Judging the Judges: A Systematic Evaluation of Bias Mitigation Strategies in LLM-as-a-Judge Pipelines

Anonymous authors
Paper under double-blind review

Abstract

LLM-as-a-Judge has become the dominant paradigm for evaluating language model outputs, yet LLM judges exhibit systematic biases that compromise evaluation reliability. We present a comprehensive empirical study comparing nine debiasing strategies across five judge models from four provider families (Google, Anthropic, OpenAI, Meta), three benchmarks (MT-Bench n=400, LLMBar n=200, custom n=225), and four bias types. Our key findings: (1) *Style bias* is the dominant bias (0.76–0.92 across all models), far exceeding position bias (≤ 0.04), yet has received minimal research attention. (2) All models show a *conciseness preference* on expansion pairs, but truncation controls confirm they correctly distinguish quality from length (0.92–1.00 accuracy), suggesting quality-sensitive evaluation rather than a simple length bias. (3) *Debiasing is beneficial but model-dependent*: the combined budget strategy significantly improves Claude Sonnet 4 by +11.2 pp ($p < 0.0001$), with directionally positive trends for other models. Only 2 of 20 non-baseline configurations show decreased agreement. We release our evaluation framework, controlled dataset, and all experimental artifacts.

1 Introduction

As large language models (LLMs) have grown in capability, evaluating their outputs has become a critical bottleneck. Human evaluation remains the gold standard but is expensive, slow, and difficult to reproduce (Zheng et al., 2024; Li et al., 2024b). This has driven widespread adoption of LLM-as-a-Judge pipelines, where one LLM evaluates the outputs of others. Major benchmarks including MT-Bench (Zheng et al., 2024), AlpacaEval (Li et al., 2024c), and Chatbot Arena (Chiang et al., 2024) now rely on LLM judges for scalable evaluation.

However, LLM judges are not neutral arbiters. Prior work has documented systematic biases: *position bias* (Wang et al., 2024a; Zheng et al., 2024), *verbosity bias* (Saito et al., 2023; Wu & Aji, 2024), *self-preference bias* (Panickssery et al., 2024; Liu et al., 2024), and *style bias* (Wu & Aji, 2024). Several mitigations have been proposed in isolation: position swapping (Zheng et al., 2024), multi-judge ensembles (Verga et al., 2024), calibrated rubrics (Kim et al., 2024), and chain-of-thought prompting (Wei et al., 2022; Shankar et al., 2024). Yet no prior work has systematically compared these strategies, measured their cross-bias interactions, or evaluated them in a unified framework.

We address this gap with the following contributions:

1. **Unified benchmark.** We compare 9 debiasing strategies across 5 judges from 4 model families, 3 benchmarks ($n = 400$ on MT-Bench), and 4 bias types.
2. **Controlled bias measurement.** We introduce a 225-pair dataset with known ground truth. Style bias (0.76–0.92) dominates all other biases. All models show conciseness preferences on expansion pairs, though truncation controls confirm quality-sensitive evaluation.
3. **Model-dependent debiasing.** Debiasing is beneficial but the optimal strategy varies by model, with statistically significant gains of +7.2 to +11.2 pp for select model-strategy pairs (after multiple testing correction) and directionally positive trends across others.

4. **Statistically grounded recommendations.** We report bootstrap 95% CIs and McNemar's tests, providing model-specific recommendations with effect sizes and significance levels.

Our findings are directly relevant to three communities: (1) researchers developing LLM evaluation pipelines, who need guidance on which debiasing strategies to apply and when; (2) model developers using LLM judges for RLHF reward modeling or benchmark comparison, who need to understand how formatting differences may confound their model rankings; and (3) benchmark maintainers (AlpacaEval, Chatbot Arena, MT-Bench), who need evidence-based guidance on whether to normalize response formatting before judging.

Our work also contributes to reproducibility in LLM evaluation. We reproduce foundational bias claims (Wang et al., 2024a; Saito et al., 2023; Panickssery et al., 2024) on current-generation models, finding that some no longer hold (e.g., position bias is now negligible across all tested models). We release a controlled dataset that enables others to monitor these biases as models evolve, and we cache all individual evaluation results, enabling downstream researchers to reanalyze our data without re-running expensive API calls.

2 Related Work

LLM-as-a-Judge. The use of LLMs as evaluators was popularized by MT-Bench (Zheng et al., 2024), which demonstrated that GPT-4 judgments correlate with human preferences at over 80% agreement. AlpacaEval (Li et al., 2024c) extended this to instruction-following evaluation, while Arena-Hard (Li et al., 2024b) introduced automated difficulty filtering to create challenging evaluation sets from crowd-sourced data. G-Eval (Liu et al., 2023) proposed using GPT-4 with chain-of-thought and form-filling paradigms for NLG evaluation, establishing prompting patterns now standard in LLM evaluation pipelines. FLASK (Ye et al., 2024) introduced fine-grained evaluation across twelve alignment skill sets, enabling more nuanced assessment than binary or scalar judgments. JudgeBench (Tan et al., 2024) specifically evaluates LLM judges' ability to make correct judgments on challenging instances where ground truth is unambiguous. Gu et al. (2024) survey over 100 papers in this rapidly growing area. RewardBench (Lambert et al., 2024) provides a systematic evaluation framework for reward models that share key properties with LLM judges.

Fine-tuned and specialized judges. An alternative to prompting general-purpose LLMs as judges is training specialized evaluation models. Prometheus 2 (Kim et al., 2024) introduces an open-source evaluator trained on evaluation-specific data, achieving competitive performance with proprietary models on rubric-based assessment. PandaLM (Wang et al., 2024b) trains a dedicated judge model for instruction-tuning optimization. Auto-J (Li et al., 2024a) proposes a generative judge that produces detailed critiques alongside its verdicts. CritiqueLLM (Ke et al., 2024) scales critique-based evaluation by training models to produce structured explanations. TIGERScore (Jiang et al., 2024) builds explainable metrics across diverse text generation tasks. While fine-tuned approaches can reduce some biases through targeted training, they introduce concerns about training data coverage and domain generalization. Our work complements this line by studying bias mitigation through prompting and aggregation strategies applicable to any general-purpose LLM judge without requiring additional training.

