audit focuses exclusively on pairwise preference data. While pairwise evaluation is currently the most prevalent paradigm, a natural extension of this work would be to investigate whether our findings regarding verdict instability and bias transfer to list-wise ranking or multi-model grading frameworks.

References

Gunja Agarwal, Arup Kumar Das, Arun Menon, Jitesh Chandra Mishra, and Vignesh Divakaran. 2026. Agentworld: Personality-aware reliability evaluation for agentic information retrieval. *Preprint*, arXiv:2608.24076.

Robert L. Brennan. 1992. Generalizability theory. *Educational Measurement: Issues and Practice*, 11(4):27–34.

Nadezhda Chirkova, Tunde Oluwaseyi Ajayi, Seth Aycock, Zain Muhammad Mujahid, Vladana Perlić, Ekaterina Borisova, and Markarit Vartampetian. 2026. LLM-as-a-qualitative-judge: automating error analysis in natural language generation. In *Proceedings of the First Workshop on Multilingual Multicultural Evaluation*, pages 99–132, Rabat, Morocco. Association for Computational Linguistics.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori Hashimoto. 2023. Alpacafarm:

A simulation framework for methods that learn from human feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Ran El-Yaniv and Yair Wiener. 2010. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(53):1605–1641.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Rajarshi Haldar and Julia Hockenmaier. 2025. 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. Association for Computational Linguistics.

Denis Janiak, Jakub Binkowski, Albert Sawczyn, Bogdan Gabrys, Ravid Shwartz-Ziv, and Tomasz Jan Kajdanowicz. 2025. The illusion of progress: Re-evaluating hallucination detection in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34728–34745, Suzhou, China. Association for Computational Linguistics.

Vineet Kumar, Kanishka, and Bhuvanesh Mandora. 2026. Structure tax: How structured output affects llms performance. *Preprint*, arXiv:2610.12056.

Vineet Kumar, Ronald Tony, Darshita Rathore, Vipasha Rana, Bhuvanesh Mandora, . Kanishka, Chetna Bansal, and Anindya Moitra. 2025. Genicious: Contextual few-shot prompting for insights discovery. In *Proceedings of the 8th International Conference on Data Science and Management of Data (12th ACM IKDD CODS and 30th COMAD)*, CODS-COMAD '24, page 405–409, New York, NY, USA. Association for Computing Machinery.

Yukyung Lee, JoongHoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. 2025. 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*, pages 15771–15798, Suzhou, China. Association for Computational Linguistics.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2025. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2757–2791, Suzhou, China. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Junyu Lu, Kai Ma, Kaichun Wang, Kelaiti Xiao, Roy Ka-Wei Lee, Bo Xu, Liang Yang, and Hongfei Lin. 2025. Is LLM an overconfident judge? unveiling the capabilities of LLMs in detecting offensive language with annotation disagreement. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5609–5626, Vienna, Austria. Association for Computational Linguistics.

Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is LLM-as-a-judge robust? investigating universal adversarial attacks on zero-shot LLM assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7499–7517, Miami, Florida, USA. Association for Computational Linguistics.

Darshita Rathore, Vineet Kumar, Vaibhav Singal, Ankur Vivek Singh, and Anindya Moitra. 2026. Conversational query engine for mixed-modality heterogeneous enterprise data sources. *Preprint*, arXiv:2606.28370.

Huanxin Sheng, Xinyi Liu, Hangfeng He, Jieyu Zhao, and Jian Kang. 2025. 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*, pages 11286–11328, Suzhou, China. Association for Computational Linguistics.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2025. Judging the judges: A systematic study of position bias in LLM-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 292–314, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Yuan Tang, Alejandro Cuadron, Chenguang Wang, Raluca Popa, and Ion Stoica. 2025. Judgebench: A benchmark for evaluating LLM-based judges. In *The Thirteenth International Conference on Learning Representations*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. 2026. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *The Fourteenth International Conference on Learning Representations*.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2024. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *The Twelfth International Conference on Learning Representations*.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*.

