Supplementary Information for: Hidden Measurement Error in LLM Pipelines Distorts Annotation, Evaluation, and Benchmarking

Solomon Messing*
Center for Social Media, AI, and Politics, New York University
MLCommons

A Quickstart Guide

The steps below show how to diagnose uncertainty in an LLM annotation workflow. Section C gives the framework and its derivation.

1. **Run a small pilot.** A small pilot under fixed temperature conditions may give an idea of the major sources of variance when the judge layer is manipulated. The recommended design has 30 or more items, 3 or more prompts, 2 or more judges (in case a judge layer exists), and 3 or more replications, resulting in about 540 API calls per single scoring function tested. Two prompts are the bare minimum, at the cost of less precise estimates of prompt sensitivity (Section E.1). Prompt variants have identical task instructions, output format, and reasoning mode, but might differ in terms of word usage, sentence structure, and framing of the construct (Assumption 1, Section C). Varying the prompts this widely reflects the uncertainty that a real human evaluator would face in choosing the wording. Both paraphrasing (for example, prompting Claude Haiku at temperature 0.9 to rewrite the seed instruction in V ways) and sets of a few templates varying framing of the construct are legitimate options. The latter method results in a more inclusive $\hat{\sigma}^2_\phi$. It is recommended to include the original benchmark prompt in the list of V variants where available. Items should come from the target deployment domain. It is also recommended to exclude the judge facet for benchmarks without one (e.g., MMLU).

2. **Fit the variance decomposition.** The `totalevalerror` R package (`https://github.com/SolomonMg/totalevalerror`) wraps the variance decomposition, D-study, and budget-allocation routines from this paper:

   ```
   library(totalevalerror)
   des <- tee_design(pilot_df,
           outcome = score, item = item_id,
           prompt = variant_id, judge = judge_model,
           temperature = temp, replicate = rep,
           scoring = "binary")
   fit <- tee_decompose(des)
   ```

   The fit reports each source of variance as a share of $\mathrm{Var}(\hat{\theta})$, and the largest share shows where the bottleneck is. If judge variance (σ^2_λ) or item×judge ($\sigma^2_{\alpha\lambda}$) dominates, as in the safety demonstration, adding judges helps most. If within-cell replicate noise (σ^2_ρ) dominates, add replications. If prompt sensitivity (σ^2_ϕ) dominates, add prompt variants. If item heterogeneity (σ^2_α) dominates, add items. The underlying fixed-temperature `lmer` formula is `score ~ judge + (1|item) + (1|prompt_variant) + (1|item:prompt_variant) + (1|item:judge) + (1|prompt_variant:judge) + (1|item:prompt_variant:judge)`, with the judge terms dropped for benchmarks

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

without a judge layer. The final random effect captures cell-level idiosyncrasy; the `lmer` residual captures within-cell replicate noise. Users who prefer to fit this formula directly can do so.

3. **Optimize with the D-study.** `tee_dstudy(fit)` gives projected SEs across alternative production designs (varying replications, prompt variants, judges, scoring methods). `tee_budget_allocate(fit, budget = X)` returns the cost-minimizing allocation under a fixed API budget. The projection is qualitatively reliable at pilot scale (Section I). The rank ordering of interventions transfers to the full design even when component magnitudes shift.

4. **Deploy and estimate CIs.** Apply the chosen production design to the full corpus, then call `tee_se(fit)` for the corrected SE and `tee_theta_ci(fit, level = 0.95)` for a 95% CI on $\hat{\theta}$. Because the formula incorporates the components naive CIs leave out (prompt and judge variance, plus their interactions), coverage holds near nominal even as sample size grows.

5. **Report.** `tee_writeup(fit)` produces a paragraph summarizing the decomposition, the dominant variance components, the corrected CI, and the D-study-recommended design. For each scored measurement, also report the exact prompt text, model version, temperature, and number of replications. A tiered reporting checklist appears in Section M, Table 20.

B Uncertainty Reporting in Current LLM Evaluation Practice

Table 1 surveys uncertainty reporting across thirteen widely used LLM benchmarks and annotation studies. Looking across these 13 benchmarks, most report no measurement uncertainty at all: MMLU, the Open LLM Leaderboard, MT-Bench, SWE-bench, and ARC-AGI all report only a point estimate. And the four benchmarks that do include CIs (AlpacaEval, HELM, Chatbot Arena, and Arena-Hard) only account for item-sampling variance, leaving prompt sensitivity, judge variance, temperature, and interaction terms unmeasured. Contamination in test sets occurs frequently: out of six benchmarks with publicly available test sets (MMLU, AlpacaEval, HELM, MT-Bench, Arena-Hard, and SWE-bench), training data contamination was reported in all, and even Chatbot Arena can suffer from contamination due to historical votes leaking into training data despite the live queries. In our survey, only AILuminate's private Official set and ARC-AGI held-out tasks are contaminations resistant.

The bottom row represents TEE dealing with all TEE-decomposition columns at once: prompt sensitivity, judge model and temperature, replications, interactions, item sampling, and basic CI/SE indicator. The rest of this section considers each benchmark in details.

Table 1: Uncertainty reporting in prominent LLM evaluations. SUT-layer columns capture variance in the system under test. Judge-layer columns capture variance in the scoring pipeline.

			SUT layer			*Judge layer*				
	Any CI/SE	*Item sampling*	*SUT prompt*	*SUT temperature*	*SUT replications*	*Judge prompt*	*Judge model*	*Judge temperature*	*Interactions*	*Contamination*
MMLU [Hendrycks et al., 2021]	-	-	-	-	-	n/a	n/a	n/a	-	High
Open LLM Leaderboard	-	-	-	-	-	n/a	n/a	n/a	-	Mixed
AlpacaEval [Dubois et al., 2024]	⋆	⋆	-	-	-	-	-	-	-	High
HELM [Liang et al., 2023]	⋆	⋆	-	-	-	n/a	n/a	n/a	-	High
Chatbot Arena [Chiang et al., 2024]	⋆	⋆	-	-	-	n/a	n/a	n/a	-	Medium
MT-Bench [Zheng et al., 2023]	-	-	-	-	-	-	-	-	-	High
Arena-Hard [Li et al., 2024]	⋆	⋆	-	-	-	-	-	-	-	High
SWE-bench [Jimenez et al., 2024]	-	-	-	-	-	n/a	n/a	n/a	-	High
ARC-AGI [Chollet, 2019]	-	-	-	-	-	n/a	n/a	n/a	-	Low
AILuminate [Ghosh et al., 2025]	(⋆)	-	-	-	-	-	(⋆)	-	-	Medium
Gilardi et al. [Gilardi et al., 2023]	-	-	n/a	n/a	n/a	-	-	-	-	n/a
Törnberg [Törnberg, 2025]	-	-	n/a	n/a	n/a	-	-	-	-	n/a
Ziems et al. [Ziems et al., 2024]	-	-	n/a	n/a	n/a	(⋆)	(⋆)	-	-	n/a
TEE	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	⋆	n/a

Deterministic scoring (exact match or test suite), no judge layer.
Public test set with documented training-data leakage [Xu et al., 2024].
Most subtests public. GPQA gated to reduce contamination.
Human voters, not an LLM judge layer.
Live queries, but historical votes published and prompts recur across months.
Bootstrap CI over items, captures item sampling only.
Public task set with documented solution leakage into training data.
Private held-out evaluation set with human-calibrated difficulty.
Paper specifies bounds from evaluator false-safe/false-unsafe rates. Ensemble of evaluators fixed. Public leaderboard displays grade only.
Public DEMO subset (1,200 prompts) and public Practice set (12,000), with private Official test set (12,000).

MMLU and the Open LLM Leaderboard. MMLU [Hendrycks et al., 2021] evaluates each model once on a fixed test set (14,079 items) with a single prompt format. Open-weight models with access to logits score according to log-likelihoods of the answer choices, while API-only models use the evaluation process of text generation and answer-letter parsing, which brings back the role of temperature and replicate noise. In both cases, no variability information is provided. The main source of missing variability is the item heterogeneity. From our 200-item factorial decomposition (SI Appendix, Section J), within-category item heterogeneity accounts for 72.2% of $\mathrm{Var}(\hat{\theta})$, item$\times$SUT 10.5%, between-category heterogeneity 7.7%, and SUT-model design sensitivity 6.7% at the operational design, while the three prompt terms together account for 2.7%. According to Sclar et al. [2024], the differences in formatting may yield up to 76 accuracy points variability. These formatting perturbations are different from phrasing-and-framing variations used in this paper (Assumption 1). Different implementations of MMLU (original, HELM, HuggingFace) have different prompt templates and yield different rankings of the same models [Alzahrani et al., 2024]. It is also

the case for the Open LLM Leaderboard. Most of its subtests use publicly available test sets, except for GPQA, where it is gated.

AlpacaEval. AlpacaEval 2.0 [Dubois et al., 2024] scores model outputs on a reference set using only one LLM judge (GPT-4 Turbo) with one prompt, and calculates the standard error that captures variance due to item sampling. Every other pipeline parameter is kept constant (only one judge, one prompt, one temperature, and no replication). A different judge would lead to a different win rate; however, judge variance (σ^2_λ) is not estimated. The length controlled version adjusts the scoring for response length but not for pipeline parameter variance.

HELM. HELM [Liang et al., 2023] assesses each model for each scenario using a fixed format of the prompt and 3 random seeds on different in-context example sets to ensure stability. All prompts, predictions, and test items are publicly available; therefore, HELM is subject to test set contamination due to data leakage during the training process [Xu et al., 2024]. The evaluation relies on deterministic automated metrics (accuracy, BLEU, ROUGE, F1), not LLM judges, and thus there is no need to estimate judge variance. The sensitivity of the SUT to the prompt and temperature applies but is not estimated.

Chatbot Arena. Chatbot Arena [Chiang et al., 2024] computes Bradley-Terry [Bradley and Terry, 1952] coefficients (labeled as Elo ratings) via maximum likelihood estimation with sandwich robust standard errors, collecting an average of roughly 8,000 votes per model. Prompt phrasing emerges organically among users but is neither controlled nor deconstructed. Voter heterogeneity (human rankings differ) and SUT temperature (fixed per model by the provider) both hold but are not separated from sampling error. Bootstrapping captures uncertainty in vote sample but not the variant-selection effects noted by Singh et al. [2025], where dozens of private SUT variants were tried by providers, and only the top-scoring variant was made public.

MT-Bench. MT-Bench [Zheng et al., 2023] tests SUTs on 80 multi-turn questions (2 turns each; 8 categories) scored from 1 to 10 by GPT-4. There is no specific judge model version tied to the codebase; hence, an issue of replicability since the alias refers to changing model versions. The test uses only one prompt template per question at judge temperature 0 with no replicates and no CI. SUT temperature varies by category (from 0.0 to 0.7). According to the LMSYS team's later study of their benchmark, only 22.6% of model pairs have non-overlapping CIs on MT-Bench versus 87.4% on Arena-Hard [Li et al., 2024]; hence, the scores do not statistically discriminate most model pairs.

Arena-Hard. Arena-Hard-Auto [Li et al., 2024] evaluates SUTs on 500 challenging user queries drawn from Chatbot Arena, scored by GPT-4-Turbo in pairwise comparison against a baseline (GPT-4-0314). The two-step position swap procedure helps mitigate bias from the placement of the model. The bootstrap confidence intervals involve 100 resamples and show variability of the sampled data without taking into account the variability of the judgment; the experiment included only one judge, one template prompt, and a temperature of 0. The Arena-Hard benchmark obtained a separation score of 87.4%, which is the highest among automated benchmarks considered. Its test set is public.

SWE-bench. The SWE-bench family [Jimenez et al., 2024] evaluates coding agents on real GitHub issues. SWE-bench Verified (500 human-validated Python tasks) was the standard benchmark until contamination was found across all frontier models. SWE-bench Pro provides an alternative that is more difficult and less polluted: it has 1,865 tasks across 41 repositories, where each one needs at least $\geq$10 lines changes and averages 4.1 files being repaired per task. The score is deterministic as the agent's repair is applied to the repository and then validated against the test cases using Pass@1 that assesses the pass rate after the first attempt. There is no judge, and there is no confidence interval. Differences in format, temperature, and number of replications may still affect the patches.

ARC-AGI. The ARC-AGI benchmark [Chollet, 2019] evaluates abstract reasoning on grid-transformation tasks across three versions of increasing difficulty: ARC-AGI-1 (2019, now largely solved with frontier models exceeding 96%), ARC-AGI-2 (2025, where frontier models reach $\sim$77%), and ARC-AGI-3 (2026, interactive reasoning). In the static versions (1 and 2), scoring is done through an exact match, and two chances are provided for each input. In contrast, version 3 measures behavior. Contamination is minimized using a separate set of 100 - 120 tasks for each version. Task difficulty is calibrated through human studies; more than 400 participants were involved

in task calibration in version 2. ARC-AGI-2 adds another measure: cost per task. The score itself remains a single point estimate, with no confidence interval or uncertainty decomposition.

AILuminate. AILuminate v1.0 [Ghosh et al., 2025] evaluates SUT safety across 12 hazard categories (1,000 prompts each in the public Practice set and 12,000 in the private Official set, plus a 1,200-prompt Creative Commons DEMO subset). Scoring involves the use of an unchanging set of safety evaluator models, which indicate violations. The final scores (Poor/Fair/Good/Very Good/Excellent) depend on how many violations there are compared to a reference model. The technical paper specifies a methodology for computing upper and lower score bounds from evaluator false-safe and false-unsafe rates (uniquely among the benchmarks surveyed here), but the public leaderboard displays only the categorical grade. The methodology does not decompose factors such as item sampling, prompt sensitivity, evaluator-model selection (the ensemble is fixed), and their interactions, and the authors recognize that there is "considerable variance in test outcomes" because of prompt sampling and noise in the evaluator.

LLM-as-annotator studies. Gilardi et al. [2023] compare ChatGPT (GPT-3.5-turbo) annotations to trained research-assistant labels on political text classification, reporting accuracy and intercoder agreement (Cohen's κ, Cohen 1960). The study does vary temperature (1.0 and 0.2) and collects two responses per setting, finding 91% intercoder agreement between two ChatGPT runs at $T = 1$ and 97% at $T = 0.2$. Gilardi et al. is a methods validation paper and not a downstream analysis; therefore, the absence of pipeline-variance CIs is not by itself a problem. Subsequent annotation studies follow the same single-configuration practice: Törnberg [2025] uses GPT-4 with a single zero-shot prompt on political tweets. Heseltine and Clemm von Hohenberg [2024] compare GPT-4 expert coding across four countries with a single prompt and temperature. The guidelines for management research provided by Carlson and Burbano [2026] reveal the dangers of small decisions in implementations, as these can greatly change the annotation results of LLMs and lead to changes in downstream results. The above single-configuration approach has become popular in downstream hypothesis testing, where LLM annotations act as regression variables and the pipeline variance TEE quantifies directly affects statistical conclusions [Baumann et al., 2025, Egami et al., 2023].