Known biases. Position bias was documented by Wang et al. (2024a) and confirmed across models (Zheng et al., 2024; Wu & Aji, 2024). Li et al. (2024d) propose split-and-merge calibration to address position effects. Verbosity bias has been observed in both pairwise and pointwise settings (Saito et al., 2023; Wu & Aji, 2024); Dubois et al. (2024) introduce length-controlled win rates as a post-hoc correction. Self-preference was measured by Panickssery et al. (2024) and further characterized by Liu et al. (2024). Xu et al. (2024) highlight that format-level biases remain underexplored, a gap our style bias measurement addresses directly. See also Stureborg et al. (2024); Chen et al. (2024); Huang et al. (2024) for related bias analyses across evaluation contexts.

Mitigation strategies. Position swapping is the most common mitigation (Zheng et al., 2024). Verga et al. (2024) propose multi-model panels and show that diverse judge ensembles improve reliability. Shankar et al. (2024) demonstrate that chain-of-thought prompting improves evaluation quality by encouraging more deliberate reasoning. Park et al. (2024) show that debiased training data can reduce bias in fine-

tuned evaluators. Zhang et al. (2024) find that wider and deeper model panels produce fairer evaluations, suggesting architectural diversity complements prompting-based mitigation. However, these strategies have been evaluated in isolation, making direct comparison impossible. Most closely related, CALM (Koo et al., 2024) benchmarks cognitive biases in LLM evaluators but does not compare mitigation strategies, studies biases in isolation rather than measuring cross-bias interactions, and tests only two models from one provider family. We extend this work in all three dimensions: we compare nine strategies, measure cross-bias effects, and evaluate five models from four independent provider families.

3 Methodology

3.1 Experimental Framework

We design a factorial experiment crossing four dimensions: judge models, debiasing strategies, benchmarks, and bias types. All experiments use pairwise comparison, where the judge receives a question and two candidate responses and must output a structured JSON verdict (A, B, or tie) with reasoning.

3.2 Judge Models

We evaluate five models from four provider families, spanning three capability tiers:

- **Gemini 2.5 Pro** (Gemini Team, 2024) (frontier, Google): Accessed via Vertex AI.
- **Claude Sonnet 4** (frontier, Anthropic): Accessed via Vertex AI.
- **GPT-4o** (OpenAI, 2024) (frontier, OpenAI): Accessed via OpenAI API.
- **Gemini 2.5 Flash** (Gemini Team, 2024) (mid-tier, Google): Optimized for speed and cost.
- **Llama 3.3-70B** (Grattafiori et al., 2024) (open-source, Meta): Accessed via Vertex AI Model Garden (MaaS).

Using models from four families enables measurement of self-preference bias and strengthens generalizability. All models use temperature 0.1 to minimize stochastic variation while avoiding the degenerate outputs occasionally observed at temperature 0; Gemini models use structured JSON output, while others parse JSON from free-text responses.

3.3 Debiasing Strategies

We implement nine strategies spanning three mechanism categories:

Single-call prompting (1× cost).

B0: Baseline (Naive) Minimal prompt requesting a JSON verdict with reasoning.

S4: Calibrated Rubric Structured 5-criteria rubric (accuracy, relevance, completeness, clarity, reasoning depth), each scored 1–10, verdict derived from total scores.

S5: Chain-of-Thought (CoT) Explicit step-by-step analysis required before verdict.

Multi-call aggregation (2–3× cost).

S1: Position Swap Two calls with A/B order reversed; consensus verdict or tie on disagreement. Cost: 2×.

S2: Same-Family Ensemble Three calls at temperatures {0.0, 0.3, 0.7}, majority vote. Cost: 3×.

Combined approaches (2× cost).

S8: Combined Budget Position swap + CoT + rubric: two swapped calls using a merged CoT-rubric prompt. Cost: 2×.

We also implemented S3 (cross-family ensemble), S6 (reference-guided), and S7 (combined full); results are available in our released artifacts.

3.4 Benchmarks

MT-Bench (Zheng et al., 2024) provides 3,355 pairwise comparisons with human preference labels across 8 categories. We sample the same 400 instances for all configurations (enabling paired comparisons), yielding bootstrap 95% CIs of approximately ±0.05.

LLMBar (Zeng et al., 2024) contains 419 adversarial instances where one response is clearly better but the other contains superficially appealing distractors. We use 200 instances.

Custom Controlled Dataset. Our novel contribution: 225 synthetic pairs for isolating individual bias types. The first 200 pairs span four categories (50 each) where an unbiased judge should say "tie": **LENGTH** (expansion, ∼2.8× longer), **POSITION** (identical responses), **STYLE** (markdown vs. plain prose), and **MODEL_ORIGIN** (Gemini Pro vs. Claude answers). An additional 25 **LENGTH (truncation)** pairs test whether judges correctly prefer genuinely complete answers over mechanical truncations. Details in Appendix A.

3.5 Metrics

All metrics include bootstrap 95% confidence intervals ($n_{\text{boot}} = 2000$). We report human agreement rate, Cohen's κ (chance-corrected), and bias scores ($P(\text{prefers A}) - P(\text{prefers B})$). Strategy comparisons use McNemar's test with continuity correction.

4 Results

We make four main claims, each directly supported by specific experimental evidence:

1. *Style bias dominates all other biases*: evidenced by Figure 1 (bias scores 0.76–0.92 across all five models) and the controlled STYLE pairs in Appendix A.
2. *Verbosity operates as a conciseness preference, not a simple length bias*: evidenced by Section 4.1.1 (negative bias on expansion pairs across all models) combined with Appendix F (0.92–1.00 accuracy on truncation pairs where longer responses are genuinely more complete).
3. *Debiasing is statistically beneficial for specific model-strategy pairs*: evidenced by Table 1 (two McNemar results significant after Holm-Bonferroni correction) and the aggregate sign test (Appendix D).
4. *Optimal debiasing strategy is model-dependent*: evidenced by Table 1 (different best strategies per model) and the cross-bias analysis in Section 4.3.

4.1 Baseline Bias Landscape

Figure 1 presents baseline bias magnitudes across all five judge models.

Style bias is the dominant bias. All five models show strong style bias (0.76–0.92), overwhelmingly preferring markdown-formatted responses over plain prose even when content is identical. This bias substantially exceeds both position bias (≤ 0.04) and verbosity bias magnitudes (0.20–0.76) for most models, yet has received minimal research attention. The consistency across all four independent model families confirms this is not a provider-specific artifact.