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. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Appendix Contents

A Temperature Sweep Analysis **11**
B Format Dispersion Analysis **12**
C Variance Decomposition Details **13**
C.1 ANOVA on T 13
C.2 Four-Facet G-Study on Verdict Direction 13
D Prompt Templates **14**
E API Cost **16**
F Dataset Details **17**
G Assumptions underlying T **17**

A Temperature Sweep Analysis

Table 3 reports (C, φ, A, T) across sampling temperatures $T_s \in \{0, 0.5, 1.0\}$ for the four temperature-tunable judges (GPT-4o, Gemini-2.0-Flash, Claude-Sonnet-4.6, Llama-3.3-70B; GPT-5.2 and Gemini-2.5-Pro are excluded because their public APIs do not expose a temperature parameter). Verdict-first and CoT rows aggregate over their JSON and XML variants for compactness, collapsing the 5-format grid of Table 1 to a 3-format view.

T_s	Judge	Format	MT-Bench				AlpacaFarm				JudgeBench				LLMBar			
			C	φ	A	T	C	φ	A	T	C	φ	A	T	C	φ	A	T
$T=0$	GPT-4o	Direct	98.0	11.2	64.8	58.0	95.5	9.1	58.6	51.6	94.5	24.0	51.0	36.9	98.5	8.0	71.5	66.5
		Verdict-first	97.4	12.8	63.0	55.1	94.2	13.1	60.6	50.9	93.8	33.0	53.5	34.7	97.0	6.5	75.2	69.8
		CoT	90.3	17.9	61.2	47.2	92.6	21.8	59.6	45.1	88.0	46.5	54.0	27.1	92.8	22.0	72.0	56.6
	Gemini-2.0-Flash	Direct	99.5	16.3	60.7	52.3	99.5	22.2	52.3	41.0	99.0	46.0	55.5	32.2	98.0	23.0	61.5	49.0
		Verdict-first	99.5	38.2	57.4	38.1	98.5	41.8	60.4	38.9	99.2	90.0	51.5	6.4	98.2	45.2	64.0	40.7
		CoT	97.4	21.9	58.6	46.4	98.0	25.2	61.7	48.1	96.0	44.2	57.4	33.9	97.2	39.2	62.9	42.1
	Claude-Sonnet-4.6	Direct	99.5	7.7	67.6	63.4	97.9	7.4	54.7	49.9	98.8	14.9	57.3	49.3	99.5	12.2	79.7	73.2
		Verdict-first	95.1	4.8	67.9	62.3	97.4	7.7	57.3	52.1	94.0	9.9	77.6	68.3	99.2	13.1	81.7	74.5
		CoT	86.8	5.4	69.0	57.5	88.5	8.7	59.6	48.9	82.5	5.1	84.6	67.7	93.7	5.6	87.2	79.1
	Llama-3.3-70B	Direct	100	13.3	60.2	53.6	100	16.3	60.7	52.6	100	26.0	55.5	42.5	100	12.0	63.5	57.5
		Verdict-first	100	13.3	62.2	55.6	100	15.2	60.4	52.8	99.8	29.7	46.1	31.2	99.8	16.0	64.0	55.9
		CoT	98.7	14.3	62.0	54.1	99.0	19.8	62.3	51.9	99.5	44.7	51.5	29.0	99.5	14.5	66.2	58.7
$T=0.5$	GPT-4o	Direct	91.9	8.6	65.1	55.9	91.9	11.8	56.5	46.5	77.5	24.0	49.5	29.1	97.9	7.4	71.1	66.0
		Verdict-first	93.0	13.5	62.4	51.8	89.0	13.4	59.1	46.6	75.2	29.1	52.9	28.8	95.2	5.6	75.5	69.2
		CoT	84.8	20.1	61.7	43.8	84.4	21.5	59.9	41.5	62.1	44.7	52.5	18.7	89.9	18.4	72.0	56.5