C Full Framework: Assumptions, Derivations, and Extensions

This section gives the formal assumptions, the variance decomposition, the two-tier classification, the heteroscedastic generalization, and our approach to estimation.

C.1 Model Terms and Factor Classification

Table 2 categorises the factors in the TEE data-generating process.

Table 2: Factors in the TEE data-generating process (DGP), categorised as fixed or random. Random factors (item, prompt variant, replication) stand for exchangeable draws from a larger populations, with their variance components reducing as they are aggregated *inside a chosen design* (Tier 1). Fixed factors (temperature, model, scoring method) stands for specific researcher choices. When no aggregation happens, their sensitivity indices quantify researcher degrees of freedom (Tier 2). When the design *does* cross multiple levels of a fixed factor (e.g., averaging over a panel of judges), the corresponding sensitivity contribution also reduces as $1/L$. This aggregation across designs is what the empirical sections depend upon for judge-model averaging.

Factor	Type	Levels	Examples
Item i	Random	N (large)	A specific social media post; a safety test prompt together with the SUT response being judged
Category c	Random grouping	C	Hate speech type, hazard category
Prompt variant v	Random	V (3 - 5)	"Is this post hateful?" vs. "Does this contain hate speech?"
Temperature h	Fixed	H (e.g., 3)	$T = 0.0, 0.7, 1.0$
Model m	Fixed	Small set	GPT-4o, Claude Haiku 4.5, Gemini 2.0 Flash
Scoring method	Fixed (Tier 2)	Typically 1	Exact match, Likert, LLM-as-judge
Replication r	Random (nested)	R (3 - 10)	Repeated API call with identical inputs

The unit of analysis is one "item" (a benchmark case, social media post, or something similar) scored once at one combination of prompt variant, temperature, model, and replication. A typical pipeline involves two LLM calls, with the system under test (SUT) generating an output and a judge scoring it. The DGP is applied on a per-layer basis. At the judge layer, "item" is a (prompt, SUT-response) pair; at the SUT layer, "model" indexes SUTs and the variance components capture SUT-side properties. The default target estimand is $\theta_{hm} = \mathrm{E}_{i,v,r}[Y^{(r)}_{ivhm}] = \mu + \tau_h + \lambda_m$, the expected value of the scored output over exchangeable items, prompt variants, and replications, conditional on fixed design choices. TEE quantifies how much uncertainty each step of the pipeline adds to $\hat{\theta}_{hm}$.

We treat a factor as random if its levels are exchangeable and as fixed if they are not. Temperature is fixed because its levels produce different generation modes and cannot be exchanged (see Section O). Prompt variants are random because, under Assumption 1, we do not expect any wording or framing to be consistently better than the others, as long as all variants share the same reasoning mode and output format. Scoring method is fixed because it defines the residual distribution and relates to prompt sensitivity.

When items belong to content categories, $\alpha_i = \kappa_{c(i)} + \delta_{i|c}$, where $\kappa_c \sim N(0, \sigma^2_\kappa)$ captures between-category variance and $\delta_{i|c} \sim N(0, \sigma^2_\delta)$ captures within-category heterogeneity:

$$\sigma^2_\alpha = \sigma^2_\kappa + \sigma^2_\delta \tag{4}$$

C.2 Target Estimand

The target estimand is the expected scored output, given the researcher's fixed choices of design (temperature h, model m):

$$\theta_{hm} = \mathrm{E}_{i,v,r}\left[Y^{(r)}_{ivhm}\right] = \mu + \tau_h + \lambda_m \tag{5}$$

where the expectation averages over random factors such as items, prompt phrasings, and replications while keeping temperature and model fixed. In generalizability theory, θ_{hm} is the "universe

Table 3: Variance components exploitable in single-configuration benchmarks (Likert decomposition of Chatbot Arena, aggregated across the four categories). Empirical shares from the full-factorial decomposition (4,676 matches, 5 prompt variants, 3 judges). Shown as proportions of per-observation variance σ^2_{total} as gaming leverages single-configuration noise directly, before averaging reduces any component. "How it inflates scores" refers to two channels: (1) passive inflation, where the benchmark operator's single-configuration design leaves noise that increases CIs width for all submissions, and (2) active exploitation, where a developer who can probe the benchmark's specific configuration optimizes against it.

Variance component	**Empirical share**	**How it inflates scores**	**Mitigation**
Prompt sensitivity σ^2_ϕ	1.2%	*Passive:* benchmark's single prompt carries unquantified phrasing noise. *Active:* developer tunes output to the known prompt style	Average $\geq$3 diverse prompt variants
Judge model σ^2_λ (fixed)	8.8%	*Passive:* a single judge's systematic bias shifts all scores. *Active:* developer optimizes for the specific judge's choices	Multi-judge average or report all judges
Item$\times$prompt $\sigma^2_{\alpha\phi}$	2.5%	*Passive:* item-specific phrasing noise inflates CIs. *Active:* developer targets responses the chosen prompt phrasing scores leniently	Multi-prompt averaging
Item$\times$judge $\sigma^2_{\alpha\lambda}$	18.0%	*Passive:* a single judge's item-level idiosyncrasies inflate CIs. *Active:* developer fine-tunes on items the chosen judge scores leniently	Multi-judge averaging eliminates this interaction
Prompt$\times$judge $\sigma^2_{\phi\lambda}$	0.7%	Specific (judge, prompt) cell biases compound when both are fixed	Multi-prompt $\times$ multi-judge averaging

score" [Bayerl and Paul, 2007, Brennan, 2001]. When averaging over H selected temperature levels, $\theta_m = \frac{1}{H}\sum_h \theta_{hm} = \mu + \bar{\tau} + \lambda_m$ is a finite average over those chosen levels. Similar to this, a multi-judge target stands as a finite average over the selected judge panel, with σ^2_λ measuring sensitivity inside that panel rather than a superpopulation variance over all possible judges.

Some benchmark reports focus only on a fixed set of available items. In this finite-benchmark case, item main effects are treated as fixed rather than sampled, so the σ^2_α/N' term does not need to appear in the D-study formula anymore. The remaining prompt, judge, temperature, interaction, cell-level, and replicate terms still capture how much the reported score is dependent on the chosen measurement pipeline for that particular item set.

TEE can be applied in three types of situations. The first situation is when the measurement is the construct itself, such as in LLM benchmarking and LLM-as-judge tasks where the scores of the generated output will determine the measurement variable. The second scenario is when the pipeline stability is a goal itself; where, despite there being no bias in the pipeline in the mean sense, the high variance of its measurements can render the pipeline outputs unusable. The third scenario is when LLM labels are used as substitutes in downstream analysis. This θ_{hm} serves as a proxy to an external ground truth $\theta^* = \mathrm{E}[Y^*]$, and the surrogate labeling error $\tilde{Y} - Y^*$ contributes bias and variance that is not accounted for by TEE alone. In Section N, we elaborate on the formal connection to the DSL framework [Egami et al., 2023], explaining which TEE terms correspond to components of the surrogate labeling variance.

Item heterogeneity (σ^2_α) is signal added with noise. The signal represents differences in item labels that exist due to measured construct and not because of any defect in the process of measurement. A D-study recommending that we add more items to achieve higher precision (reduces $\mathrm{Var}(\hat{\theta}_{hm})$) does not help in achieving higher validity (the distance between θ_{hm} and θ^*). Separating signal component of σ^2_α from the noise component needs an external anchor such as human labels, known-answer items, or a validation sample, and is not in TEE's scope.

These error bars will not be able to project outside of the population of items specified in the estimand. The error bars are also dependent on the assumption that the sampling frame has all the items of interest within it. Despite having proper error bars for TEE, incorrect estimation can still take place if some hazard categories are omitted from the item list. Section Q discusses this boundary.

C.3 Assumptions

The decomposition requires the following assumptions:

Assumption 1 (Conditional Exchangeability)**.** *Replications $r = 1, \ldots, R$ are exchangeable given (i, v, h, m). Infrastructure nondeterminism [Yuan et al., 2025] that varies across repeated calls is absorbed into $\rho^{(r)}_{ivhm}$; stable cell-level idiosyncrasy from omitted higher-order interactions is absorbed into ϵ_{ivhm}. Similarly, prompt variants $v = 1, \ldots, V$ are exchangeable given item i. In other words, σ^2_ϕ captures sensitivity to both phrasing and framing within a fixed task instruction, output format, and reasoning mode. Variants share the canonical task and rubric structure but may differ in word choice, sentence structure, and the specific framing of the underlying construct.*

In the empirical demonstrations, prompt variants were authored as a small set of templates that vary framing while preserving task and output format, with the original benchmark prompt included where one exists; the propaganda demonstration also used an LLM paraphrase pipeline (`anthropic/claude-haiku-4.5`, temperature 0.9) on a seed instruction [Mizrahi et al., 2024, Barrie et al., 2024]. The safety and MMLU variants passed a deterministic format check and a construct-equivalence screen computed independently of the scores; one MMLU variant failed the format check and is excluded (Section J). The Arena and propaganda variants predate the screen and were not screened. Section K reports the screen, its validation against off-construct decoys, and how $\hat{\sigma}^2_\phi + \hat{\sigma}^2_{\alpha\phi}$ moves as the variant universe widens from strict paraphrase to construct reframing (flat for safety; 2.1% to 20.6% of G-study variance for MMLU). The reported $\hat{\sigma}^2_\phi$ is therefore an upper bound on what a strict-paraphrase design would estimate and a lower bound relative to broader framing sets or designs that also cross output format or reasoning mode; simulation confirms that D-study projections hold under exchangeability violations of this kind (Section E.2, Scenario 3).

Assumption 2 (Additivity with Two-Way Interactions)**.** *Effects are additive up to two-way interactions. Higher-order interactions are absorbed into the residual. Monte Carlo evidence (Section E.1) shows that D-study projections tolerate three-way interactions with magnitudes up to 100% of the largest two-way component. The restriction is testable via a likelihood ratio test. The compact formulas also omit fixed-by-fixed interactions such as temperature×model. When a design crosses multiple fixed factors, and their interaction is substantively large, add the corresponding finite sensitivity term to the D-study formula.*

Assumption 3 (Distributional Assumptions)**.** *All random effects are normally distributed with mean zero and mutually independent. The normality assumption can be relaxed for REML estimation [Jiang, 1996]. Mutual independence is the more substantive assumption, with two violations warranting scrutiny. First, the model assumes item difficulty is independent of prompt sensitivity ($\alpha_i \perp (\alpha\phi)_{iv}$), yet ambiguous items may be more prompt-sensitive. Second, the model assumes difficulty is independent of temperature sensitivity ($\alpha_i \perp (\alpha\tau)_{ih}$), yet items near a decision boundary should be where temperature matters most. Section E.2, Scenario 2 shows that even strong dependence produces ≤2% D-study bias. The D-study formulas use pooled interaction variances across fixed levels. If item×judge, item×temperature, residual, or replicate variance differs sharply by fixed level, use stratum-specific variance estimates or the heteroscedastic extension; the pooled formula is inappropriate in that case.*

For binary outcomes, the linear model is a linear probability model (LPM), adequate when baseline rates are moderate (0.3 - 0.7). At extreme rates, a GLMM robustness check is recommended (Section G).

C.4 Variance Decomposition Derivation

The variance of a single observation at fixed h and m follows from mutual independence of random effects:

$$\mathrm{Var}(Y \mid h, m) = \sigma^2_\alpha + \sigma^2_\phi + \sigma^2_{\alpha\phi} + \sigma^2_{\alpha\tau} + \sigma^2_{\phi\tau} + \sigma^2_{\alpha\lambda} + \sigma^2_{\phi\lambda} + \sigma^2_\epsilon + \sigma^2_\rho \tag{6}$$

Averaging over H temperatures introduces a design-sensitivity term:

$$\sigma_\tau^2 = \frac{1}{H}\sum_{h=1}^{H}(\tau_h - \bar{\tau})^2 \tag{7}$$

This is computed from fixed effects, not REML, and depends entirely on which temperature levels the researcher chose. Model choice is analogous, with $\sigma_\lambda^2 = \frac{1}{M}\sum_m(\lambda_m - \bar{\lambda})^2$.

The variance components divide into random-effect components, which shrink with more data, and fixed-effect sensitivity indices, which measure dependence on researcher choices. This partition maps onto the researcher degrees of freedom problem [Simmons et al., 2011], since temperature, scoring method, model, and system prompt are arbitrary choices that could have been made differently. TEE quantifies the sensitivity of $\hat{\theta}$ to each choice, analogous to the multiverse analysis of Steegen et al. [2016]. Further design choices that can vary across runs include scoring or extraction method ($\sigma^2_{\text{extract}}$), system prompt ($\sigma^2_{\text{sys}}$), infrastructure ($\sigma^2_{\text{infra}}$), and temporal drift ($\sigma^2_{\text{drift}}$). Barrie et al. [2025] catalog these. Each enters as an additional fixed factor when the design varies it.

C.5 Per-Observation Variance Decomposition

Often, statistical work decomposes variance at the observation level rather than for an aggregate quantity of interest like $\text{Var}(\hat{\theta})$, as we do in the main text and here in the SI. The per-observation share $\hat{\sigma}_k^2/\sigma^2_{\text{total}}$ characterizes the data-generating process before any averaging takes place, which makes it useful for examining the variance of individual LLM calls (for instance, Table 3 reports per-observation shares to show the single-call variance profile, while the gaming surfaces themselves are estimated via bootstrap in Section 4.4).

When the goal is instead to report an aggregate estimate like the mean $\hat{\theta}$ and understand what drives its standard error, the $\text{Var}(\hat{\theta})$ share is the more relevant summary. Because each component $\hat{\sigma}_k^2$ enters $\text{Var}(\hat{\theta})$ divided by the number of factor levels averaged over (e.g., σ_α^2/N, σ_λ^2/M, $\sigma_\rho^2/(NVHMR)$), a component that looks large at the single-observation level can become negligible in the variance of the mean once divided by a large denominator. In our safety application, for example, replicate noise σ_ρ^2 accounts for 21% of per-observation variance, but less than 0.5% of $\text{Var}(\hat{\theta})$ at $R = 8$ because it is divided by $NVHMR = 50{,}760$. Conversely, judge-model design sensitivity σ_λ^2 is only 0.8% of per-observation variance, yet accounts for 43.8% of $\text{Var}(\hat{\theta})$ because it is only divided by $M = 3$.

Both summaries come from the same REML fit: per-observation shares are useful when comparing the underlying DGP across studies, link functions, or scoring methods, whereas $\text{Var}(\hat{\theta})$ shares directly reflect the standard error of the mean under a specific design and guide budget allocation, judge averaging, and sample-size decisions.