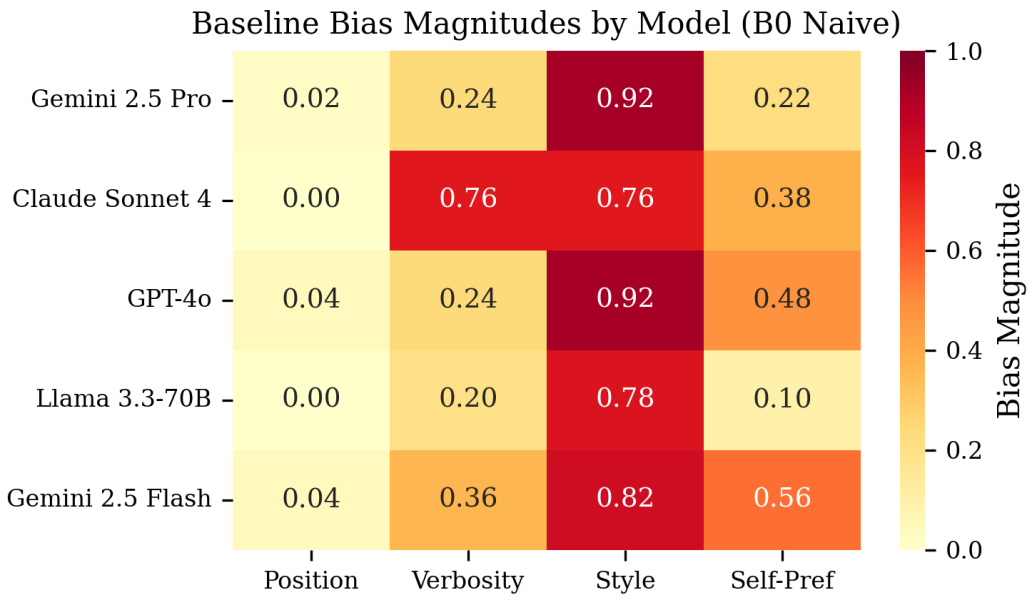

Figure 1: Baseline bias magnitudes by model (B0). Style bias dominates (0.76–0.92). Position bias is negligible (≤ 0.04). Verbosity values reflect preference for *shorter* responses.

Position bias is negligible. Contrary to emphasis in prior work (Wang et al., 2024a; Zheng et al., 2024), position bias is ≤ 0.04 across all five models, likely reflecting improvements in instruction tuning since earlier studies.

4.1.1 Conciseness preference on controlled pairs

All five models prefer *shorter* responses on our controlled LENGTH expansion pairs, suggesting current models may not exhibit the traditional "verbosity bias" toward longer outputs (Saito et al., 2023): Claude Sonnet 4 (-0.76), Flash (-0.36), GPT-4o (-0.24), Pro (-0.24), Llama (-0.20). However, on truncation pairs where the longer response is genuinely more complete, all models correctly prefer the complete version (accuracy 0.92–1.00). This asymmetry indicates quality-sensitive evaluation: judges penalize filler content but recognize genuine completeness. The consistency across all four model families suggests shared training dynamics. We discuss implications in Section 5.

Self-preference varies across models. In MODEL_ORIGIN pairs (A=Gemini, B=Claude), self-preference is not uniformly tied to model family: Flash shows strong self-preference ($+0.56$), while Pro shows the opposite (-0.22). GPT-4o strongly prefers Claude's responses (-0.48), and Llama shows near-zero self-preference (-0.10).

4.2 Strategy Effectiveness

Table 1 presents human agreement rates on MT-Bench ($n = 400$) with bootstrap 95% CIs.

Debiasing effects are model-dependent. Two model-strategy pairs show statistically significant improvements after Holm-Bonferroni correction: Claude S8 ($+11.2$ pp, $p < 0.0001$) and Claude S5 ($+7.2$ pp, $p = 0.004$). Gemini Pro S1 ($+4.6$ pp, $p = 0.012$ unadjusted) is marginally significant after correction. For GPT-4o, Llama, and Gemini Flash, improvements are directionally positive ($+0.1$ to $+3.8$ pp) but do not reach statistical significance at $n = 400$. We cannot claim debiasing helps these models without qualification; we can only claim the aggregate pattern (sign test, $p < 0.001$, Appendix D) is consistent with a positive effect. Larger sample sizes would be needed to confirm model-specific effects. A sign test across all 20 non-baseline

Table 1: Human agreement rates on MT-Bench ($n = 400$) with 95% bootstrap CIs ($n_{\text{boot}} = 2000$). Best per model in **bold**. Teal indicates > 5 pp improvement. *Significant at $p < 0.05$, McNemar's test; †marginal after Holm-Bonferroni correction (Appendix D).

Model	B0 Baseline	S1 Pos. Swap	S4 Rubric	S5 CoT	S8 Combined
Gemini 2.5 Pro	.652	**.698**†	.677	.650	.675
	[.60,.70]	[.65,.74]	[.63,.72]	[.60,.70]	[.63,.72]
Claude Sonnet 4	.588	.590	.605	.660*	**.700***
	[.54,.64]	[.54,.64]	[.56,.66]	[.61,.70]	[.66,.74]
GPT-4o	.634	.610	.635	.647	**.662**
	[.59,.68]	[.56,.66]	[.59,.68]	[.60,.69]	[.62,.71]
Llama 3.3-70B	.647	.670	.675	.665	**.685**
	[.60,.69]	[.62,.71]	[.63,.72]	[.62,.71]	[.64,.73]
Gemini 2.5 Flash	.640	**.665**	.647	**.665**	.652
	[.59,.69]	[.62,.71]	[.60,.69]	[.62,.71]	[.60,.70]

S2 and S3 (ensemble strategies) omitted; results in released artifacts.

configurations confirms the overall positive direction: 18 of 20 show improvement ($p < 0.001$; Appendix D), indicating debiasing has a systematically positive effect even where individual improvements are modest. The two negative results are GPT-4o with position swap (−2.4 pp) and Gemini Pro with CoT (−0.2 pp, within CI overlap).