	Gemini-2.0-Flash	Direct	96.3	12.9	60.4	52.0	91.9	24.7	50.0	34.6	85.0	46.0	55.5	27.6	94.9	25.0	62.2	47.2
		Verdict-first	96.0	37.3	57.9	37.7	93.3	43.0	59.4	35.4	95.8	90.0	51.5	6.2	94.6	47.2	64.5	38.7
		CoT	83.2	19.1	58.5	40.7	80.9	24.7	59.7	38.3	62.8	43.0	57.5	22.6	82.3	39.5	63.7	36.2
	Claude-Sonnet-4.6	Direct	98.3	8.0	67.6	62.5	95.1	5.5	54.1	48.8	94.8	16.2	55.2	44.7	95.9	12.6	79.0	69.7
		Verdict-first	94.8	5.1	68.3	62.3	94.9	8.1	57.7	50.9	91.9	9.9	77.6	66.8	99.0	13.7	81.5	73.9
		CoT	84.3	5.1	68.8	55.8	87.0	10.9	58.9	46.5	79.7	6.3	85.2	65.4	93.9	6.7	87.1	78.6
	Llama-3.3-70B	Direct	98.9	9.8	61.4	55.9	98.4	17.2	58.6	49.2	97.5	26.3	55.3	41.1	98.5	11.0	64.0	57.6
		Verdict-first	97.9	11.8	62.7	55.6	97.3	16.7	58.6	48.9	89.2	29.2	46.5	28.5	97.2	16.5	63.8	54.0
		CoT	90.9	13.9	61.2	49.3	90.9	19.9	60.8	46.2	78.5	46.0	51.8	22.6	92.8	15.0	65.8	54.1
$T=1.0$	GPT-4o	Direct	87.1	8.2	66.5	54.4	83.5	8.0	58.0	45.1	58.6	22.2	49.0	22.2	93.0	10.0	72.0	62.3
		Verdict-first	88.1	13.9	63.7	50.0	82.2	10.5	60.5	45.4	58.8	31.3	53.8	22.4	90.2	6.0	74.9	64.9
		CoT	82.0	21.6	62.9	42.7	75.8	18.5	60.2	38.6	50.0	35.9	54.0	18.0	81.0	17.0	72.5	51.8
	Gemini-2.0-Flash	Direct	93.3	15.5	61.3	50.0	88.0	22.0	53.5	37.4	78.0	46.0	55.5	25.4	89.0	25.0	62.0	44.1
		Verdict-first	94.3	37.7	57.9	36.8	88.5	41.0	60.8	35.7	93.7	89.9	52.0	6.6	87.7	47.5	63.1	34.5
		CoT	75.9	19.0	57.3	36.3	63.8	26.5	60.8	30.3	50.8	43.9	59.8	19.2	62.5	40.3	67.8	29.8
	Claude-Sonnet-4.6	Direct	95.1	8.7	66.5	59.1	90.5	8.1	54.8	45.9	89.4	19.5	56.4	41.7	92.5	13.1	79.4	67.4
		Verdict-first	93.2	4.9	67.7	60.8	92.5	8.0	57.2	49.2	89.5	9.7	77.8	65.3	97.5	12.6	81.2	73.0
		CoT	83.6	3.3	68.5	55.9	83.0	8.0	59.0	45.6	77.5	4.6	85.0	64.1	90.2	6.0	87.5	76.2
	Llama-3.3-70B	Direct	99.0	12.4	60.8	54.1	96.0	17.0	61.0	50.4	93.0	28.0	54.5	37.7	98.0	12.0	63.5	56.4
		Verdict-first	94.6	14.0	62.8	52.8	94.8	15.0	60.8	50.5	83.7	31.1	46.4	25.8	93.2	17.0	64.0	51.7
		CoT	89.4	13.5	62.7	50.0	89.2	19.0	62.0	46.8	71.8	44.0	51.0	20.8	90.2	15.5	65.5	52.1

Table 3: Stochastic consistency C, position flip rate φ, accuracy A, and trustworthy rate $T = C \cdot (A - \varphi/2)$ (all %) across sampling temperatures $T_s \in \{0, 0.5, 1.0\}$. Verdict-first and CoT rows are averages over JSON and XML format variants. T shaded ≥70 / 60–70 / 50–60 / 40–50 / <40; support columns shaded in muted greys.