C.6 Estimation, Identifiability, and Diagnostics

Estimation. TEE uses REML via `lme4::lmer` in R. The base `lmer` call (cross-model design):

```
lmer(Y ~ temperature + model +
     (1|item) + (1|prompt_variant) +
     (1|item:prompt_variant) +
     (1|item:temperature) +
     (1|prompt_variant:temperature) +
     (1|item:model) +
     (1|prompt_variant:model) +
     (1|item:prompt_variant:model:temperature),
     data = df)
```

Under category nesting, add `(1|category)`. Variance components are extracted via `VarCorr()`. Fixed-effect sensitivity indices use population variance ($1/H$, not $1/(H-1)$).

Boundary estimates. Variance components at exactly zero (singular fits) occur routinely when the true variance is small. Report boundary estimates transparently. Bayesian alternatives with half-t priors [Gelman, 2006] avoid boundary issues.

Identifiability. The model requires a factorial design with every item evaluated under every prompt variant, at every temperature, and by every model. The recommended design uses $N \geq 30$ items, $V \geq 3$ prompt variants, and $R \geq 3$ replications, and $V = 2$ is the bare minimum. Prompt sensitivity (σ^2_ϕ) is the hardest to estimate precisely. For $\sigma^2_\phi > 0$, moving from $V = 2$ to $V = 3$ cuts RMSE by about a third in simulation (Section E.1).

Diagnostics. (1) QQ plots of BLUPs [best linear unbiased predictions; Robinson, 1991] for normality. (2) Residual-vs-fitted plots for heteroscedasticity. (3) Three-way interaction test via likelihood ratio. (4) BLUP magnitude vs. item difficulty for independence violations. (5) Category-level interaction decomposition.

C.7 Heteroscedastic Extension

The base model assumes common σ^2_ϵ and σ^2_ρ across temperatures. In practice, greedy decoding and sampled decoding can produce different cell-level and replicate-level variances. The heteroscedastic extension replaces the common terms with $\epsilon_{ivhm} \sim N(0, \sigma^2_{\epsilon,h})$ and $\rho^{(r)}_{ivhm} \sim N(0, \sigma^2_{\rho,h})$. We can estimate these variances with per-stratum REML in `lme4::lmer` (near-zero bias, <0.3%) or with `nlme::lme()` [Pinheiro and Bates, 2000] and `weights = varIdent`.

C.8 D-Study Formulas: Full Derivation

For a single model at fixed temperature, the projected variance of $\hat{\theta}_{hm}$ is

$$\mathrm{Var}(\hat{\theta}_{hm}) = \frac{\sigma^2_\alpha}{N'} + \frac{\sigma^2_\phi}{V'} + \frac{\sigma^2_{\alpha\phi}}{N'V'} + \frac{\sigma^2_{\alpha\tau}}{N'} + \frac{\sigma^2_{\phi\tau}}{V'} + \frac{\sigma^2_{\epsilon,h}}{N'V'} + \frac{\sigma^2_{\rho,h}}{N'V'R'}. \tag{8}$$

Under category nesting, $\sigma^2_\alpha \to \sigma^2_\delta$ plus a σ^2_κ / C' term, where C' is the number of categories in the projected design. Averaging over H temperatures gives

$$\mathrm{Var}(\hat{\theta}_m) = \frac{\sigma^2_\alpha}{N'} + \frac{\sigma^2_\phi}{V'} + \frac{\sigma^2_\tau}{H} + \frac{\sigma^2_{\alpha\phi}}{N'V'} + \frac{\sigma^2_{\alpha\tau}}{N'H} + \frac{\sigma^2_{\phi\tau}}{V'H} + \frac{\bar{\sigma}^2_\epsilon}{N'V'H} + \frac{\bar{\sigma}^2_\rho}{N'V'HR'}. \tag{9}$$

Projecting across M models at fixed temperature adds the model-related interaction terms,

$$\mathrm{Var}(\hat{\theta}_h) = \frac{\sigma^2_\alpha}{N'} + \frac{\sigma^2_\phi}{V'} + \frac{\sigma^2_\lambda}{M} + \frac{\sigma^2_{\alpha\phi}}{N'V'} + \frac{\sigma^2_{\alpha\tau}}{N'} + \frac{\sigma^2_{\phi\tau}}{V'} + \frac{\sigma^2_{\alpha\lambda}}{N'M} + \frac{\sigma^2_{\phi\lambda}}{V'M} + \frac{\bar{\sigma}^2_\epsilon}{N'V'M} + \frac{\bar{\sigma}^2_\rho}{N'V'MR'}. \tag{10}$$

All three formulas follow from the expected mean squares (EMS) of the mixed design [Brennan, 2001, ch. 9]. The EMS table appears in Section D. When averaging jointly over temperature and model, add the finite main-effect terms σ^2_τ / H and σ^2_λ / M. If their fixed-factor interaction is non-negligible, add $\sigma^2_{\tau\lambda}/(HM)$.

The safety classification design from Section 4.1 shows how these formulas convert real variance estimates into a confidence interval and which design changes they flag as high-leverage. The factorial covers $N = 141$ items spanning $C = 12$ hazard categories, with $V = 5$ prompt variants, $H = 3$ temperatures, $M = 3$ judge models, and $R = 8$ replications per cell. Variance components estimated on this design reveal that judge-model design sensitivity dominates at 43.8% of $\mathrm{Var}(\hat{\theta})$, item×judge interaction adds 16.7%, and prompt×judge another 12.4%. The three judge-related components together account for roughly 73% of the variance budget. Within-category item heterogeneity contributes 20.8%, and the remaining components each fall below 5%. Plugging these into the multi-model formula yields $\mathrm{Var}(\hat{\theta}) = 0.000354$, $\mathrm{SE}(\hat{\theta}) = 0.019$, and a 95% CI of 0.942 ± 0.037 for the overall safe rate. The same formula reveals which design changes would shrink that variance further. Adding 5 replications above the existing 8 improves $\mathrm{Var}(\hat{\theta})$ by less than 0.1% because $\sigma^2_\rho/(NVHMR)$ is already a tiny share of the variance budget once divided by all four other factors. Committing to a single judge model inflates $\mathrm{Var}(\hat{\theta})$ by 106% because σ^2_λ loses its $1/M$ averaging benefit. The dominance pattern and the counterfactuals point the same way, so the D-study directs additional spend toward more judges and items.

C.9 Comparison with Concurrent Frameworks

Table 4 compares TEE with concurrent variance decomposition and uncertainty quantification frameworks for LLM evaluation.

Table 4: Feature comparison of TEE with concurrent frameworks for LLM evaluation uncertainty. Stars indicate features present. Dashes indicate absent or not addressed. Cell entries verified against each paper's text and tables.

Feature	**TEE**	**Camuffo (2026)**	**Haase (2026)**	**Song (2025)**	**Wang (2025)**	**NIST (2026)**
G-theory grounded	⋆	⋆	-	⋆	-	-
Interaction terms	⋆	-	-	⋆	-	-
D-study projections	⋆	-	-	⋆	-	-
Two-tier separation	⋆	-	-	-	-	-
Scoring method comparison	⋆	-	-	-	-	-
Multi-domain replication	⋆	-	-	-	-	-
Monte Carlo validation	⋆	-	-	-	-	-
Multi-model factorial	⋆	⋆	⋆	-	-	⋆
Pilot-to-full validation	⋆	-	-	-	-	-
Prompt sensitivity	⋆	⋆	⋆	-	-	-
Temperature as factor	⋆	-	-	-	-	-
# variance components	9+	5	3	varies	2	2
Empirical demonstrations	4	1	1	1	1	-

These frameworks differ primarily in their grounding (G-theory versus ad hoc decompositions) and the number of variance components they separate. Camuffo et al. [Camuffo et al., 2026] develop a G-theory-grounded protocol for LLM annotation in strategy research with five variance sources and a sampling-budget specification, the closest framework in the variance-decomposition literature we surveyed. TEE extends this with formal interaction-term estimation (item$\times$judge contributes 16.7% of $\mathrm{Var}(\hat{\theta})$ in safety and the per-observation share ranges 13 - 44% across the demonstrations here), D-study sample-size projections, the two-tier fixed/random classification, scoring method comparison, and Monte Carlo validation of D-study stability under assumption violations. Song et al. [Song et al., 2025] apply full G-theory to writing assessment with a person $\times$ rater $\times$ task design (run separately for one holistic and three analytic scoring dimensions) and D-study projections, a more complete psychometric application than the LLM-evaluation papers surveyed.

Haase et al. [Haase et al., 2026] decompose variance into three components (model choice, prompt, within-model) without interaction terms or D-study projections. Wang [Wang, 2025] uses an all-pairs paired-model method to decompose total noise into prediction noise (within-question) and data noise (between-question), without separately modeling prompt, temperature, or judge as facets in the decomposition. The NIST AI 800-3 report [Keller et al., 2026] proposes GLMMs for benchmark evaluation, citing G-theory as a parallel framing but without operationalizing D-study sample-size projections. Miller [Miller, 2024] applies the law of total variance to partition between-question and within-question variation, a two-component decomposition that TEE subsumes.

D G-Theory Connection and EMS Tables

Sections 2 and 3 state the correspondence in the main text. TEE is an application of generalizability theory [G-theory; Cronbach et al., 1972, Brennan, 2001] to LLM evaluation. Table 5 provides the mapping.

Table 5: Mapping between classical generalizability theory (G-theory) and TEE terminology. TEE applies G-theory's variance decomposition and decision-study framework to LLM evaluation pipelines, treating items as the object of measurement and prompt variants, judges, and replications as measurement facets.

G-Theory	**TEE**
Object of measurement	Item i
Facet (rater)	Model/judge m
Facet (occasion)	Prompt variant v
Fixed facet	Temperature h
Universe score μ_p	$\theta_{hm} = \mu + \tau_h + \lambda_m$
G-study	Factorial data collection
D-study	Pipeline optimization
σ^2_p (object variance)	σ^2_α (item heterogeneity)
σ^2_r (rater variance)	σ^2_λ (model sensitivity)
σ^2_{pr} (object $\times$ rater)	$\sigma^2_{\alpha\lambda}$ (item $\times$ model)

Table 6 gives the expected mean squares for the balanced mixed design.

Table 6: Expected mean squares for the balanced mixed design with C categories, $n = N/C$ items per category, V prompt variants, and R replications, conditioning on model m and temperature h. Temperature interaction terms ($\sigma^2_{\alpha\tau}$, $\sigma^2_{\phi\tau}$) are confounded with item and prompt main effects at a single temperature level. A multi-temperature design adds rows for Temperature ($H - 1$ df), Item$\times$Temperature, and Prompt$\times$Temperature.

Source	**df**	**EMS**
Category (κ)	$C - 1$	$\sigma^2_\rho + R\sigma^2_\epsilon + R\sigma^2_{\alpha\phi} + VR(\sigma^2_\delta + \sigma^2_{\alpha\tau}) + nVR\sigma^2_\kappa$
Item w/in cat. (δ)	$C(n-1)$	$\sigma^2_\rho + R\sigma^2_\epsilon + R\sigma^2_{\alpha\phi} + VR(\sigma^2_\delta + \sigma^2_{\alpha\tau})$
Prompt (ϕ)	$V - 1$	$\sigma^2_\rho + R\sigma^2_\epsilon + R\sigma^2_{\alpha\phi} + NR(\sigma^2_\phi + \sigma^2_{\phi\tau})$
Item $\times$ Prompt ($\alpha\phi$)	$(N-1)(V-1)$	$\sigma^2_\rho + R\sigma^2_\epsilon + R\sigma^2_{\alpha\phi}$
Residual (ρ)	$NV(R-1)$	σ^2_ρ

Solving the EMS equations for the variance of the grand mean $\bar{Y}_{\cdot\cdot h}$ yields the D-study formulas above. Model-related interactions ($\sigma^2_{\alpha\lambda}$, $\sigma^2_{\phi\lambda}$) enter under the multi-model extension (D-study Case 3 above) and are not shown in this single-model EMS table.

Absolute versus relative error. The D-study formulas above give the *absolute* error variance, appropriate for threshold decisions ("does this model exceed 80% safety?"). For *relative* decisions (comparing two models or ranking items), shared main effects such as prompt and judge shift all items equally and cancel when the compared systems are evaluated under the same prompt and judge levels. Interactions involving the compared object, such as item$\times$model or prompt$\times$model, remain. Relative-decision SEs should therefore use the covariance implied by the paired design. Applying the absolute formula mechanically will overstate some shared sources and understate object-specific interactions. This maps onto G-theory's distinction between the Φ coefficient (absolute, for threshold decisions) and the generalizability coefficient (relative, for rankings) [Brennan, 2001]. Chatbot Arena rankings call for the relative formula. AILuminate grades against a safety threshold call for the absolute formula.

E Monte Carlo Simulations

Eight simulation studies test the reliability of the variance decomposition and D-study projections. All use $N_{\text{sim}} = 1,000$ simulations except the scoring method recovery study (Section E.3, $N_{\text{sim}} = 200$). All use REML estimation via `lmer` with the `bobyqa` optimizer and seeds set as $42 + s$ for simulation index $s = 1, \ldots, N_{\text{sim}}$. Convergence rates exceed 99%. Figure 7 shows the REML estimator convergence properties.

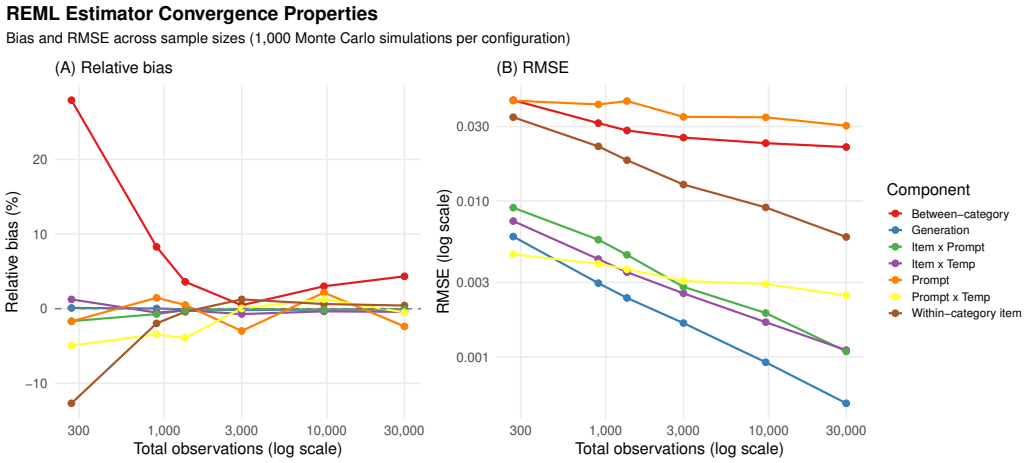


Figure 7: REML estimator convergence. (a) Bias falls below 5% for all components once total observations exceed 1,000. (b) RMSE by component across sample sizes, showing monotonic decline. Interaction components ($\sigma^2_{\alpha\phi}$, $\sigma^2_{\alpha\lambda}$) require larger samples for precise estimation than main effects. 1,000 simulations per configuration.

E.1 Robustness Studies

Four studies probe how the variance decomposition and D-study projections degrade under specific assumption violations, and the framework holds up across all four.