Position swap shows model-dependent effects. Position swap (S1) provides the largest single-strategy gain for Gemini Pro (+4.6 pp, κ: 0.442 → 0.528) but is the only strategy that hurts GPT-4o (−2.4 pp). The key distinction is between position *bias* (negligible on controlled pairs, ≤ 0.04) and position *sensitivity*: models that more frequently reverse verdicts when order changes benefit from tie resolution. On LLMBar's adversarial instances, position swap consistently hurts all models (−2.5 to −7.0 pp), as tie resolution discards correct verdicts on clear-cut cases.

CoT is the safest single strategy. CoT (S5) improves or is neutral for all models on MT-Bench and improves *all* models on LLMBar (+1.5 to +13.0 pp; point estimates at $n = 200$), making it the only universally beneficial strategy across both benchmarks. S4 (rubric) is neutral-to-positive for all models (+0.1 to +2.8 pp). Claude with S8 achieves the highest agreement of any configuration (0.700, $\kappa = 0.530$).

4.3 Cross-Bias Interactions

Figure 2 reveals how strategies designed for one bias type affect others.

CoT provides the largest style bias reduction (−0.14 avg.) while rubric provides the largest verbosity bias reduction (−0.11 avg.). This complementarity explains S8's effectiveness. Position swap reduces style bias (−0.12) but *increases* verbosity bias (+0.07 avg.), a cross-bias interaction that single-strategy evaluations would miss. S8 achieves the largest overall style bias reduction (−0.26, from 0.84 to 0.58 avg.).

4.4 Generalization to LLMBar

CoT (S5) is best for *all* models on LLMBar, with Claude achieving 86.5% (+13.0 pp over baseline). Position swap hurts all models (−2.5 to −7.0 pp), confirming task-dependence: position swap suits natural evaluation data (MT-Bench) but harms performance on curated benchmarks with unambiguous quality differences.

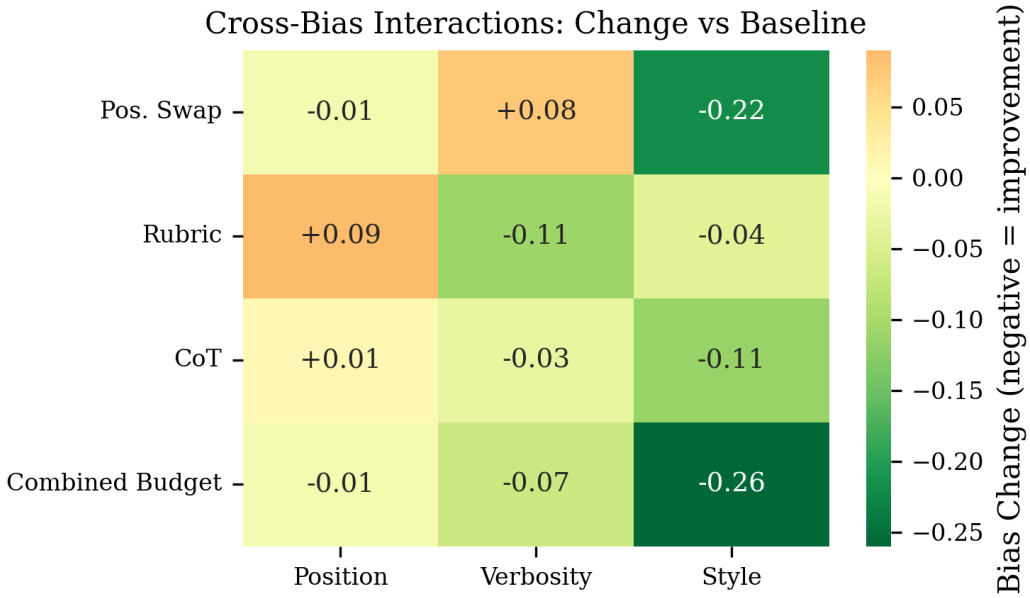

Figure 2: Cross-bias interactions: change in bias magnitude vs. baseline, averaged across Pro, Claude, GPT-4o, and Llama (Flash excluded as mid-tier). Negative (green) = improvement.

Table 2: Human agreement on LLMBar (199–200 adversarial instances). Best per model in **bold**.

Model	B0	S1	S4	S5	S8
Gemini 2.5 Pro	.820	.780	.830	**.835**	.780
Claude Sonnet 4	.735	.700	.745	**.865**	.800
GPT-4o	.814	.789	.854	**.864**	.759
Llama 3.3-70B	.655	.585	.595	**.700**	.535
Gemini 2.5 Flash	.840	.795	.825	**.855**	.815

5 Discussion

5.1 Style Bias: The Dominant but Underappreciated Bias

Our most robust finding is style bias severity (0.76–0.92 across all five models). In benchmarks like AlpacaEval and Chatbot Arena, where models produce outputs with distinct formatting conventions, formatting differences could substantially influence win-rate differences when content quality is comparable. S8 achieves meaningful style bias reduction (avg. $0.84 \rightarrow 0.58$). We recommend that evaluation frameworks either normalize formatting or explicitly instruct judges to disregard it.

5.2 Nuancing the Verbosity Narrative

All five models prefer concise responses over padded alternatives, which *complicates* the established "verbosity bias" narrative (Saito et al., 2023; Wu & Aji, 2024). Rather than a uniform directional preference, the picture is more nuanced: judges penalize filler content (expansion pairs: -0.20 to -0.76) but correctly reward completeness (truncation accuracy: 0.92–1.00). This suggests quality-sensitive evaluation rather than a uniform length bias. The strong directional consistency across all four model families still suggests a systematic preference worth monitoring. Regardless of direction, the magnitude confirms that response length significantly influences judge verdicts. Detailed per-strategy analysis is in Appendix F.

5.3 Matching Strategy to Context

On MT-Bench, the statistically confirmed improvements (Claude S8: $p < 0.0001$; Claude S5: $p = 0.004$) and the marginally significant Pro S1 ($p = 0.012$ unadjusted) demonstrate that debiasing can meaningfully improve evaluation reliability. For GPT-4o, Llama, and Gemini Flash, improvements are directionally positive (+0.1 to +3.8 pp) but do not reach statistical significance at $n = 400$. We cannot claim debiasing helps these models without qualification; we can only claim the aggregate pattern (sign test, $p < 0.001$, Appendix D) is consistent with a positive effect. Larger sample sizes would be needed to confirm model-specific effects. On LLMBar, CoT helps all models (+1.5 to +13.0 pp), making it the safest default for unknown evaluation contexts.