B Format Dispersion Analysis

To quantify the sensitivity of a judge to prompt formatting, we define the *format dispersion* of the trustworthy rate as

$$\Delta T_{\text{fmt}}(J, D) = \max_f T(J, f, D) - \min_f T(J, f, D),$$

where J denotes the judge model, D the benchmark dataset, and f the prompting format. Intuitively, ΔT_{fmt} measures how much the observed trustworthiness of a judge changes solely due to formatting choices while keeping the judge and evaluation data fixed. A low value indicates that the judge is robust to presentation format, whereas a high value indicates strong format dependence.

Table 4 reports $\Delta T_{\text{fmt}}(J, D)$ together with the corresponding worst-case trustworthy rate

$$T_{\min}(J, D) = \min_f T(J, f, D).$$

Judge	MT Bench	AlpacaFarm	JudgeBench	LLMBar	Mean
GPT-5.2	$3.6_{52.0}$	$5.5_{37.1}$	$3.9_{46.8}$	$6.3_{63.2}$	4.8
GPT-4o	$11.8_{46.2}$	$7.7_{45.1}$	$10.1_{26.8}$	$17.1_{53.7}$	11.7
Gemini-2.5-Pro	$11.8_{39.9}$	$17.9_{29.2}$	$48.6_{28.9}$	$21.1_{51.5}$	24.9
Gemini-2.0-Flash	$16.9_{35.4}$	$18.3_{32.4}$	$34.1_{2.0}$	$9.9_{39.1}$	19.8
Claude-Sonnet-4.6	$8.9_{56.3}$	$4.1_{48.9}$	$22.5_{49.3}$	$7.8_{73.2}$	10.8
Llama-3.3-70B	$3.8_{52.8}$	$3.8_{50.0}$	$17.1_{25.4}$	$3.5_{55.5}$	7.1
Mean	9.5	9.6	22.7	10.8	

Table 4: Format-induced dispersion of the trustworthy rate, $\Delta T_{\text{fmt}}(J, D)$, with the worst-case trustworthy rate $T_{\min}$ shown in subscript. Shading indicates dispersion magnitude: <5, 5–10, 10–20, 20–35, and >35. Low ΔT_{fmt} indicates a format-robust judge. All values are reported in percentage points.

Findings

Two broad patterns emerge from Table 4.

Format robustness varies substantially across judges. GPT-5.2 and Llama-3.3-70B exhibit consistently low dispersion across benchmarks, suggesting that their trustworthiness is largely invariant to prompt format. In contrast, the two Gemini variants show substantially larger swings, with mean dispersions of 24.9% and 19.8% respectively. The most extreme case occurs for Gemini-2.5-Pro on JudgeBench, where the trustworthy rate ranges from 28.9% under Direct prompting to 77.5% under VF-JSON, yielding a dispersion of 48.6 points attributable solely to formatting changes.

JudgeBench is markedly more format-sensitive. Average dispersion across judges is approximately 10 points on MT-Bench, AlpacaFarm, and LLMBar, but rises to 22.7% on JudgeBench. Moreover, five of the six largest dispersion values occur on JudgeBench, and the only configuration with $T_{\min} < 10\%$ appears on this benchmark. One possible explanation is that harder or more knowledge-intensive comparisons weaken the judge's underlying confidence signal, thereby amplifying the effect of surface-level prompt presentation. We leave systematic investigation of this hypothesis to future work.

Implications for evaluation reporting. These results suggest that reporting a single (J, f, D) configuration can substantially misrepresent the underlying behavior of a judge, especially on format-sensitive benchmarks. For example, Gemini-2.5-Pro appears among the strongest judges on JudgeBench under VF-JSON prompting, yet among the weakest under Direct prompting. We therefore recommend reporting either the interval $(T_{\min}, T_{\max})$ across a fixed format set or, equivalently, $(T_{\min}, \Delta T_{\text{fmt}})$ when comparing judge reliability across models.