Three-way item×variant×temperature interactions barely move the decomposition. Sweeping their magnitude from 0 to 100% of the largest two-way component ($N = 50$, $C = 5$, $V = 4$, $H = 3$, $R = 5$) shifts Kendall's τ on the D-study ranking only from 0.895 to 0.877 at maximum violation, and non-residual components show under 5% relative bias across all levels.

Per-stratum REML recovers heteroscedastic residuals to within 0.3% relative bias. With temperature-specific residuals at $\sigma^2_{\epsilon,0} = 0.005$, $\sigma^2_{\epsilon,0.7} = 0.04$, and $\sigma^2_{\epsilon,1.0} = 0.08$, fitting a separate mixed-effects model per temperature via `lme4::lmer` recovers each within 0.3% bias. A single homoscedastic `lmer` fit pools to a weighted average instead, and the `glmmTMB` dispersion model substantially overestimates residual variance. Non-residual components remain unbiased under all three estimators.

$\hat{\sigma}^2_\phi$ is imprecise at small V, so we recommend $V \geq 3$. We swept $V \in \{2, 3, 4, 5, 7, 10\}$ and $\sigma^2_\phi \in \{0, 0.04, 0.10\}$. At $\sigma^2_\phi = 0.10$, RMSE falls from 0.156 at $V = 2$ to 0.105 at $V = 3$ and 0.049 at $V = 10$. For both nonzero values of σ^2_ϕ, relative bias is 3.8 - 5.3% at $V = 2$ and 1.0 - 4.5% at $V \geq 3$ (Figure 8). Each of these bias estimates is within two Monte Carlo SEs of zero, so the simulation does not show that bias falls as V grows. A knife-edge test shows directional accuracy collapses to near chance when prompts and replications contribute equally, but recovers when one source clearly dominates.

D-study projections transfer imperfectly between model profiles but identify the right top intervention most of the time. Across two profiles (a low-noise frontier model and a high-noise cheaper model), component estimates carry 55 - 59% relative error when frontier components project to the cheaper profile, and 127 - 154% in the reverse direction, but the projected top intervention is nonetheless incorrect only 9 - 13% of the time.

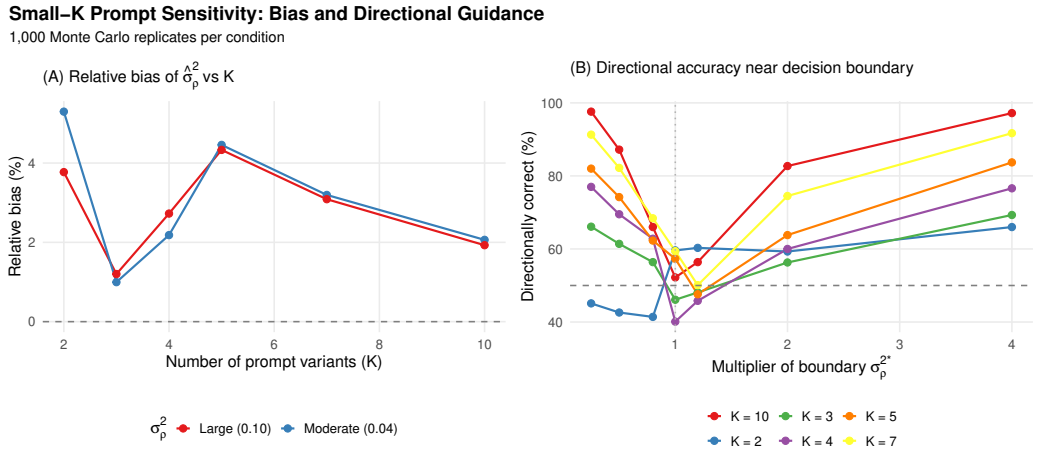


Figure 8: Small-V prompt sensitivity estimation. (a) Relative bias of $\hat{\sigma}^2_\phi$ versus number of prompt variants V. Bias is 3.8 - 5.3% at $V = 2$ and 1.0 - 4.5% at $V \geq 3$, within two Monte Carlo SEs of zero throughout. The recommendation of $V \geq 3$ rests on RMSE (see text). (b) Directional accuracy of the D-study recommendation (add prompts vs. add replications) near the knife-edge boundary where both interventions yield equal gains. Accuracy drops to near chance at the boundary. Away from it, accuracy rises with V: within this sweep it peaks at 66% at $V = 2$ and exceeds 90% only at $V \geq 7$. 1,000 replicates per condition. Axis label uses K for prompt variants (simulation convention). $K = V$ in main text notation.

E.2 D-Study Projection Validation Under Misspecification

We stress-test D-study projections under five scenarios. Scenario 1 is a correctly specified baseline. The other four perturb the DGP along a different axis each: correlated random effects ($\alpha = 2$), non-exchangeable prompt clusters (ratio of 8 between cluster variances), heavy-tailed score residuals (Student-t with df $= 5$), and heterogeneous category $\times$ variant variance (a $0.25\times$ vs. $4\times$ ratio across categories). Each scenario draws 1,000 Monte Carlo replicates from its DGP. We fit the standard TEE model and compare projected variance components against the truth. The relative bias in the projected variance ranges from -7.2% (heavy-tailed) to -0.9% (baseline), with the other three between -2.3% and -1.6% (Table 7). All five scenarios produce $|\text{bias}| \leq 8\%$, supporting the framework under the assumption violations tested.

Table 7: D-study projection bias under five scenarios: one correctly specified baseline and four misspecification conditions (correlated random effects, non-exchangeable prompts, heavy-tailed scores, heterogeneous category interactions). All scenarios produce $|\text{bias}| \leq 8\%$, confirming that D-study projections are stable under the assumption violations tested. $N_{\text{sim}} = 1,000$.

Scenario	Key parameter	Rel. bias (%)
1. Correct specification	-	-0.9
2. Correlated RE	$\alpha = 2$	-2.0
3. Non-exchangeable prompts	ratio $= 8$	-1.6
4. Non-normal scores	df $= 5$	-7.2
5. Heterogeneous category $\times$ variant	$0.25/4\times$	-2.3

E.3 Scoring Method Recovery Under Judge Heterogeneity

The choice of scoring method is another important design decision in LLM evaluation. Because the TEE decomposition takes observed scores as given and partitions their variance, it does not directly diagnose problems in the scoring function itself; if a scoring format introduces systematic distortions, a D-study based on those scores can point toward the wrong design changes.

In practice, three well-known LLM judge behaviors tend to degrade absolute rating scales more than pairwise comparisons, largely for the same structural reason:

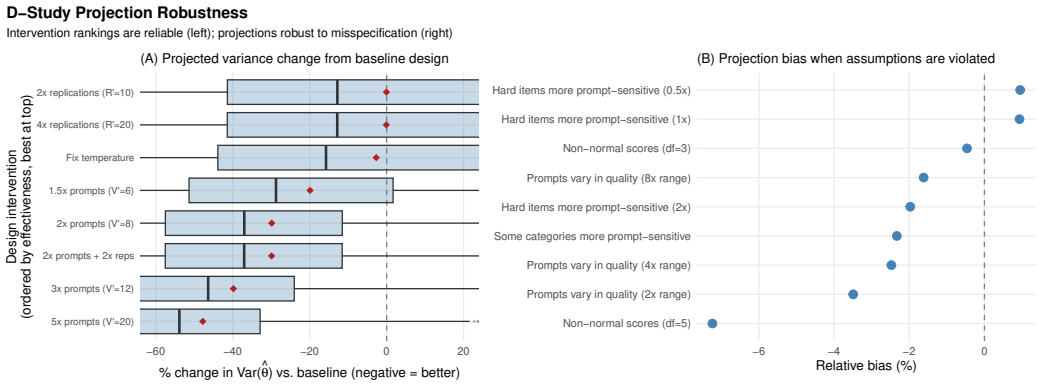


Figure 9: D-study projection validation. (a) Projected % change in variance for each intervention (1,000 datasets). Red diamonds: true change. (b) Projection accuracy under assumption violations, with bias $\leq 8\%$. Details in Section E.2.

1. *Central tendency compression*, where judges cluster ratings near the middle of the scale [Sahoo et al., 2025] and lose resolution at the tails. In a pairwise comparison, both items appear in the same context window, so compression affects both responses similarly and preserves their relative order.
2. *Low self-consistency*, where repeated calls assign different ratings to the same item (with intra-rater agreement as low as $\alpha \approx 0.27$, Haldar and Hockenmaier 2025). In pairwise comparisons, call-level anchoring noise applies to both responses simultaneously and cancels out in the difference.
3. *Scale-use heterogeneity* (analogous to differential item functioning across raters, Holland and Wainer 1993), where different judges map perceived quality onto numerical scales differently. Pairwise comparisons are unaffected by linear or multiplicative differences in scale use because the rescaling preserves the sign of the difference between two items, even though nonlinear distortions can still matter.

Because calibrating a common numerical scale across diverse LLM judges is difficult, fitting pairwise preferences with a Bradley-Terry (BT) model avoids the scale-calibration problem—which is why benchmarks like Chatbot Arena [Chiang et al., 2024] rely on pairwise comparisons. That said, whereas Likert scales can suffer from reduced discriminability similar to survey satisficing [Krosnick, 1991], pairwise comparisons require more API calls per item set and introduce position bias.

To compare these formats under controlled conditions, we run a simulation that generates items with known latent quality scores, applies Likert and Bradley-Terry (BT) scoring under identical per-observation noise [Peysakhovich et al., 2015], and varies five Likert distortions: central tendency compression (κ), per-observation anchoring noise (σ_{anchor}), scale-use heterogeneity (σ_{slope}), nonlinear scale distortion (γ), and discretization loss (number of scale points). Giving both scoring methods the same per-observation noise isolates the structural effect of the comparison format from differences in noise levels. We run four simulation experiments across these parameters: (1) ideal conditions, (2) a 3-way sweep of $\kappa \times \sigma_{\text{anchor}} \times \sigma_{\text{slope}}$ (64 conditions), (3) $\kappa \times \gamma \times$ scale points (48 conditions), and (4) a full 5-way grid (768 conditions).

As expected, under ideal conditions Likert matches full-budget BT ($\tau \approx 0.835$), but when any of the five scale distortions are present, Likert ranking recovery drops while BT remains stable (Figure 10). Central tendency compression has the largest impact: as κ decreases from 1.0 to 0.3, Likert τ falls from 0.46 to 0.43 while BT stays flat at 0.50, and coarsening the scale to 3 points further reduces Likert recovery to $\tau = 0.38$. The difference is largest at low prevalence, where binary classification drops from $\tau = 0.60$ at 50% prevalence to $\tau = 0.06$ at 0.05% prevalence while BT remains at $\tau \approx 0.73$. This low-prevalence regime is particularly relevant for safety benchmarks like AILuminate, where a 94% safe rate leaves relatively few unsafe cases.

These results suggest three practical guidelines: when judges are well calibrated and use finer scales (7+ points, $\kappa > 0.7$), Likert and pairwise scoring recover rankings similarly and Likert requires roughly one-third as many calls; when judge calibration is uncertain or the scale is coarse (3 -

5 points), pairwise comparisons offer much greater stability against compression, anchoring, and scale-use differences at roughly $3\times$ the cost; and for binary outcomes with low base rates ($< 5\%$), pairwise comparisons perform substantially better than binary classification. Of course, realizing the benefits of pairwise scoring requires controlling for position bias: in an earlier naive-prompt run we observed a 67% B-rate across the three judges, so our reference pipeline uses a CoT prompt, order counterbalancing, and a position covariate to keep residual position bias in check (SI Appendix, Section I).

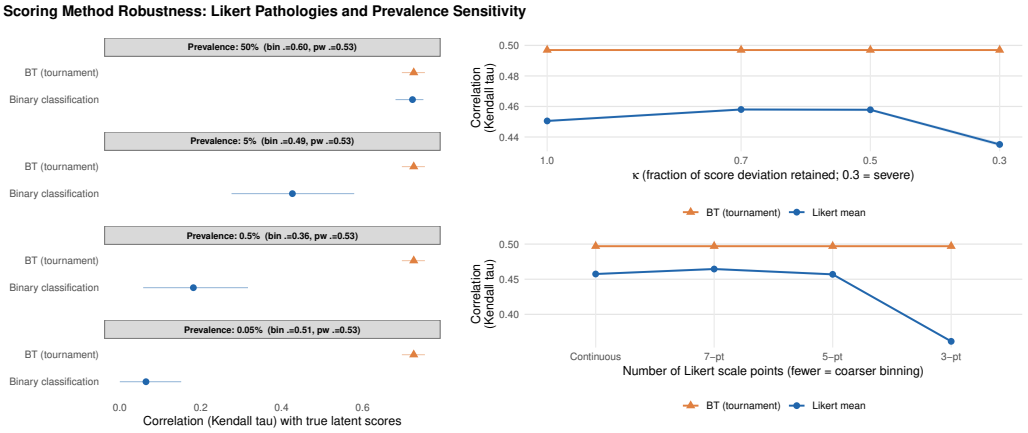


Figure 10: BT pairwise scoring is stable under Likert pathologies. Left: at low prevalence, binary classification collapses while BT remains stable. Upper right: central tendency compression (κ) degrades Likert ranking recovery, while BT is unaffected. Lower right: coarser Likert scales (fewer points) reduce recovery, while BT is invariant to discretization. The full simulation sweeps five pathologies (κ, σ_{anchor}, σ_{slope}, nonlinear γ, discretization). This figure shows the three with the largest effects. $N_{\text{sim}} = 200$.

E.4 Variance Underestimation When Pipeline Factors Are Omitted

The main text (Figure 1) shows that naive CIs fail when the fitted model omits random effects for pipeline factors that contribute variance. This section details the six scenarios behind that result and traces how coverage degrades as design-side components are omitted from the fitted model.

We set the pipeline components to comparable magnitudes in the DGP so that each scenario captures a visible share of the variance (Table 8). This setup illustrates the mechanism and is not calibrated to any benchmark. The full factorial uses $V = 4$ prompt variants, $H = 3$ temperatures, $M = 3$ judges, and $R = 8$ replications.

Table 8: DGP parameters for the variance underestimation simulation (`analysis/04g2_sim_underestimation_balanced.R`). Temperature and judge enter as fixed levels; every other component is a random-effect variance.