5.4 Practical Application Guide

Practitioners building LLM evaluation pipelines can use our findings as follows:

- **Natural evaluation data (MT-Bench-style):** Consult Algorithm 1 for model-specific strategy recommendations. For models not covered, default to CoT forcing (S5), which is universally positive in our tests.
- **Adversarial or high-stakes evaluation (LLMBar-style):** Always use CoT forcing (S5). Avoid position swap, which consistently hurts performance on curated benchmarks (−2.5 to −7.0 pp).
- **Model comparison via LLM judging (AlpacaEval, Arena-style):** Normalize response formatting before judging. Our style bias results (0.76–0.92) indicate that unnormalized formatting comparisons may produce win-rate differences driven by formatting alone.
- **Budget-constrained pipelines:** Llama 3.3-70B via Vertex AI Model Garden provides zero-API-cost judging with competitive accuracy (68.5% with S8).
- **Best overall accuracy:** Claude Sonnet 4 with S8 (70.0%, $\kappa = 0.530$, $p < 0.0001$) or Gemini 2.5 Pro with S1 (69.8%, $\kappa = 0.528$, $p = 0.012$ unadjusted).
- **Default for unknown contexts:** CoT (S5) at 1× cost, the only universally positive strategy across both benchmarks.

We formalize these recommendations as Algorithm 1, which selects a debiasing strategy based on model family, task type, and budget constraints. Cost-accuracy Pareto analysis is provided in Appendix G.

5.5 Limitations

(1) Expansion-based LENGTH pairs may confound length with filler quality; truncation pairs partially address this. (2) STYLE pairs' markdown preference may partially reflect legitimate readability advantages rather than pure bias. (3) Additional open-weight families (Mistral, Qwen) may exhibit different bias profiles. (4) GPT-4o token counts are estimated (1 token per 4 characters) as usage metadata was not consistently available. (5) Non-significant improvements for GPT-4o, Llama, and Flash may reach significance with larger sample sizes; our $n = 400$ provides bootstrap 95% CIs of approximately ±0.05, which may be insufficient to detect smaller but practically meaningful effects.

6 Conclusion

We present a systematic comparison of debiasing strategies for LLM-as-a-Judge spanning five models from four provider families, evaluated at $n = 400$ on MT-Bench with bootstrap CIs. Style bias (0.76–0.92) is the dominant but underappreciated bias, far exceeding position bias (≤ 0.04). On verbosity, all models prefer concise responses over padded alternatives, though truncation controls confirm judges correctly distinguish quality from length. Debiasing yields statistically significant improvements for Claude (S8: +11.2 pp,

Algorithm 1 JudgeStrategySelector: Recommended judge configuration.

Require: Model family $\mathcal{M}$, task type $\mathcal{T}$, budget multiplier k
Ensure: Strategy $\mathcal{S}$

```
if T = adversarial / high-stakes then
    S ← S5 (CoT Forcing) {Helps all models on adversarial data}
else if M = Claude then
    if k ≥ 2 then
        S ← S8 (Combined Budget) {+11.2 pp, p < 0.0001}
    else
        S ← S5 (CoT Forcing) {+7.2 pp, p = 0.004}
    end if
else if M = Llama (open-source) then
    if k ≥ 2 then
        S ← S8 (Combined Budget) {+3.8 pp, zero API cost}
    else
        S ← S4 (Calibrated Rubric) {+2.8 pp at 1× cost}
    end if
else if M = Gemini Pro then
    S ← S1 (Pos. Swap) {+4.6 pp, p = 0.012}
else if M = Gemini Flash then
    S ← S5 (CoT Forcing) {+2.5 pp at 1× cost}
else if M = GPT-4o then
    if k ≥ 2 then
        S ← S8 (Combined Budget) {+2.8 pp}
    else
        S ← S5 (CoT Forcing) {+1.3 pp at 1× cost}
    end if
else
    S ← S5 (CoT Forcing) {Safest default for unknown models}
end if
return S
```

$p < 0.0001$; S5: +7.2 pp, $p = 0.004$) and a marginally significant gain for Gemini Pro (S1: +4.6 pp, $p = 0.012$ unadjusted), with directionally positive trends for other models (GPT-4o S8: +2.8 pp, Llama S8: +3.8 pp, Flash CoT: +2.5 pp). CoT forcing helps all models on adversarial data (LLMBar: +1.5 to +13.0 pp). Only 2 of 20 non-baseline configurations show decreased agreement. We release our evaluation framework, controlled dataset, and all experimental artifacts.[1]

Broader Impact Statement

LLM judges exhibit systematic biases that can distort model rankings. Style bias (0.76–0.92) could disadvantage models trained on corpora with less markdown-heavy text. The finding that models now show conciseness preferences rather than the expected verbosity bias highlights the dynamic nature of these biases: mitigations calibrated to past behavior may become counterproductive as training evolves. We release our framework and dataset to enable ongoing monitoring of these biases as the LLM evaluation ecosystem continues to develop.

References

Guiming Hardy Chen, Shunian Chen, Zichen Zhang, Junying Liu, and Benyou Wang. Humans or LLMs as the judge? a study on judgement biases. *arXiv preprint arXiv:2402.10669*, 2024.

[1] Code, data, and cached results will be made available upon acceptance.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating LLMs by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Gemini Team. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jiawei Gu, Xuhui Liang, Zhichao Shi, Pawan Kumar Bhatia, and Lianmin Zheng. A survey on LLM-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

Isabel O Huang, Erica R Osborn, and Albert G Gutierrez. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1–40, 2024.

Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. TIGERScore: Towards building explainable metric for all text generation tasks. *arXiv preprint arXiv:2310.00752*, 2024.

Pei Ke, Bosi Wen, Zhuoer Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, et al. CritiqueLLM: Scaling LLM-as-critic for effective and explainable evaluation of large language model generation. *arXiv preprint arXiv:2311.18702*, 2024.