C Variance Decomposition Details

This appendix reports the variance decomposition referenced in §5.4. We pursue two complementary analyses: an ANOVA on the trustworthy rate T (§C.1) and a Generalizability-theory study on verdict direction (§C.2).

C.1 ANOVA on T

Treating each of the 120 (judge, format, dataset) cells at $T_s = 0$ as an observation, we fit a three-factor ANOVA with judge, format, and dataset as factors and their pairwise interactions. Table 5 reports the resulting η^2 values.

Source	η^2 (%)
Judge	27.0
Dataset	26.2
Judge × Dataset	**28.1**
Judge × Format	8.3
Format	0.7
Format × Dataset	1.3
Residual	8.4

Table 5: Variance decomposition (η^2, %) of T across 120 (judge, format, dataset) cells. The judge × dataset interaction is the dominant source.

C.2 Four-Facet G-Study on Verdict Direction

Standard G-theory analyses for LLM judges treat the design as a two-facet (pair × judge) model and collapse format and order into residual variance (Brennan, 1992). This obscures two effects that T directly measures: order sensitivity (mapped to φ) and stochastic noise (mapped to C). We instead estimate a four-facet fully crossed model — pair (p) × judge (j) × format (f) × order (o) — with repetitions nested within cells. Table 6 reports the seven variance components plus residual for verdict direction at $T_s = 0$. Components shown are percentages of the total variance estimated by Restricted Maximum Likelihood; the remainder is the pair main effect (σ_p^2).

	Main effects			Interactions + residual			
Dataset	σ_j^2	σ_f^2	σ_o^2	σ_{pj}^2	σ_{pf}^2	σ_{po}^2	Res.
MT-Bench	0.0	0.0	2.1	10.3	1.1	5.0	4.2
AlpacaFarm	2.1	0.0	1.8	17.5	1.4	3.8	6.0
JudgeBench	0.9	0.1	**6.2**	**34.8**	4.4	6.2	7.9
LLMBar	0.3	0.0	2.6	22.7	1.8	4.2	4.3

Table 6: Four-facet variance decomposition (% of σ^2_{total}) for verdict direction at $T_s = 0$.

Three patterns emerge. First, the order main effect is the only non-pair main effect of non-negligible size (2–6% across datasets, peaking at 6.2% on JudgeBench), confirming that position bias is a structured measurement facet — the variance mapped by φ in our formalization. Second, the pair × judge interaction is the largest within-pair source of disagreement (10–35%); on JudgeBench it actually exceeds the pair main effect itself, indicating that reliability is item-specific rather than judge-level. Third, the pair × order interaction is itself substantial (4–6%): position bias varies across items, exactly the pair-level structure that φ aggregates.

D Prompt Templates

All prompts share a common user message template and differ only in the system prompt and output format instruction. Placeholders `{question}`, `{response_a}`, and `{response_b}` are filled per evaluation pair. When the presentation order is BA, the contents of Response A and Response B are swapped; all other text is identical.

D.1 User Message Template (All Formats)

```
[Question]
{question}

[Response A]
{response_a}

[Response B]
{response_b}

{format-specific instruction appended here}
```

D.2 Pairwise Evaluation (5 Formats)

System Prompt (shared by all 5 pairwise formats).

> You are an expert and impartial judge evaluating the quality of two AI assistant responses to a user question. Your task is to determine which response is better. Focus on helpfulness, accuracy, depth, clarity, and relevance. Avoid any position bias — the order in which responses appear should not influence your judgment.

Format 1: Direct (baseline). Single-token verdict, no rationale. Max tokens: 32.

> Reply with exactly one word: A, B. A means Response A is better. B means Response B is better.

Format 2: JSON Verdict-First. Verdict before rationale ("decide then justify"). Max tokens: 4096.