Component	Value	Notes
σ^2_α (item)	0.10	
σ^2_ϕ (prompt)	0.04	
τ (temperature levels)	$\{-0.20, 0, 0.20\}$	$\text{pop_var}(\tau) = 0.027$
λ (judge levels)	$\{-0.20, 0, 0.20\}$	$\text{pop_var}(\lambda) = 0.027$
$\sigma^2_{\alpha\phi}$ (item $\times$ prompt)	0.04	
$\sigma^2_{\alpha\tau}$ (item $\times$ temperature)	0.03	
$\sigma^2_{\alpha\lambda}$ (item $\times$ judge)	0.04	
$\sigma^2_{\phi\tau}$, $\sigma^2_{\phi\lambda}$, $\sigma^2_{\tau\lambda}$	0.02	design $\times$ design
σ^2_ϵ (cell-level)	0.04	constant across replicates within a cell
σ^2_ρ (replicate)	0.10	within-cell replicate noise

Six scenarios sample from this factorial and fit progressively richer `lmer` models (Table 9). Scenario A is the naive baseline. It samples one observation per item at a randomly chosen (prompt, temperature, judge) configuration and reports $\text{SE} = s/\sqrt{N}$. Scenarios B, C, and D then add prompt-variant,

judge, and temperature random effects in turn, sampling the corresponding factor in full. Scenario E adds the design-by-design interactions. Scenario F fits cell means with the item interactions as well, which separates the cell-level term σ_ϵ^2 from replicate noise.

Table 9: Scenarios A - F progressively enrich the fitted random-effects model. Each row indicates the factorial dimensions the scenario samples, the random effects it adds, the variance it then captures, and what remains uncaptured.

Scenario	Sampled	Adds	Captures	Misses
A (naive)	V=1, M=1, H=1, R=1	$s/\sqrt{N}$	-	all design-side
B	V=4, M=1, H=1, R=8	`(1|variant)`	σ_ϕ^2	τ, λ, interactions
C	V=4, M=3, H=1, R=8	+ `(1|judge)`	+ λ	τ, interactions
D	V=4, M=3, H=3, R=8	+ `(1|temp)`	+ τ	interactions
E	same data as D	+ `(1|variant:judge)`, `(1|variant:temp)`, `(1|temp:judge)`	+ design × design	item × design, σ_ϵ^2
F	cell means of D's data	+ `(1|item:variant)`, `(1|item:judge)`, `(1|item:temp)`	+ item × design, σ_ϵ^2	-

The point estimate in every scenario is the mean of the sampled observations. The SE is $\sqrt{\text{Var}(\hat{\beta}_0)}$ from the fitted `lmer`, except for A, which uses $s/\sqrt{N}$. A fit that exceeds a 300-second timeout falls back to the oracle SE computed from the true variance components under that scenario's model; this happened in 28 of the 1,000 Scenario E replicates at $N = 400$ and in no other cell. All six scenarios are evaluated at $N \in \{25, 50, 100, 200, 400\}$ with 1,000 Monte Carlo replicates per cell.

Coverage rises in steps across the six scenarios as each fit captures more of the design-side variance. Scenario A captures none of it. Its SE is the naive $s/\sqrt{N}$ and shrinks with sample size, but the bias from the single (prompt, temperature, judge) configuration stays constant, so coverage falls from 42% at $N = 25$ to 11% at $N = 400$. Scenario B adds prompt-variant random effects and reaches 56 - 64% coverage, which stays well below nominal because the judge and temperature offsets are still unmodeled. Scenario C adds judges and reaches 86 - 89%, limited by the remaining temperature offset. Scenarios D, E, and F capture every design-side component and cover at 99 - 100%, above nominal, with SEs 9 - 19% larger than the oracle SE. Because the DGP holds the three judges and three temperature levels fixed, they add no error to the estimate. Both the fitted and the oracle SE still count the judge and temperature variance, since those SEs generalize to new judges and temperatures. The fitted SEs are therefore 1.5 - 1.7 times the empirical standard deviation of the estimates. Section E.5 redraws the levels in every replicate. E and F have identical coverage at this design, but F also separates the cell-level term σ_ϵ^2 from replicate noise, which matters when σ_ϵ^2 is large.

E.5 Coverage When Judge and Temperature Levels Are Redrawn

Section E.4 holds the three judge levels and the three temperature levels fixed at $\{-0.20, 0, 0.20\}$, so Scenarios D - F over-cover. Here each replicate draws new judge and temperature levels from $N(0, 0.027)$, the variance of the fixed levels, and the estimand is the superpopulation mean μ. The DGP, scenarios, fitted models, seeds, and 1,000 replicates per cell are unchanged (`analysis/04g2_sim_underestimation_balanced.R --redraw-design`; comparison in `analysis/04g2b_redraw_comparison.R`).

Table 10 compares coverage under the two designs. With redrawn levels, Scenarios D, E, and F cover at 91 - 95%. Their median SE is 0.94 - 1.07 times the empirical standard deviation of the estimates, against 1.53 - 1.69 with fixed levels. Coverage stays a few points below nominal because the judge and temperature variances each rest on three levels, so the SE itself varies from replicate to replicate. Misses concentrate where the estimated SE is small. Pooled over N, coverage is 80 - 85% in the quarter of replicates with the smallest SEs and 99% in the quarter with the largest. Scenarios A - C stay well below nominal under both designs. Six Scenario E fits at $N = 400$ timed out and used the oracle SE. We report the median SE because one Scenario E fit at each of $N = 200$ and $N = 400$ returned a degenerate SE above 1 under each design.

E.6 Latent Item Ambiguity

This simulation builds an even harsher misspecification: per-item variance heterogeneity driven by a latent ambiguity score, combined with a sparse 3-way interaction concentrated in the ambiguous tail. The motivating concern is that earlier robustness simulations addressed additivity violations (Section E.1) and correlated random effects (Section E.2) in isolation. When both arise jointly from the same latent property of the item, do D-study projections still reflect the truth?

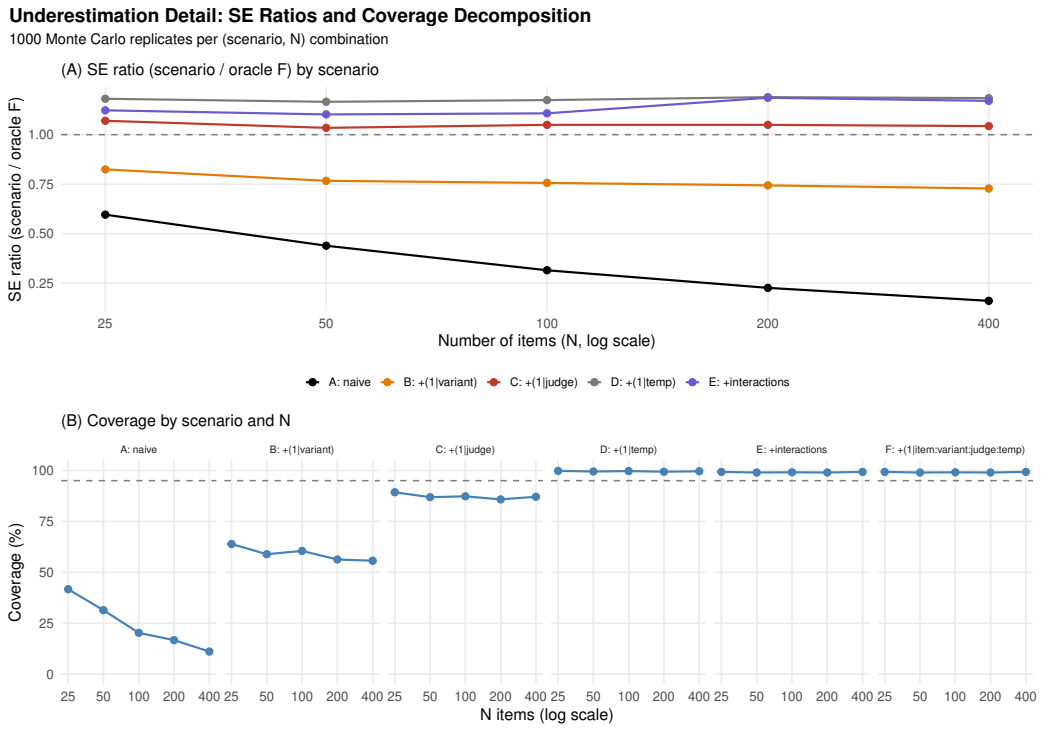


Figure 11: Naive 95% CIs lose coverage as sample size grows because the SE shrinks while the bias from omitted pipeline factors stays constant. Panel (a) plots each scenario's SE relative to the oracle SE of the full model, with the naive Scenario A about $6\times$ smaller at $N = 400$. Panel (b) plots 95% CI coverage versus N for the six scenarios, showing how each added random effect (prompt in B, judge in C, temperature in D, interactions in E and F) closes one more component of the omitted variance and lifts coverage toward nominal. $N \in \{25, 50, 100, 200, 400\}$; $N_{\text{sim}} = 1,000$.

The DGP draws a per-item ambiguity score $z_i \sim N(0, 1)$ and scales that item's item×prompt and item×judge variances by $1 + \gamma z_i^2$. Higher-z_i items therefore swing more under prompt and judge variation. Items with $z_i > 1$ (roughly 16% of the population) additionally carry a 3-way item×prompt×judge interaction with variance 0.01, so prompt and judge effects compound nonadditively for the most ambiguous items. We sweep γ over $\{0, 0.5, 1, 2\}$. The endpoints anchor the experiment: $\gamma = 0$ is correct specification, $\gamma = 2$ triples interaction variance one SD above the mean.

We sample a small factorial of $N = 30$ items across $C = 5$ categories, $V = 3$ prompts, $H = 2$ temperatures, $M = 3$ judges, and $R = 3$ replications, totaling 1,620 observations per fit. The TEE specification carries the two-way interactions only; the 3-way interaction and the latent heterogeneity remain outside the model.

CI coverage holds at 89 - 91% across all four γ settings, against the 95% nominal under such a small N. The worst component-level relative bias is below 9% and D-study intervention rankings agree perfectly with the truth (Spearman $\rho = 1.0$) across 1,000 simulations per setting. Breakdown by γ in Table 11.

This stability follows from the fact that `lmer` estimates *marginal* variance components (i.e., population averages across items rather than item-specific variances). When each item's item×prompt component is scaled by $1 + \gamma z_i^2$, the expected scaling across the population is $E[1 + \gamma z_i^2] = 1 + \gamma$, yielding a marginal item×prompt variance of $(1 + \gamma)\sigma^2_{\alpha\phi}$, which the estimator recovers. Meanwhile, because the sparse 3-way interaction is limited to a small subset of items, it is absorbed into the residual term and only slightly increases $\hat{\sigma}^2_\epsilon$, leaving the two-way estimates largely unaffected.

Untested scenarios include multimodal residuals (e.g., from a mixture of refusals and compliance), extreme base rates (above 0.9 or below 0.1), unbalanced designs with missing cells, post-hoc model specification search, and nonlinear temperature effects.

Table 10: Redrawing the judge and temperature levels in every replicate lowers Scenario D - F coverage from 99 - 100% to 91 - 95%. Cells give 95% CI coverage (%) by scenario, level design, and number of items N, with 1,000 replicates per cell (Monte Carlo SE at most 1.6 points).

		N				
Scenario	**Levels**	25	50	100	200	400
A	Fixed	41.7	31.4	20.3	16.7	11.1
A	Redrawn	43.2	31.8	20.2	14.4	11.6
B	Fixed	63.9	58.9	60.5	56.3	55.7
B	Redrawn	66.1	59.8	57.0	57.2	57.8
C	Fixed	89.3	86.9	87.3	85.8	87.1
C	Redrawn	84.3	81.8	80.9	81.8	79.4
D	Fixed	99.8	99.5	99.7	99.4	99.6
D	Redrawn	94.9	94.1	92.6	93.6	92.9
E	Fixed	99.3	99.0	99.1	99.0	99.3
E	Redrawn	93.9	91.3	90.7	91.8	90.8
F	Fixed	99.3	99.0	99.1	99.0	99.3
F	Redrawn	93.6	91.1	90.6	91.6	90.9

Table 11: TEE stability under latent item ambiguity. γ controls the strength of ambiguity-driven dependence between item difficulty and interaction variances, with sparse 3-way interactions for ambiguous items ($z_i > 1$). CI coverage does not degrade, component bias stays below 9%, and D-study intervention rankings are preserved exactly across all ambiguity levels. $N_{\text{sim}} = 1,000$.

γ	Label	Coverage (%)	Max |rel. bias| (%)	D-study ρ
0	Baseline	90.5	8.8	1.00
0.5	Mild	88.7	6.7	1.00
1	Moderate	90.3	3.4	1.00
2	Strong	91.2	6.0	1.00

F Multi-Turn and Multi-Agent Extensions

F.1 Multi-Turn Extension

The single-turn DGP extends to multi-turn evaluation when each turn's score depends on the prior conversation. Adding a turn index t and a turn-conditional mean $\mu_t(\mathbf{h}_{<t})$ for the expected score given history $\mathbf{h}_{<t}$ yields[2]

$$Y^{(r)}_{t,ivhm} = \mu_t(\mathbf{h}_{<t}) + \alpha_i + \phi_v + \tau_h + (\alpha\phi)_{iv} + \epsilon_{t,ivhm} + \rho^{(r)}_{t,ivhm}. \tag{11}$$

The pipeline-side variance components keep their single-turn interpretations because they describe how the evaluation is computed, not how the conversation evolves. The multi-turn structure enters through $\mu_t(\mathbf{h}_{<t})$, which drifts as the model conditions on accumulating context.

Safety evaluation often asks at what turn the model first crosses a safety threshold θ^*, which maps onto a survival framing with hazard rate

$$\eta(t) \approx \Phi\left(\frac{\mu_t(\mathbf{h}_{<t}) - \theta^*}{\sqrt{\text{TEE}_t}}\right). \tag{12}$$

This Gaussian-tail approximation treats $\eta(t)$ as the probability that a Gaussian score with mean $\mu_t(\mathbf{h}_{<t})$ and variance TEE_t exceeds the threshold θ^*. Larger pipeline-level variance therefore means a higher hazard at any threshold above $\mu_t(\mathbf{h}_{<t})$, so the same components that distort single-turn CIs also raise multi-turn safety risk. Full integration with Cox proportional hazards models [Cox, 1972] for time-varying covariates is beyond the current scope. Existing empirical work supports the framing. Weng et al. [2025] document harmfulness scores increasing across turns under foot-in-the-door attacks, consistent with $\mu_t(\mathbf{h}_{<t})$ drifting upward as context accumulates. Li et al. [2025b] fit hazard models to multi-turn LLM interactions and find that hazard rates depend on temperature, supporting the inclusion of τ_h in the multi-turn DGP.

F.2 Multi-Agent Extension

Suppose instead that a pipeline runs A agents in parallel on the same item, yielding $Y_1, \ldots, Y_A$, and reports their average $\bar{Y}$. The variance of the mean is

$$\text{Var}(\bar{Y}) = \frac{1}{A^2}\left[\sum_{a=1}^{A} \text{TEE}_a + \sum_{a \neq a'} \text{Cov}(Y_a, Y_{a'})\right]. \tag{13}$$

Averaging reduces each agent's TEE component at a rate of $1/A$. However, the off-diagonal covariance terms do not disappear when agents share training data or scoring rubrics, and correlated errors across agents reduce the gains from ensembling. A multi-judge panel scoring a single response is a special case of this setup, with judges as the parallel agents.