Seungone Kim, Juyoung Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*, 2024.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. Benchmarking cognitive biases in large language models as evaluators. *arXiv preprint arXiv:2309.17012*, 2024.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A Smith, and Hannaneh Hajishirzi. RewardBench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*, 2024a.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*, 2024b.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. AlpacaEval: An automatic evaluator for instruction-following language models. `https://github.com/tatsu-lab/alpaca_eval`, 2024c.

Ziqi Li, Yixin Wang, Iain Murray, and Jacob Eisenstein. Split and merge: Aligning position biases in LLM-based evaluators. *arXiv preprint arXiv:2310.01432*, 2024d.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. Aligning with human judgement: The role of pairwise preference in large language model evaluators. *arXiv preprint arXiv:2403.16950*, 2024.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*, 2024.

Junsoo Park, Seungyeon Jwa, Minjoon Ren, Daeho Kim, and Eunsol Choi. OffsetBias: Leveraging debiased data for tuning evaluators. *arXiv preprint arXiv:2407.06551*, 2024.

Keita Saito, Akifumi Watanabe, Jenia Jitsev, and Yutaka Uchida. Verbosity bias in preference labeling by large language models. *arXiv preprint arXiv:2310.10076*, 2023.

Shreya Shankar, J.D. Zamfirescu-Pereira, Björn Hartmann, Aditya G. Parameswaran, and Ian Arawjo. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. *arXiv preprint arXiv:2404.12272*, 2024.

Rickard Stureborg, Dimitris Alikaniotis, and Jay Deyoung. Large language models are inconsistent and biased evaluators. *arXiv preprint arXiv:2405.01724*, 2024.

Sijun Tan, Sanjay Nair, Aishwarya Nair, Rohan Kishore, Sida Wu, Shailesh Patil, and Sungdong Kim. JudgeBench: A benchmark for evaluating LLM-based judges. *arXiv preprint arXiv:2410.12784*, 2024.

Pat Verga, Sebastian Hofstätter, Sophia Althammer, Yixuan Su, Iryna Gurevych, and Hannaneh Hajishirzi. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*, 2024.

Peiyi Wang, Lei Li, Liang Chen, Dawei Cai, Feng Niu, Zhifang Fan, Zhifang Sui, Bryan Hooi, and Ji-Rong Wen. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*, 2024a.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Hao, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. PandaLM: An automatic evaluation benchmark for LLM instruction tuning optimization. *arXiv preprint arXiv:2306.05087*, 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Minghao Wu and Alham Fikri Aji. Style over substance: Evaluation biases for large language models. *arXiv preprint arXiv:2307.03025*, 2024.

Canyu Xu, Zhenglong Xie, Jiaxin Li, and Chao Xiao. On the perils of LLM-as-a-judge: An investigation of evaluation biases. *arXiv preprint arXiv:2407.05318*, 2024.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. FLASK: Fine-grained language model evaluation based on alignment skill sets. *arXiv preprint arXiv:2307.10928*, 2024.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. LLMBar: A benchmark for evaluating LLM evaluators on pairwise comparisons. *arXiv preprint arXiv:2310.07641*, 2024.

Xinghua Zhang, Bowen Yu, Haiyang Yu, Yangyu Lv, Tingwen Liu, Fei Huang, Hongbo Xu, and Yongbin Li. Wider and deeper LLM networks are fairer LLM evaluators. *arXiv preprint arXiv:2308.01862*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, et al. Judging LLM-as-a-judge with MT-Bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

A Custom Controlled Dataset Details

Our controlled dataset consists of 225 pairs. The first 200 use 50 diverse questions spanning five domains: mathematics (10), coding (10), creative writing (10), factual QA (10), and instruction following (10). For each question, we generate four pair types (expected verdict: tie):

- **LENGTH (expansion)**: A base response (avg. 64 words) and an expanded version (avg. 178 words, 2.8× ratio). Response A is expanded in 34/50 cases; in 16/50 cases the generation model's expansion was shorter than the base.
- **POSITION**: Identical responses in both slots.
- **STYLE**: Same content in markdown (A) and plain prose (B).
- **MODEL_ORIGIN**: Gemini 2.5 Pro (A) and Claude Sonnet 4 (B) independently answer with matched depth/style instructions.

An additional 25 **LENGTH (truncation)** pairs address the verbosity confound: a comprehensive, high-quality answer (avg. 485 words) is mechanically truncated to ∼40% of its length (avg. 182 words) by cutting at sentence boundaries. The long response is genuinely more complete (expected verdict: A). If judges prefer the shorter version, this confirms conciseness bias independent of filler quality.

LENGTH, POSITION, and STYLE pairs were generated by Gemini 2.5 Flash to avoid model-family confounds. Limitations of the controlled dataset are discussed in Section 5.

Usage instructions. The dataset is structured for plug-and-play bias measurement. Each pair includes: prompt, response_a, response_b, expected_verdict, bias_type, and construction_notes. Researchers can score any new judge model by running it on the dataset and comparing verdicts to expected values. A reference scoring script is provided in the accompanying repository.

Intended uses and limitations. This dataset is designed for screening LLM judge models for basic formatting, position, and length biases. It provides a fast, controlled signal for whether a given judge exhibits known bias patterns. It is *not* intended as a comprehensive benchmark for ranking judge quality across all dimensions, nor as a substitute for thorough bias auditing across diverse domains, languages, or culturally sensitive contexts. The 225-pair size enables rapid screening but limits statistical power for fine-grained subgroup analyses.

B Full Results Tables

C Experimental Infrastructure

Gemini models were accessed via the `google-genai` SDK with Vertex AI (`vertexai=True`, location: `global`); model IDs: `gemini-2.5-pro` and `gemini-2.5-flash` (alias IDs that resolve to the latest stable checkpoint at query time). Claude Sonnet 4 was accessed via the `AnthropicVertex` client (model: `claude-sonnet-4@20250514`, region: `global`). GPT-4o was accessed via the OpenAI API (model: `gpt-4o`; alias ID, exact checkpoint not pinned); token counts are estimated (1 token per 4 characters) as usage metadata was not consistently available. Llama 3.3-70B was accessed via Vertex AI Model Garden as a Model-as-a-Service (MaaS) endpoint using the same `google-genai` SDK (model: `meta/llama-3.3-70b-instruct-maas`, location: `us-central1`), incurring zero API cost. Experiments were conducted between April and June 2025; alias model IDs may resolve to different checkpoints over time, which is an inherent reproducibility limitation of API-based evaluation. Gemini API calls used structured JSON output (`response_mime_type="application/json"`); all other models parse JSON from free-text responses with fallback extraction. Concurrency was limited to 10 parallel requests via asyncio semaphores. Individual results were cached to disk, enabling resumable runs across sessions.