> Reply with a JSON object only (no markdown fences), in this exact shape:
> `{"winner": "A", "rationale": "your reasoning here"}`
> winner must be A, B.

Format 3: JSON CoT (rationale-first). Rationale before verdict ("think then decide"). Max tokens: 4096.

> Reply with a JSON object only (no markdown fences), in this exact shape:
> `{"rationale": "your reasoning here", "winner": "A"}`
> winner must be A, B.

Format 4: XML Verdict-First. Verdict before rationale in XML. Max tokens: 4096.

> Reply with an XML block only, in this exact shape:
> `<judgment><winner>A</winner><rationale>your reasoning here</rationale></judgment>`
> winner must be A, B.

Format 5: XML CoT (rationale-first). Rationale before verdict in XML. Max tokens: 4096.

> Reply with an XML block only, in this exact shape:
> `<judgment><rationale>your reasoning here</rationale><winner>A</winner></judgment>`
> winner must be A, B.

D.3 GEval Rubric-Based Evaluation

GEval uses a single API call per evaluation with a structured rubric in the system prompt and a JSON output format requiring both scores and a verdict.

Shared Output Format Instruction (prepended to user message).

> IMPORTANT: You must respond with ONLY a JSON object. Do not write any free text, do not solve the problem yourself, do not use markdown fences. Output exactly this shape and nothing else:
> `{"rationale": "your evaluation here", "score_a": 4, "score_b": 2, "winner": "A"}`
> score_a and score_b are integers from 1 (very poor) to 5 (excellent) rating each response against the rubric criteria independently.
> winner must be exactly one of: A, B.
> winner should be A if score_a > score_b, B if score_b > score_a.

GEval System Prompt — JudgeBench.

You are an expert and impartial judge evaluating the quality of two AI assistant responses to a user question. Your task is to determine which response is better based on a structured rubric.

Apply the following evaluation rubric in order. Consider criteria earlier in the list more important when responses are otherwise close.

RUBRIC:

1. **Factual Accuracy** — Does the response contain correct facts, calculations, code, or logical deductions? Errors in reasoning, arithmetic, or factual claims are heavily penalized.

2. **Correctness of Final Answer** — For questions with a definitive correct answer (math, science, code, multiple-choice), does the response arrive at the right answer? A response with the correct final answer but imperfect explanation is preferred over a response with an incorrect answer and polished presentation.

3. **Completeness** — Does the response address all parts of the question without omitting required steps, sub-problems, or output requirements stated in the prompt?

4. **Clarity and Presentation** — Is the response well-organized, clearly explained, and easy to follow? Prefer structured, step-by-step exposition where appropriate.

Scoring guidance: If one response has a factual error or wrong answer and the other does not, the response without the error wins regardless of other criteria. A tie is appropriate only when both responses are correct and equivalent in quality across all criteria, or both contain equivalent errors. Avoid any position bias — the order in which responses appear should not influence your judgment.

GEval System Prompt — LLMBar.

You are an expert and impartial judge evaluating the quality of two AI assistant responses to a user instruction. Your task is to determine which response is better based on a structured rubric.

Apply the following evaluation rubric in order. Consider criteria earlier in the list more important when responses are otherwise close.

RUBRIC:

1. **Instruction Adherence** — Does the response follow every explicit and implicit constraint specified in the instruction? Violations of format, scope, perspective, tone, length, or output type are penalized. A response that is fluent but ignores or modifies the instruction is worse than a plainer response that follows it exactly.

2. **Completeness** — Does the response address all aspects of the instruction without adding unsolicited content or dropping required elements?

3. **Relevance** — Is the response directly focused on the task without introducing irrelevant information or tangential elaborations that were not requested?

4. **Clarity** — Is the response well-written and easy to understand, given that it correctly addresses the instruction?

Scoring guidance: Instruction adherence is the primary criterion. A response that deviates from the instruction in any material way loses to one that follows it, even if the deviating response is more informative or better written. A tie is appropriate only when both responses follow the instruction equally and are equivalent in all other criteria. Avoid any position bias — the order in which responses appear should not influence your judgment.