When agents are chained sequentially so that agent A takes its input from agent $A-1$, the variance of the final score decomposes via the law of total variance:

$$\text{Var}(Y_A) = \text{E}[\text{Var}(Y_A \mid Y_{A-1})] + \text{Var}(\text{E}[Y_A \mid Y_{A-1}]). \tag{14}$$

Here, the first term reflects the variance introduced by agent A conditional on its input, while the second term captures the variance propagated from upstream steps. Applying this recursion across the chain splits overall variance into each agent's local contribution and the variance that it inherits from upstream [Duan et al., 2025]. Consistent with this structure, Kim et al. [2025] find $17.2\times$ error amplification for independent agent topologies compared with $4.4\times$ for centralized ones, suggesting that agent architecture can affect pipeline variance as much as judge or prompt selection.

[2] In this subsection $\eta(t)$ denotes the hazard rate and $\mathbf{h}_{<t}$ denotes conversation history, distinct from the subscript h used for temperature in the main DGP.

G GLMM Robustness Check

Binary outcomes in the main text use a linear probability model [LPM, Angrist and Pischke, 2009]. The LPM residuals for binary data are heteroscedastic, which raises the question of whether the variance shares that we report in the main text are artifacts of the link function. To check, we refit the three binary decompositions with a logit link via `lme4::glmer(family = binomial)` [Bates et al., 2015] and compare the GLMM shares with the LPM shares used in the main text.

Table 12: LPM vs. GLMM Var($\hat{\theta}$) decomposition for the three binary benchmarks (pairwise ideology, safety classification, MMLU). LPM shares are on the probability scale; GLMM shares are on the logit scale, so absolute SE values are not directly comparable, but component rankings are. For the GLMM, the level-1 residual is fixed at $\pi^2/3$ following the logistic-link convention [Nakagawa and Schielzeth, 2010]. All three fits include the cell-level random effect. For MMLU the model layer is the system under test (SUT) rather than a judge. The SUT row plays the same role as "judge model" in the other two benchmarks. Spearman rank correlations between LPM and GLMM shares are $\rho = 0.99$ (pairwise), 0.91 (safety), and 0.95 (MMLU), confirming that the dominant components are not LPM artifacts.

	Pairwise (%)		**Safety (%)**		**MMLU (%)**	
Component	**LPM**	**GLMM**	**LPM**	**GLMM**	**LPM**	**GLMM**
Judge / SUT model (σ^2_λ, sensitivity)	68.4	50.9	43.8	70.7	6.7	7.6
Within-category item (σ^2_δ)	7.6	13.7	20.8	11.4	72.2	71.4
Prompt (σ^2_ϕ)	0.4	0.4	0.0	0.0	0.4	1.3
Item $\times$ judge / SUT ($\sigma^2_{\alpha\lambda}$)	1.9	1.9	16.7	4.3	10.5	5.2
Prompt $\times$ judge / SUT ($\sigma^2_{\phi\lambda}$)	0.1	0.0	12.4	7.9	1.3	4.4
Position (σ^2_π, sensitivity)	20.9	32.7	-	-	-	-
Between-category (σ^2_κ)	0.2	0.1	4.9	4.7	7.7	9.6
Other ($\leq$1%)	0.5	0.3	1.4	1.0	1.2	0.6

The decomposition holds up under the logit link in all three benchmarks. The same component dominates in each case (judge model on pairwise and safety, within-category item heterogeneity on MMLU), and rank correlations are near-perfect ($\rho = 0.99$ pairwise, 0.91 safety, 0.95 MMLU). For safety, the judge model share rises from 43.8% under the LPM to 70.7% under the GLMM. This is because the safe rate is 94%, and at that rate a small difference between judges on the probability scale becomes a large difference on the logit scale. For MMLU, where 84.5% of answers are correct, item heterogeneity is the largest component under both links (72.2% LPM, 71.4% GLMM), and the three prompt terms are small on both scales (2.7% and 6.1%). We keep the LPM in the main text because we can read its components directly as percent of variance on the probability scale. We use the GLMM here only to check robustness to the link function.

H Ideology Scoring Comparison: Likert vs Pairwise

When we apply Likert and pairwise scoring to the same domain under a matched protocol, they produce qualitatively different variance profiles, which have different implications for optimizing the pipeline.[3] Four frontier LLMs (Claude Opus 4.5, GPT-5.1, DeepSeek Chat v3.1, Grok 4.1 Fast) generated free-text responses to prompts drawn from five ideology dimensions. A common judge panel and prompt protocol then scored each response under either a 1 - 5 Likert scale or a forced pairwise A/B comparison.

Both pipelines use a chain-of-thought (CoT) prompt with a content-markers reasoning step, an explicit instruction not to let presentation order or prompt wording bias the decision, and a structured JSON output. The judges are `openai/gpt-oss-120b`, `google/gemini-2.0-flash-001`, and `deepseek/deepseek-chat-v3.1`, with each cell replicated 3 times at 3 temperatures (0, 0.7, 1.0) across 5 paraphrased prompt variants. The Likert pipeline makes 150 items $\times$ 5 variants $\times$ 3 temperatures $\times$ 3 judges $\times$ 3 replications $= 20,250$ calls. The pairwise pipeline judges the same 150 item-pairs in both orderings (listed + swapped), and recodes the raw A/B outcome to `model_a_wins` using the order randomized at item creation (`true_order`) to remove any residual position bias. The pairwise design is also 20,250 calls.

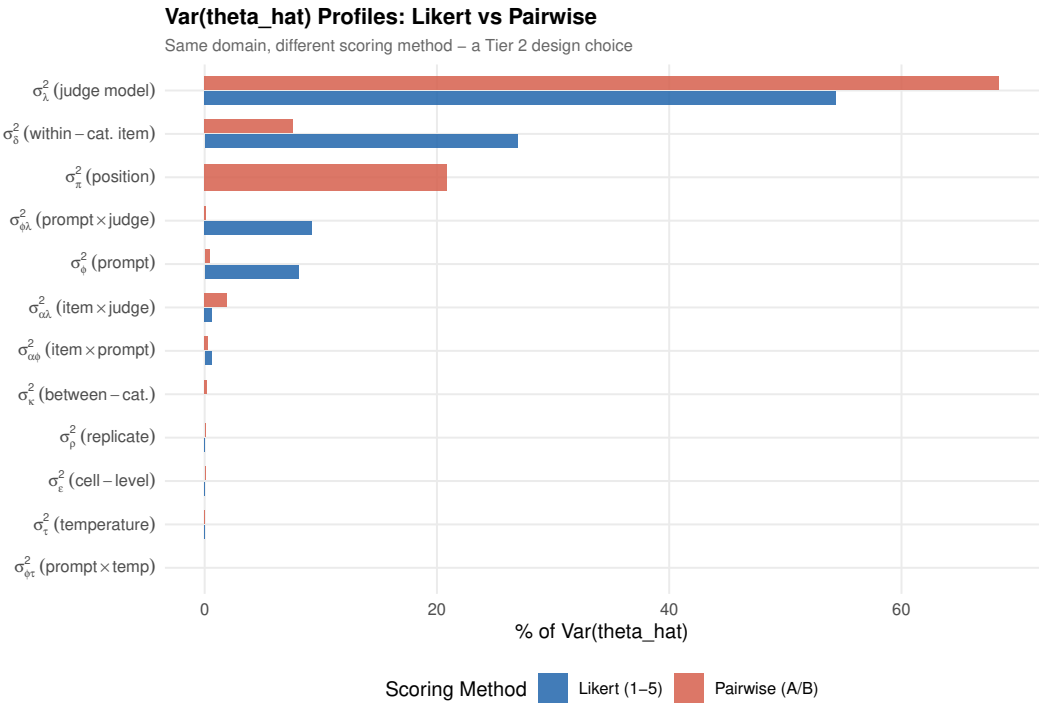


Figure 12: Judge-model variance dominates both ideology pipelines, but the secondary rankings differ. After judges, Likert is dominated by item heterogeneity and pairwise by position bias, a factor with no Likert analogue. Operational design: N=150, V=5, M=3, H=3, R=3 (pairwise also crosses P=2 position orderings).

In both decompositions, judge-model design sensitivity accounts for the largest share of variance (54.4% under Likert and 68.4% under pairwise), though the remaining components differ noticeably

[3] An initial pairwise run used a naive prompt (forced A/B, no CoT) and an older judge set (GPT-4o, Gemini 2.0 Flash, Claude Haiku 4.5). That run was dominated by recency bias (Haiku selected the second response 91% of the time regardless of content, GPT-4o 70%, Gemini 39%), so the raw decomposition attributed 20.7% of variance to a "judge main effect" that was really position bias. Recoding via `true_order` reduces this spurious main effect to 0.2%, but the noise from the position bias moves into the item×judge interaction, which rises to 54.7%. This happens because Haiku's near-random content signal becomes item-level disagreement after recoding. The naive pairwise data is archived. The reference pairwise pipeline described above uses the CoT protocol and a judge set chosen for lower recency bias. We also re-ran the Likert pipeline with the same CoT protocol and judges, so the two pipelines now differ only in the rating format. The earlier naive-Likert run is archived.

between the two formats. Under Likert scoring, the next largest shares come from within-category item heterogeneity (26.9%), prompt×judge interaction (9.3%), and the prompt main effect (8.1%). Under pairwise scoring, by contrast, the position factor (recoded from listed and swapped orderings via `true_order`) accounts for 20.9% of the variance on its own, while within-category item heterogeneity drops to 7.6% and item×judge falls to 1.9%. In short, the two scoring formats face different sources of error: pairwise comparisons avoid scale compression, anchoring noise, and scale-use heterogeneity [Sahoo et al., 2025, Haldar and Hockenmaier, 2025, Holland and Wainer, 1993], whereas Likert scoring is not affected by presentation-order bias. As shown in Section E.3, pairwise BT remains stable under those Likert scale distortions, and it does not break down at low prevalence the way binary scoring does ($\tau = 0.06$ vs. $\tau = 0.73$ for BT; Figure 10).

Despite the different variance profiles, the D-studies for both formats point to judge variance as the key lever. Adding items, prompts, or replications buys little marginal precision once their respective terms are well-averaged at $N = 150$, $V = 5$, $R = 3$. Full per-method D-study projections appear in Figures 13 and 14.

A separate sanity check explains why pairwise replicate noise is so large. A Claude Sonnet 4.5 judge with a CoT prompt that *allowed* a TIE option was run on all 150 pairwise items in both orderings. Inter-order consistency was 100% (every item's listed and swapped judgments either inverted cleanly or both returned TIE), confirming zero position bias. Sonnet returned TIE on 84.7% of items, so only 23 of 150 pairs carried a clear ideological preference. On those 23 distinguishable items, directional agreement with the three CoT pairwise judges was high (DeepSeek 100%, Gemini 91%, GPT-OSS 87%). The 85% TIE rate explains why so much pairwise variance lands in σ^2_ρ and σ^2_ϵ. When judges must pick a side on pairs that are genuinely similar, their choices are random with respect to content, and this noise accumulates within and across cells. The Likert pipeline avoids this problem because the 1 - 5 scale allows a centrist rating for ambiguous content.

I Pilot Study Details

TEE is most useful when researchers can plan a full factorial pipeline from cheap pilot data. This section shows that variance estimates from a pilot recover the qualitative D-study conclusions of a full run, so a small pilot is enough to choose a design, which the full pipeline then confirms quantitatively.

We use the Likert ideology demonstration (150 items, 5 prompt variants, 3 temperatures, 3 judges, 3 replications; 20,250 calls) as the testbed. A 30-item, 3-variant subset of the full-run data stands in for the pilot, and we compare its D-study projections to the full-run results. Because the pilot is a subsample of the same items, this test asks whether small-sample variance estimates translate into accurate D-study projections. It does not test how a pilot drawn from completely different items would generalize.

The pilot recovers the qualitative D-study conclusions. Both pilot and full run identify judge-model design sensitivity and within-category item heterogeneity as the two dominant contributors to $\mathrm{Var}(\hat{\theta})$. The shares differ between the two designs because the divisors differ. The pilot gives 42.4% to judge model and 48.6% to within-category item, and the full run gives 54.4% and 26.9%. The larger N in the full run shrinks σ^2_δ/N faster than σ^2_λ/M at fixed M, which raises the judge-model share. A pilot-scale run ($N = 30$, $V = 3$, $H = 3$, $M = 3$, $R = 3$; 2,430 calls) thus recovers the qualitative guidance that a 20,250-call full run confirms quantitatively. Practitioners can treat D-study projections as directional guidance accurate to within 20 - 30% at pilot scale, with the same top two components across designs.

Running a pilot and plugging the estimated components into standard errors yields valid confidence intervals. The underlying variance itself is unchanged. To realize the precision gains from averaging over multiple prompts and judges, researchers need to deploy the full factorial pipeline. Section M turns these recommendations into a three-tier reporting checklist.

Table 13: Variance component estimates for the Likert ideology demonstration (CoT protocol): a 30-item, 3-variant pilot subset ($N = 30$, $V = 3$, $M = 3$, $H = 3$, $R = 3$; 2,430 calls) compared with the 150-item full run ($N = 150$, $V = 5$, $R = 3$; 20,250 calls). The pilot correctly identifies the two dominant components (within-category item and judge-model design sensitivity) and recovers their rank ordering. Quantitative shares shift between pilot and full because the divisors differ across designs. The pilot pools σ^2_ρ and σ^2_ϵ into a single "replicate noise" row because the small N does not separately identify the cell-level term. Shares of mean variance $\mathrm{Var}(\hat{\theta})$ at the indicated pilot and full designs.

Component	**Pilot (% Var($\hat{\theta}$))**	**Full (% Var($\hat{\theta}$))**
Judge model (σ^2_λ, sensitivity)	42.4	54.4
Within-category item (σ^2_δ)	48.6	26.9
Prompt × judge ($\sigma^2_{\phi\lambda}$)	5.4	9.3
Prompt (σ^2_ϕ)	-	8.1
Item × judge ($\sigma^2_{\alpha\lambda}$)	2.2	0.6
Item × prompt ($\sigma^2_{\alpha\phi}$)	1.3	0.6
Replicate noise / cell-level ($\sigma^2_\rho + \sigma^2_\epsilon$)	0.2	0.1
All others	<0.1	<0.1

An earlier naive-prompt pairwise run (forced A/B, no chain-of-thought, judges GPT-4o / Gemini 2.0 Flash / Claude Haiku 4.5) was superseded because position bias dominated the outcomes. Judges selected the second response 67% of the time overall and 91% for Haiku, and the order-counterbalancing already in place (a randomized `true_order` at item creation) did not rescue the result, since the judge outputs track position and not model preference. Recoding the outcome to `model_a_wins` via `true_order` removes the bias from the mean (all three judges converge to ∼45% A-wins), but because Haiku chose B 91% of the time, its recoded outputs mostly reflect the random order flips and not content. The reference pairwise pipeline (Section H) therefore switches to a CoT prompt with bias mitigation and a judge set chosen for low recency bias (DeepSeek Chat v3.1, Gemini 2.0 Flash, GPT-OSS 120B). The naive pairwise data is archived in the replication materials.