Table 3: Verdict distributions on MT-Bench ($n = 400$). Tie rates are generally moderate (3–31%).

Model	Strategy	A	B	Tie	Tie%
Gemini 2.5 Pro	Baseline	219	171	10	3%
Gemini 2.5 Pro	Pos. Swap	178	164	58	15%
Gemini 2.5 Pro	Rubric	205	174	21	5%
Gemini 2.5 Pro	CoT	213	175	12	3%
Gemini 2.5 Pro	Combined	180	163	57	14%
Claude Sonnet 4	Baseline	162	142	96	24%
Claude Sonnet 4	Pos. Swap	141	134	125	31%
Claude Sonnet 4	Rubric	155	153	92	23%
Claude Sonnet 4	CoT	200	185	15	4%
Claude Sonnet 4	Combined	182	164	54	14%
GPT-4o	Baseline	199	166	34	9%
GPT-4o	Pos. Swap	163	149	88	22%
GPT-4o	Rubric	203	174	23	6%
GPT-4o	CoT	204	179	17	4%
GPT-4o	Combined	173	160	67	17%
Llama 3.3-70B	Baseline	202	183	15	4%
Llama 3.3-70B	Pos. Swap	175	159	66	17%
Llama 3.3-70B	Rubric	203	167	30	8%
Llama 3.3-70B	CoT	197	188	15	4%
Llama 3.3-70B	Combined	165	158	77	19%
Gemini 2.5 Flash	Baseline	207	181	12	3%
Gemini 2.5 Flash	Pos. Swap	180	162	58	15%
Gemini 2.5 Flash	Rubric	203	171	26	7%
Gemini 2.5 Flash	CoT	212	175	13	3%
Gemini 2.5 Flash	Combined	162	163	75	19%
Gold labels		*158*	*145*	*97*	*24%*

D Statistical Tests

Table 4: McNemar's test (with continuity correction) comparing baseline to best strategy for each model on MT-Bench ($n = 400$). b: baseline correct, strategy wrong. c: baseline wrong, strategy correct. *Significant at $p < 0.05$ (unadjusted); †survives Holm-Bonferroni correction ($m = 7$).

Model	Comparison	b	c	χ^2	p-value
Claude Sonnet 4	B0 vs. S8	22	67	21.75	$<0.0001^{*\dagger}$
Claude Sonnet 4	B0 vs. S5	33	62	8.25	$0.004^{*\dagger}$
Gemini 2.5 Pro	B0 vs. S1	14	32	6.28	0.012^{*}
Gemini 2.5 Pro	B0 vs. S8	20	29	1.31	0.253
Llama 3.3-70B	B0 vs. S8	26	41	2.93	0.087
GPT-4o	B0 vs. S8	29	39	1.19	0.275
Gemini 2.5 Flash	B0 vs. S1	19	29	1.69	0.194

Multiple testing. We report 7 McNemar tests without family-wise correction. Under Holm-Bonferroni correction ($\alpha = 0.05$, $m = 7$), Claude S8 ($p < 0.0001$) and Claude S5 ($p = 0.004$) remain significant. Pro S1 ($p = 0.012$) does not survive correction (adjusted threshold: $0.05/5 = 0.010$); we retain it as marginally significant and note the unadjusted p-value throughout.

Aggregate directional evidence. As a complementary analysis, we apply a sign test across all 20 non-baseline model-strategy configurations in Table 1: 18 of 20 show positive improvement over baseline, yielding $p < 0.001$ under the null hypothesis of random direction. Note that these 20 configurations are not fully independent, as they share the same underlying 400 evaluation instances; the sign test should therefore be interpreted as supportive evidence rather than a strict hypothesis test. This confirms that debiasing strategies have a systematically positive effect on agreement, though individual improvements vary in magnitude and statistical significance.

Table 5: Cohen's κ on MT-Bench ($n = 400$). Claude S8 achieves the highest inter-rater agreement ($\kappa = 0.530$).

Model	Strategy	κ	Interpretation
Gemini 2.5 Pro	Baseline	0.442	Moderate
Gemini 2.5 Pro	Pos. Swap	0.528	Moderate
Gemini 2.5 Pro	Rubric	0.486	Moderate
Gemini 2.5 Pro	CoT	0.439	Moderate
Gemini 2.5 Pro	Combined	0.492	Moderate
Claude Sonnet 4	Baseline	0.368	Fair
Claude Sonnet 4	Rubric	0.395	Fair
Claude Sonnet 4	CoT	0.457	Moderate
Claude Sonnet 4	Combined	0.530	Moderate
GPT-4o	Baseline	0.421	Moderate
GPT-4o	Pos. Swap	0.401	Fair
GPT-4o	Rubric	0.419	Moderate
GPT-4o	CoT	0.437	Moderate
GPT-4o	Combined	0.476	Moderate
Llama 3.3-70B	Baseline	0.437	Moderate
Llama 3.3-70B	Combined	0.513	Moderate
Gemini 2.5 Flash	Baseline	0.423	Moderate
Gemini 2.5 Flash	Pos. Swap	0.477	Moderate
Gemini 2.5 Flash	CoT	0.463	Moderate
Gemini 2.5 Flash	Combined	0.463	Moderate

E Per-Category Analysis

Performance varies substantially across MT-Bench categories: extraction and writing tend to have higher agreement, while math and reasoning are more challenging for all models. Llama 3.3-70B performs comparably to proprietary models on writing and extraction but lags on math-heavy tasks.

F Verbosity: Expansion vs. Truncation Details

Table 6: LENGTH results split by construction direction. Expansion bias: preference for longer (+) or shorter (−) version when filler is added ($n = 50$). Truncation accuracy: fraction correctly preferring the complete version ($n = 25$).