D.4 FLASK Multi-Criterion Evaluation

FLASK issues 4 separate API calls per evaluation one per criterion and aggregates by majority vote. Each call uses the same JSON output format as GEval (Section D.3) with an additional criterion definition injected into the user message.

FLASK System Prompt — JudgeBench.

You are an expert and impartial judge evaluating the quality of two AI assistant responses to a user question. Your task is to determine which response is better on one specific criterion. Evaluate only the criterion stated — do not factor in other quality dimensions. Avoid any position bias.

FLASK System Prompt — LLMBar.

You are an expert and impartial judge evaluating the quality of two AI assistant responses to a user instruction. Your task is to determine which response is better on one specific criterion. Evaluate only the criterion stated — do not factor in other quality dimensions. Avoid any position bias.

FLASK Criteria — JudgeBench. The following criterion definitions are injected into the user message as `[Criterion: {name}]` blocks. Each is evaluated independently in a separate API call.

1. Factual Accuracy. Which response contains fewer factual errors, incorrect calculations, or flawed reasoning steps? A response with any factual error loses to one that is entirely correct.
2. Completeness. Which response more fully addresses all parts of the question, including all required sub-problems, output formats, or conditions stated in the prompt?
3. Logical Coherence. Which response presents a more logically consistent chain of reasoning, with conclusions that follow from stated premises and no internal contradictions?
4. Instruction Following. Which response more precisely follows the structural and formatting requirements stated in the question (e.g., box the answer, return JSON, produce code in a specific language)?

FLASK Criteria — LLMBar.

1. Instruction Adherence. Which response more faithfully follows every explicit and implicit constraint in the instruction — including required format, scope, perspective, tone, length, and output type — without omitting or modifying any stated requirement?
2. Relevance. Which response stays more tightly focused on the requested task, without adding unsolicited content, tangential elaborations, or off-topic information?
3. Completeness. Which response more fully addresses all elements of the instruction without omitting required parts or prematurely terminating the response?
4. Clarity. Which response is better written, more clearly organized, and easier to understand, assuming it has otherwise satisfied the instruction?

D.5 Prompt Design Summary

Framework	Format	Rubric	Scores	Cost	Output
Pairwise	Direct	No	No	1	A / B
Pairwise	JSON VF	No	No	1	JSON
Pairwise	JSON CoT	No	No	1	JSON
Pairwise	XML VF	No	No	1	XML
Pairwise	XML CoT	No	No	1	XML
GEval	JSON CoT	Yes	1–5	1	JSON
FLASK	JSON CoT	Yes	1–5	4	JSON $\times$ 4

Table 7: Prompt design matrix. All frameworks share the same user message template (Section D.1). Cost is measured in API calls per (pair, judge, order) evaluation.

E API Cost

The total API cost for conducting the cross-cutting reliability audit was approximately $2,100 USD. A detailed breakdown of costs across the three evaluation paradigms is provided below (based on public API cost of model providers):

Evaluation Paradigm	Cost (USD)
Pairwise Experiments	$1,450
FLASK Rubric Experiments	$390
G-Eval Experiments	$260
Total	**$2,100**

Table 8: Breakdown of API costs across experimental paradigms.

F Dataset Details

Table 9 summarizes the four benchmarks used in our study. All pairs were fixed at the start of the experiment and shared across all judges, formats, presentation orders, temperatures, and repetitions.

Dataset	Truth	Pairs	Domains	Sampling
MT-Bench	Subjective	80	1	Random (seed=42)
AlpacaFarm	Subjective	100	1	Random (seed=42)
JudgeBench	Objective	100	17	Stratified by domain
LLMBar	Objective	100	5	Stratified by split

Table 9: Dataset summary. Calls are computed at T_s=0 as pairs $\times$ 6 judges $\times$ 5 formats $\times$ 2 orders $\times$ 10 reps.