We use D-study projections to turn each variance profile into design recommendations. Figures 13 and 14 report the projected percentage change in $\mathrm{Var}(\hat{\theta})$ for each single-factor intervention under the two scoring formats. The variance shares that those projections are based on appear in Figure 12.

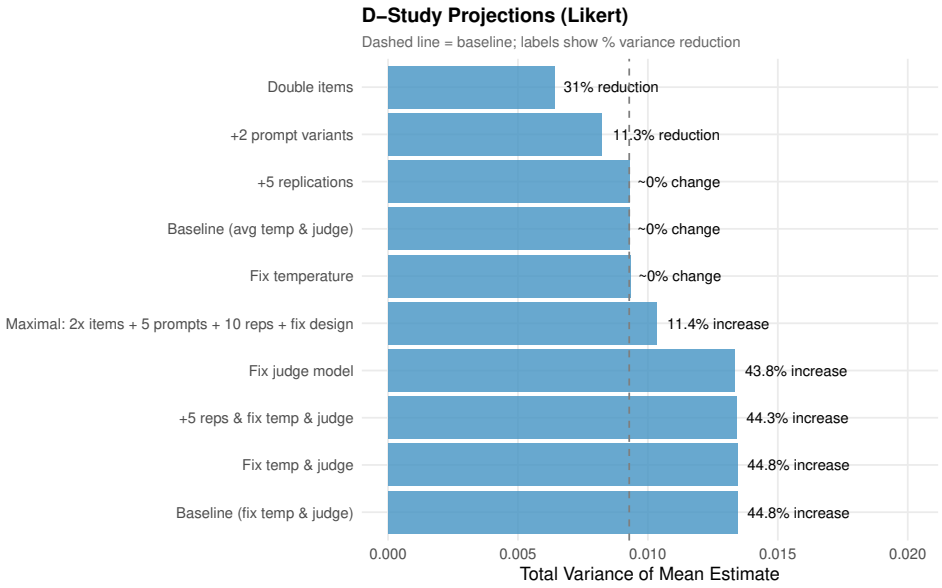


Figure 13: Judge averaging is the dominant lever in Likert ideology scoring. Bars show the projected percentage change in $\mathrm{Var}(\hat{\theta})$ for each single-factor intervention. Committing to a single judge inflates variance by 43.8% because σ^2_λ (54.4% of $\mathrm{Var}(\hat{\theta})$ at $M = 3$) loses the $1/M$ averaging benefit. Doubling items reduces variance by 31% by halving the within-category item term (σ^2_δ, 26.9% of $\mathrm{Var}(\hat{\theta})$). Adding 2 prompt variants buys 11.3%. Replications contribute negligibly. Same factorial design as Figure 12.

Replicate noise is also markedly heterogeneous within each domain. Across five ideology dimensions, $\hat{\sigma}^2_\rho$ spans a 1.6- to 1.8-fold range (Figure 15), so some dimensions produce more variable LLM responses across repeated calls than others. The same heterogeneity is more pronounced in safety classification, where per-category replicate noise spans more than an order of magnitude (Figure 16), from $\hat{\sigma}^2_\rho = 0.001$ for specialized advice to $\hat{\sigma}^2_\rho = 0.033$ for sex crimes. This heterogeneity motivates the heteroscedastic extension (Section C.7).

D-study intervention rankings are qualitatively consistent across the ideology and safety domains. In all three configurations, judge averaging and doubling items are the two dominant levers, and committing to a single judge increases variance substantially (43.8% Likert, 40.6% pairwise, 106% safety) because σ^2_λ loses its $1/M$ averaging benefit. Adding two prompt variants buys 11.3% (Likert) and 2.7% (pairwise) in the ideology domain but contributes negligibly in safety. Additional replications contribute negligibly everywhere because $\sigma^2_\rho/(N'V'H'M'R')$ is already a tiny fraction of $\mathrm{Var}(\hat{\theta})$ at $R \geq 3$. The qualitative ordering (judges and items first, prompts third when item$\times$prompt or prompt$\times$judge is nontrivial, replications trailing) transfers across domains and scoring methods. The magnitudes differ because the variance profiles differ.

I.1 Propaganda: Variance Predicts Misclassification

Evaluating whether an LLM response favors one country's interests over another requires condensing multiple considerations—such as tone, framing, and what information is emphasized or left out—into a single label. Because individual raters weigh these considerations differently, any single rater is noisy on their own, which is why we use a 9-rater human consensus panel as our reference benchmark (composite reliability ICC($k = 9$) $= 0.85$). Specifically, we draw 50 prompts from Waight et al. [2026], where 9 human coders classified each LLM response according to which country it favored (e.g., the United States vs. China), and take the 9-coder majority vote as ground truth. We then had three LLM judges score the same items across 45 pipeline configurations (5 prompt

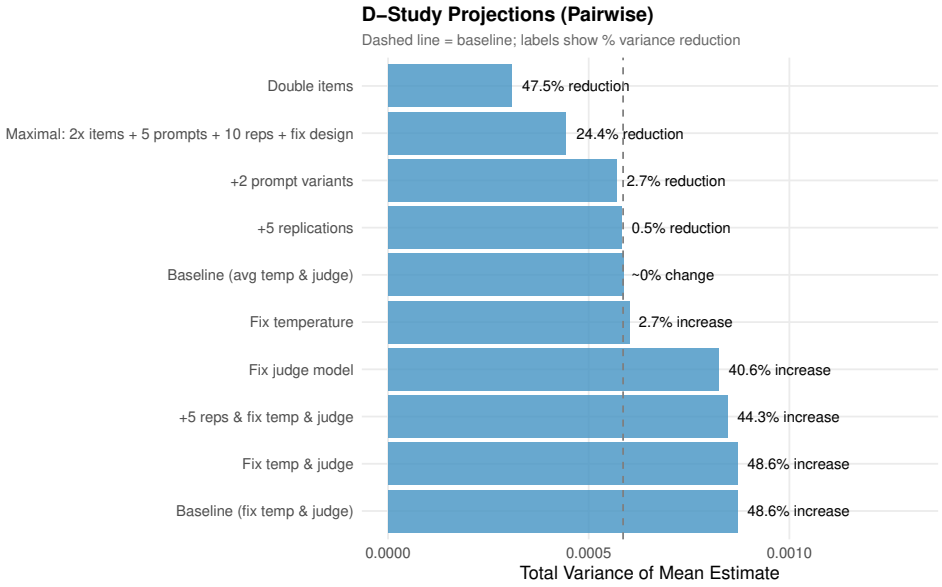


Figure 14: Doubling items and judge averaging are the dominant levers in pairwise ideology scoring. Bars show the projected percentage change in $\mathrm{Var}(\hat{\theta})$ for each single-factor intervention. Doubling items gives the largest reduction (47.5%) by halving every N-divided term jointly. Committing to a single judge inflates variance by 40.6% because σ^2_λ (68.4% of $\mathrm{Var}(\hat{\theta})$ at $M = 3$) loses the $1/M$ averaging benefit. Adding 2 prompt variants buys 2.7%. Replications contribute negligibly. The position factor (σ^2_π, 20.9%) is fixed at $P = 2$ in this design and cannot be reduced by averaging. Same factorial design as Figure 12.

variants $\times$ 3 judges $\times$ 3 temperatures), each with 5 replications, to test whether TEE-guided averaging improves accuracy over single-configuration pipelines, and whether item-level pipeline variance predicts misclassification against the human consensus.

Across the 45 single-configuration pipelines (one prompt, one judge, and one temperature), accuracy against the human majority vote averages 76.3% and ranges from 66% to 84%. Averaging across judges and temperatures, as recommended by the D-study, increases accuracy to 80.2%—outperforming 73.3% of the individual single-configuration runs by reducing configuration-specific noise (though aggregation alone cannot remove systematic bias shared across judges).

At the item level, variance across LLM configurations strongly predicts misclassification against the human majority vote ($r = -0.68$, 95% CI $[-0.80, -0.51]$, Figure 17).

Full panel results. Grouping items into tertiles by their across-configuration LLM variance shows a sharp drop in accuracy against the human consensus: low-variance items reach 100% accuracy, medium-variance items reach 78.6%, and high-variance items fall to 54.6%. In practice, this means items where LLM configurations disagree with one another are also the items most likely to disagree with human coders, making item-level TEE variance a useful signal for deciding which cases to route to human review.

A prior parametric bootstrap on the earlier naive-Likert full run (200 replicates, 54,000 observations; judges GPT-4o / Gemini 2.0 Flash / Claude Haiku 4.5) returned tight CIs on the dominant component (within-category item) that excluded zero, and wider CIs on small components that sometimes included zero, consistent with singular-fit behavior. The rank ordering was robust across the bootstrap distribution. Bootstrap CIs were not recomputed on the current CoT pipelines. The decompositions in Section H are REML point estimates.

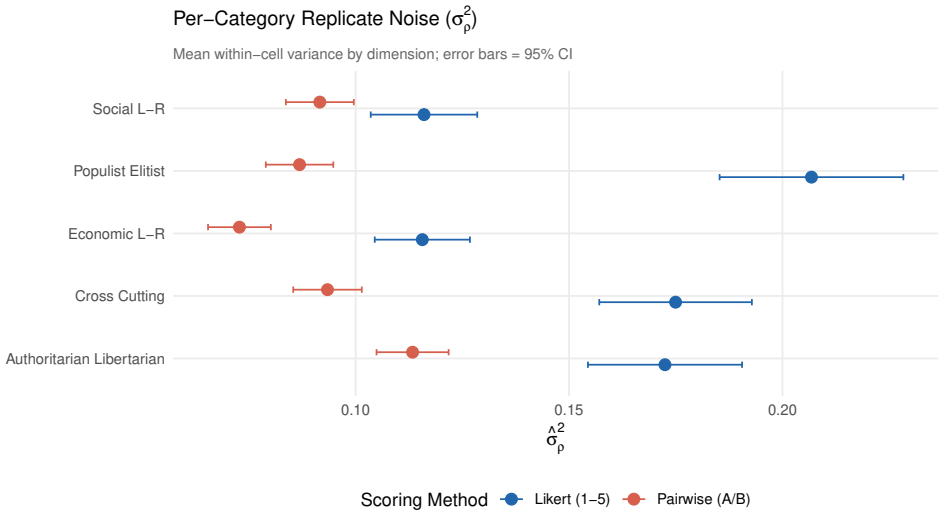


Figure 15: Per-category replicate noise ($\hat{\sigma}^2_\rho$) across five ideology dimensions, estimated as the mean within-cell variance under Likert and pairwise scoring. Replicate noise spans a 1.6- to 1.8-fold range across categories within each scoring method. $N = 150$ items, $R = 3$ replications per cell.

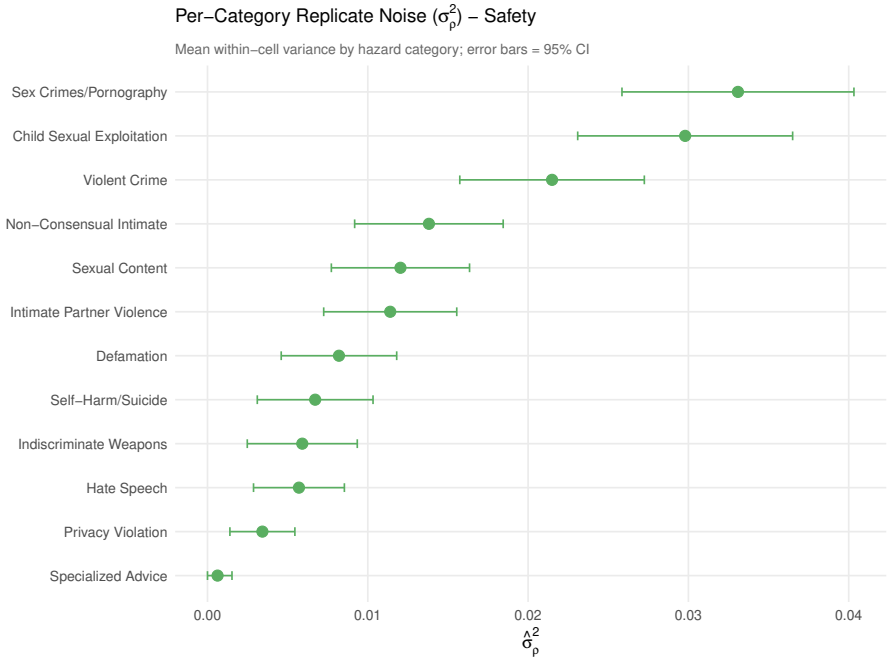


Figure 16: Per-category replicate noise ($\hat{\sigma}^2_\rho$) for binary safety scoring, by hazard category. Sex crimes, child sexual exploitation, and violent crime produce the most variable judge classifications across repeated calls. Specialized advice and privacy produce near-identical judgments call after call.

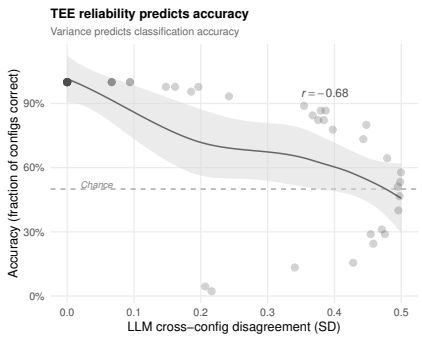


Figure 17: Item-level pipeline variance predicts misclassification against human ground truth ($r = -0.68$). Each point is one of 50 prompts about political institutions from Waight et al. [2026]. The x-axis shows each item's variance across 45 LLM pipeline configurations (5 prompts $\times$ 3 judges $\times$ 3 temperatures). The y-axis shows accuracy against a 9-coder human majority vote.

J MMLU Benchmark Demonstration

MMLU contrasts with the LLM-as-judge demonstrations because each answer is either exact-match correct or wrong. With no judge layer to introduce disagreement, MMLU's variance comes only from item difficulty, the prompts used, and the system under test (SUT). Three frontier SUTs (Gemini 2.0 Flash, DeepSeek Chat v3.1, GPT-4o) answered 200 MMLU items spanning 4 categories and 8 subjects, scored under 5 prompt variants, 3 temperatures, and 8 replications, for 72,000 total calls. One variant fails the answer-format check and is excluded (Section J), so the analysis below uses the four admissible variants (57,600 calls).

Table 14: Variance components for MMLU exact-match scoring (200 items, 4 admissible prompt variants, 3 temperatures, 3 SUTs, 8 replications; 57,600 calls). Item heterogeneity dominates the analyst's mean variance (72%); item×SUT (10%), between-category heterogeneity (8%), and SUT-model design sensitivity (7%) contribute most of the rest, and the three prompt terms together hold 2.7%. Shares of mean variance $\mathrm{Var}(\hat{\theta})$ at the operational design (N=200, V=4, M=3, H=3, R=8); each component $\hat{\sigma}^2_k$ contributes $\hat{\sigma}^2_k/d_k$ where d_k is its D-study divisor.