	Expansion Bias			Truncation Accuracy		
Model	B0	S5	S8	B0	S5	S8
Gemini 2.5 Pro	−0.24	−0.24	−0.36	0.96	1.00	0.84
Claude Sonnet 4	−0.76	−0.62	−0.54	1.00	1.00	1.00
GPT-4o	−0.24	−0.30	−0.26	1.00	0.88	0.92
Llama 3.3-70B	−0.20	−0.16	−0.02	0.92	1.00	1.00
Gemini 2.5 Flash	−0.36	−0.34	−0.34	1.00	0.92	0.96

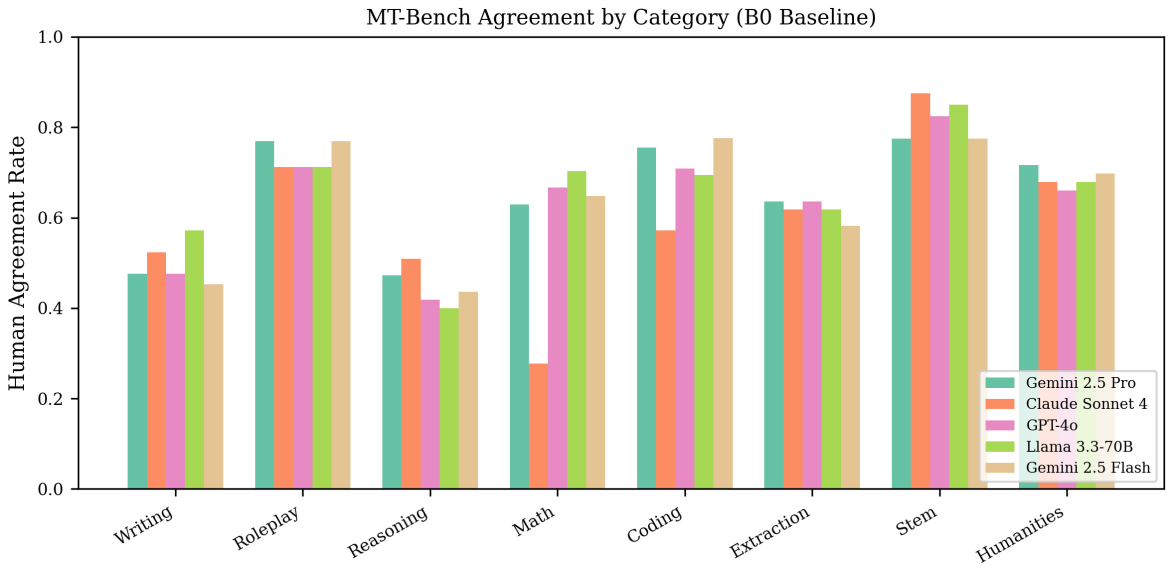

Figure 3: MT-Bench human agreement by category (B0 baseline, $n = 400$ total). Performance varies across categories, with extraction and writing showing higher agreement across all models. Only baseline (B0) is shown; per-strategy breakdowns by category are available in the released artifacts but omitted here as the small per-category sample sizes ($n \approx 50$) yield wide confidence intervals that preclude reliable strategy comparisons at the category level.

CoT (S5) and combined (S8) reduce expansion bias for Claude (from -0.76 to -0.62 and -0.54 respectively) while maintaining high truncation accuracy. Llama with S8 nearly eliminates expansion bias (-0.02) while achieving perfect truncation accuracy (1.00), suggesting the combined strategy particularly helps open-source models calibrate length sensitivity.

G Cost-Accuracy Tradeoff

Claude Sonnet 4 with S8 achieves the highest agreement (70.0%, $\kappa = 0.530$) at ∼\$0.015/eval. Gemini Pro with S1 achieves 69.8% ($\kappa = 0.528$) at ∼\$0.002/eval. At zero API cost, Llama 3.3-70B with S8 achieves 68.5% ($\kappa = 0.513$) via Vertex AI Model Garden MaaS.

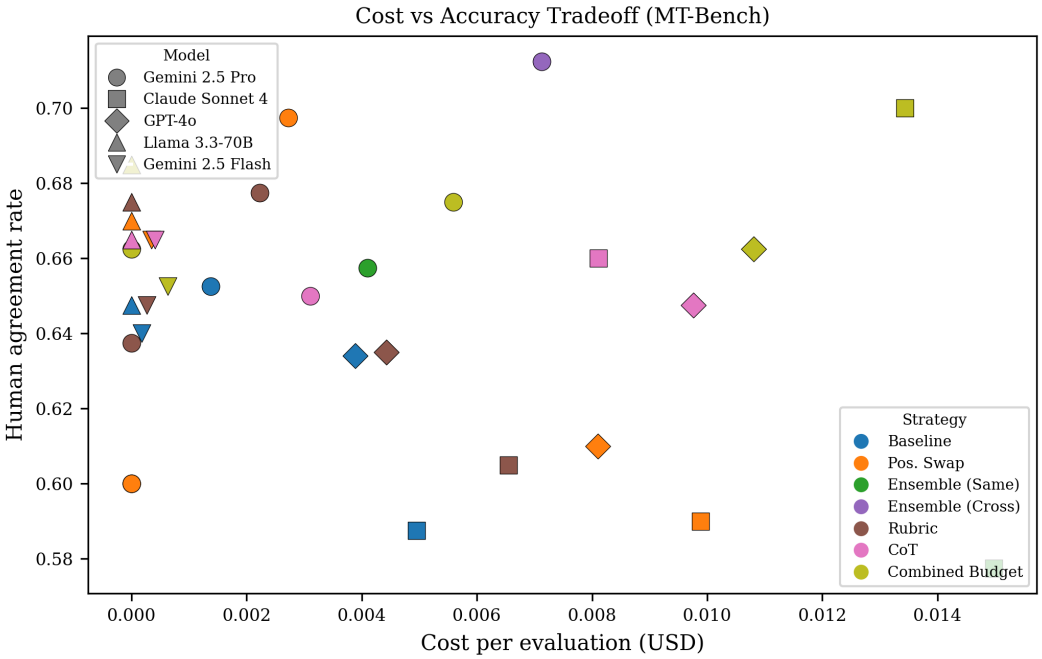

Figure 4: Cost vs. accuracy Pareto frontier on MT-Bench. Claude S8 achieves 70.0% at ∼$0.015/eval. Pro S1 achieves 69.8% at ∼$0.002/eval. Llama S8 achieves 68.5% at zero API cost.