MT-Bench (Zheng et al., 2023) Pairwise comparisons from the human judgment set. Labels are crowd-sourced preferences.

AlpacaFarm (Dubois et al., 2023) 100 instruction-following pairs with binary preference labels. The balanced label distribution makes the dataset sensitive to position bias.

JudgeBench (Tan et al., 2025) 100 pairs stratified from 620 examples across 17 domains with objective ground truth. One response in each pair is verifiably correct.

LLMBar (Zeng et al., 2024) 100 pairs sampled evenly across 5 robustness splits. Most pairs contain adversarial distractors designed to fool LLM judges.

G Assumptions underlying T

The elegant generalized expression for the trustworthy verdict rate, $T = C \cdot (A - \varphi q)$, relies on Assumption 2 (A2): that is N, P, and R are mutually independent. In practice, this assumption may not hold exactly. Here we empirically demonstrate that even when A2 is violated, T continues to behave as a conservative lower bound.

To test this, we define $T_{\text{joint}} = \Pr(N = 1 \wedge P = 1 \wedge R = 1)$, the assumption-free joint probability, and compare it against the closed-form estimate $T_{\text{closed}} = C \cdot (A - \varphi/2)$.

Table 10 reports the gap $T_{\text{closed}} - T_{\text{joint}}$ across benchmarks and judges. The gap is consistently small (typically under 5 pp.), indicating that the closed-form expression remains an empirically bounded, conservative approximation even without strict independence.

Benchmark	**Judge**	$T_{\textbf{closed}}$	$T_{\textbf{joint}}$	**Gap**	**Gap 95% CI**
MT-Bench	GPT-5.2	0.54	0.58	-0.04	[-0.068, -0.012]
	GPT-4o	0.55	0.57	-0.02	[-0.041, -0.007]
	Gemini-2.5-Pro	0.50	0.55	-0.05	[-0.072, -0.024]
	Gemini-2.0-Flash	0.51	0.52	-0.01	[-0.013, -0.001]
	Claude-Sonnet-4.6	0.62	0.64	-0.02	[-0.043, -0.003]
	Llama-3.3-70B	0.58	0.58	0.00	[-0.006, +0.000]
JudgeBench	GPT-5.2	0.49	0.55	-0.05	[-0.082, -0.023]
	GPT-4o	0.31	0.32	0.00	[-0.012, +0.023]
	Gemini-2.5-Pro	0.62	0.65	-0.03	[-0.063, +0.000]
	Gemini-2.0-Flash	0.23	0.23	0.00	[-0.008, +0.008]
	Claude-Sonnet-4.6	0.64	0.67	-0.03	[-0.058, -0.010]
	Llama-3.3-70B	0.32	0.32	0.00	[-0.001, +0.005]
LLMBar	GPT-5.2	0.67	0.70	-0.04	[-0.062, -0.015]
	GPT-4o	0.64	0.66	-0.02	[-0.032, -0.002]
	Gemini-2.5-Pro	0.65	0.68	-0.03	[-0.053, -0.008]
	Gemini-2.0-Flash	0.43	0.43	0.00	[-0.012, +0.008]
	Claude-Sonnet-4.6	0.76	0.77	-0.01	[-0.021, +0.001]
	Llama-3.3-70B	0.58	0.58	0.00	[-0.005, +0.000]
AlpacaFarm	GPT-5.2	0.40	0.43	-0.03	[-0.058, +0.002]
	GPT-4o	0.48	0.50	-0.02	[-0.032, +0.001]
	Gemini-2.5-Pro	0.39	0.41	-0.02	[-0.045, +0.005]
	Gemini-2.0-Flash	0.43	0.43	0.00	[-0.007, +0.008]
	Claude-Sonnet-4.6	0.51	0.53	-0.02	[-0.033, +0.000]
	Llama-3.3-70B	0.52	0.52	0.00	[-0.005, +0.000]

Table 10: Comparison of T_{closed} and T_{Joint} across benchmarks and judges