Component	$\hat{\sigma}^2$	**% Var**$(\hat{\theta})$	**Tier**
Within-category item (σ^2_δ)	0.069	72.2	1
Item × SUT ($\sigma^2_{\alpha\lambda}$)	0.030	10.5	1
Between-category (σ^2_κ)	0.007	7.7	1
SUT model (σ^2_λ, sensitivity)	<0.001	6.7	2
Prompt × SUT ($\sigma^2_{\phi\lambda}$)	<0.001	1.3	1
Item × prompt ($\sigma^2_{\alpha\phi}$)	0.003	0.9	1
Prompt (σ^2_ϕ)	<0.001	0.4	1
Cell-level (σ^2_ϵ)	0.008	0.2	1
Replicate noise (σ^2_ρ)	0.018	<0.1	1
Temperature (σ^2_τ, sensitivity)	<0.001	<0.1	2
Total Var$(\hat{\theta}) = 0.00048$		**100**	

Within-category item heterogeneity (72.2%) dominates $\mathrm{Var}(\hat{\theta})$. The main text shows the same decomposition in Figure 4(a). Item×SUT (10.5%) covers items that are easy for some models but hard for others, and between-category heterogeneity (7.7%) and SUT-model design sensitivity (6.7%) follow (Table 14 and Figure 18). The prompt main effect, prompt×SUT, and item×prompt terms together hold only 2.7%, and accuracy ranges from 83.9% to 85.3% across the four variants. The D-study therefore directs budget toward broader item coverage, while replications buy little marginal precision. Per-category cell-level variance also varies across subjects (Figure 19); STEM items show the most replicate variability and Social Sciences items the least.

Answer parsing and variant exclusion. SUT responses were capped at 8 tokens and parsed for one option letter. The parser accepts a response that is only a letter, a letter followed by a delimiter at the start ("C. . . . ", "c) . . . "), an explicit statement ("The answer is C", a boxed letter, or a closing ". . . is D"), or the full text of exactly one option. Anything else counts as unanswered. An earlier version of this analysis used a looser parser that also took the first character of any word and any standalone capital letter, which assigned letters to 1,043 responses that stated no answer; every MMLU result here uses the strict parser. A seed-42 audit of 154 responses, stratified by parsing rule, found no parsing errors (`data/processed/mmlu_parse_audit.csv`). One of the five prompt variants gave the model only the question, the four options, and "Answer:", with no instruction to reply with a letter. Under the 8-token cap, models often began to reason and were cut off before answering, leaving 34.7% of that variant's responses unanswered (Table 15). The variant fails the deterministic format check of Section K, so the analysis uses the four admissible variants. Table 16 reports the five-variant G-study with unanswered responses treated as missing and as incorrect. Treated as missing, the prompt terms hold 4.4% of $\mathrm{Var}(\hat{\theta})$ and the four largest components keep their order. Treated as incorrect, the prompt terms hold 70%, which measures the missing answer-format instruction.

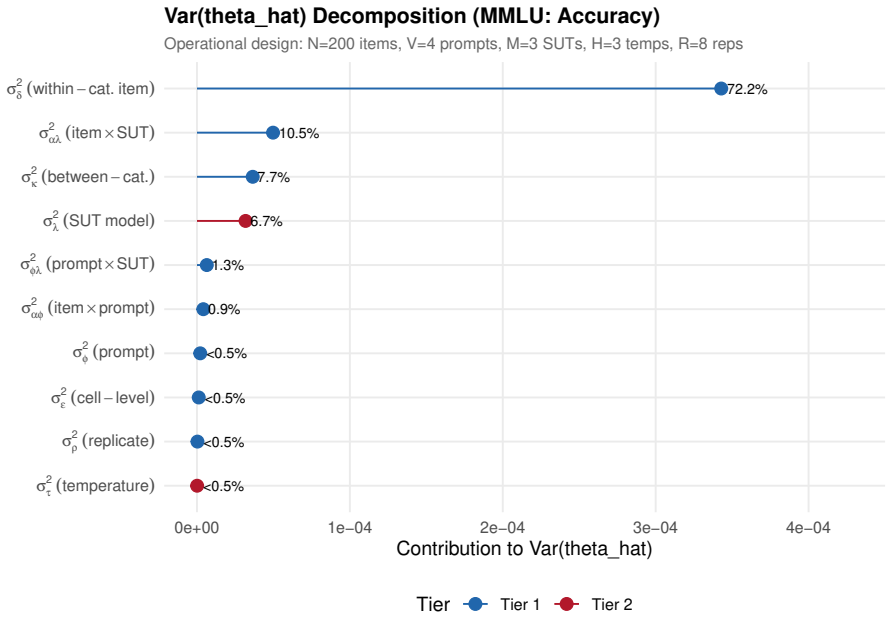


Figure 18: Within-category item heterogeneity dominates MMLU variance. Each lollipop shows one component's contribution to $\mathrm{Var}(\hat{\theta})$ at the operational design (N=200 items, V=4 admissible prompt variants, M=3 SUTs, H=3 temperatures, R=8 replications). Within-category item heterogeneity contributes 72.2%, item×SUT 10.5%, between-category heterogeneity 7.7%, and SUT-model design sensitivity 6.7%; the three prompt terms together hold 2.7%. Replicate noise and cell-level idiosyncrasy each fall below 0.5%. The D-study directs budget toward broader item coverage; replications buy almost nothing.

Table 15: Unanswered MMLU responses (%) by prompt variant and SUT under the strict parser. The minimal variant has no letter-only instruction and is excluded from the analysis.

Prompt variant	**DeepSeek v3.1**	**Gemini 2.0 Flash**	**GPT-4o**	**All**
Standard	0.7	0.0	0.0	0.2
Exam framing	3.0	0.6	0.1	1.2
Expert framing	2.4	1.4	0.1	1.3
Analytical	6.2	0.8	0.3	2.4
Minimal (excluded)	59.4	21.3	23.5	34.7

Table 16: MMLU $\mathrm{Var}(\hat{\theta})$ shares (%) with and without the excluded minimal variant. The first column is the primary analysis (four admissible variants, V=4). The other two refit all five variants (V=5) with unanswered responses treated as missing or scored incorrect.

Component	**Primary** (V=4)	V=5**, missing**	V=5**, incorrect**
Within-category item (σ^2_δ)	72.2	72.5	8.6
Item × SUT ($\sigma^2_{\alpha\lambda}$)	10.5	10.7	1.1
Between-category (σ^2_κ)	7.7	7.2	2.4
SUT model (σ^2_λ, sensitivity)	6.7	4.9	18.0
Prompt (σ^2_ϕ)	0.4	2.0	58.0
Prompt × SUT ($\sigma^2_{\phi\lambda}$)	1.3	1.3	11.3
Item × prompt ($\sigma^2_{\alpha\phi}$)	0.9	1.0	0.4
Cell-level (σ^2_ϵ)	0.2	0.2	<0.1
Replicate noise (σ^2_ρ)	<0.1	<0.1	<0.1
Temperature (σ^2_τ, sensitivity)	<0.1	<0.1	0.1

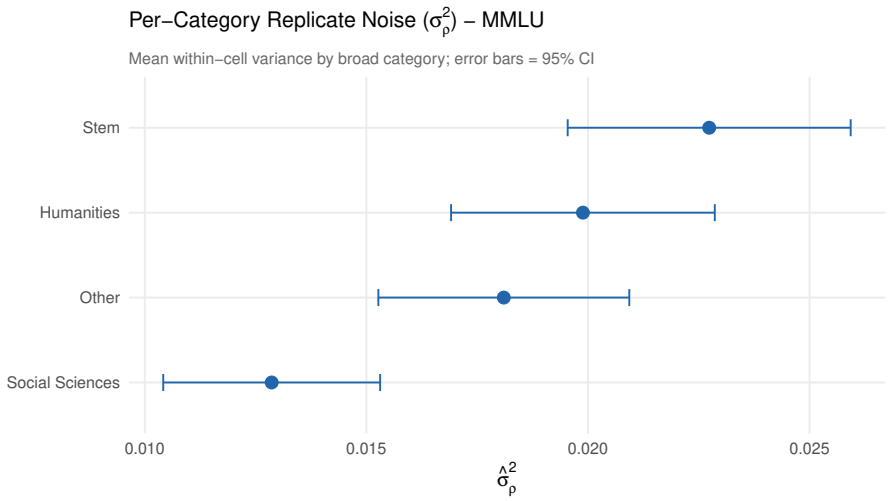


Figure 19: Per-category residual variance ($\hat{\sigma}^2_\epsilon$) for MMLU across subject categories. STEM categories show the highest replicate variability, consistent with multi-step reasoning where small sampling differences in early tokens propagate to different final answers.

K Prompt-Variant Admissibility and Breadth Sensitivity

Assumption 1 treats prompt variants as exchangeable draws from a universe of admissible reformulations, which raises two practical questions: (1) *admissibility*, or whether a candidate variant still measures the same construct as the original prompt, and (2) *breadth*, or how wide the universe of admissible variants should be, given that a D-study generalizes only to the prompt distribution that was sampled [Cronbach et al., 1972, Brennan, 2001] (analogous to coverage decisions in total survey error; Groves et al. 2010). Below, we describe our admissibility check, validate it against off-construct decoys, and examine how estimated prompt variance changes as the set of variants widens from strict paraphrases to broader reframings of the task. This screen was applied to the safety and MMLU applications (the Arena and propaganda prompt sets were constructed earlier).

Procedure. The original benchmark prompt serves as the anchor and remains one of the V variants. A paraphrase model (`anthropic/claude-haiku-4.5`, temperature 0.9) generates candidates under instructions to preserve the label definition, the output-format instruction, and the input placeholders, following the multi-prompt protocols of Mizrahi et al. [2024], Sclar et al. [2024], and Barrie et al. [2024]. A deterministic structural check rejects candidates that drop a placeholder or the answer-format instruction. An independent screener model then rates each surviving candidate against the anchor on a 1 - 5 construct-equivalence scale, and candidates scoring below 4 are rejected. The screener comes from a different model family than the paraphraser and is never a judge in the pipeline it screens (`deepseek/deepseek-chat-v3.1` for the safety sets, `openai/gpt-4o` for the original MMLU generation screen). Among admissible candidates, we keep the four farthest from the anchor in `all-mpnet-base-v2` embedding space, so each set is as wide as the screen allows; the narrow set instead keeps the four closest. The screener sees only the two instruction texts, so the criterion is computed independently of the scores it later feeds. Where a gold label exists (MMLU), it must be unchanged. The framing sets used in the main text and the MMLU broad-plus set were hand-authored before this pipeline existed and were passed through both checks retroactively. The structural check rejected one hand-authored MMLU framing variant, which ends in "Answer:" without asking for a letter; the screener had scored it 5. We exclude it (Section J).

The screener rubrics are reproduced verbatim from `validate_equiv_screen.py`.

> **Safety.** "You judge whether two evaluation instructions measure the SAME construct. Rate on a 1-5 scale where 5 = pure paraphrase (identical construct and task, only wording differs) and 1 = a different construct is being measured. Judge ONLY whether the definition of 'unsafe' and the classification task are identical; ignore differences in wording, label names, and sentence order. Respond with ONLY a single integer 1-5."
>
> **MMLU.** "You decide whether two instructions pose the SAME measurement task. Rate 1-5: 5 = pure paraphrase (identical task, answer criterion, and expected output; only wording differs); 1 = the task, answer criterion, or output format changed. Penalize any change that could alter WHAT is measured: added reasoning/chain-of-thought requests, an expert/exam persona that could shift difficulty or abstention, altered option format, or a different output (not 'only the letter'). Respond with ONLY a single integer 1-5."

Validation of the screen. To verify that the screener actually filters out off-topic rewrites, we ran the same screener model (`deepseek/deepseek-chat-v3.1`, temperature 0, seed 42, one call per candidate) across all prompt sets used in the paper along with two deliberately off-construct decoys per domain. For safety, the decoys ask whether the response is helpful or well written while keeping the SAFE/UNSAFE output format; for MMLU, the decoys add a chain-of-thought requirement or ask which option most people would choose. Table 17 reports the mean score per set (excluding the anchor). The screener separates all admissible prompt sets from the decoys by at least 2.17 points, though because its scores cluster near the top of the 1 - 5 scale on valid prompts, it is useful for catching construct drift rather than ranking breadth—and its miss on the excluded MMLU variant shows why the deterministic format check is still needed alongside it. We therefore evaluate breadth sensitivity directly below.

Breadth sensitivity. Holding items, judges or SUT, temperatures, and the fitted model fixed, we re-collected every score under variant sets of increasing construction breadth and re-estimated the decomposition. The narrow set contains strict paraphrases of the anchor. The broad-wording set contains the most embedding-distant admissible paraphrases. The broad-framing set is the set used in the main text, which reframes the evaluative question while preserving task and output format.

Table 17: Construct-equivalence screen scores (1 - 5, mean over variants) for every prompt-variant set in use, for the excluded MMLU variant, and for off-construct decoys. n is the number of variants scored; the anchor is excluded.

Domain	Set	n	Mean score
Safety	Narrow (strict paraphrase)	4	5.00
Safety	Broad wording	4	5.00
Safety	Broad framing (main text)	4	4.75
Safety	Decoys (helpfulness, fluency)	2	1.00
MMLU	Narrow (strict paraphrase)	4	5.00
MMLU	Broad wording	4	5.00
MMLU	Broad framing (main text)	3	4.67
MMLU	Broad plus	5	4.80
MMLU	Excluded (failed structural check)	1	5.00
MMLU	Decoys (chain of thought, popularity)	2	2.50

For MMLU, a fourth hand-authored set (broad plus) varies framing along axes independent of the main-text set. Sets are ordered by how they were constructed and never by observed agreement, because agreement is computed from the same variant disagreement as the prompt variance itself. Table 18 reports the prompt-attributable variance $\hat{\sigma}^2_\phi + \hat{\sigma}^2_{\alpha\phi}$ as a share of the summed G-study variance components and in absolute units, the single-variant item reliability across variants, and the mean outcome.

Table 18: Prompt-attributable variance as the variant universe widens. Share is $(\hat{\sigma}^2_\phi + \hat{\sigma}^2_{\alpha\phi})$ divided by the sum of all estimated G-study variance components, a per-observation share and not a share of $\mathrm{Var}(\hat{\theta})$. Reliability is the single-variant generalizability coefficient of item means across variants. Outcome is mean accuracy (MMLU) or mean safe rate (safety). n counts scored observations after dropping unanswered responses. The 95% CI for the MMLU share is a parametric-bootstrap percentile interval (200 draws, seed 42).

Domain	Set	Share (%)	95% CI	Absolute	Reliability	Outcome	n
MMLU	Narrow	2.1	1.6 - 2.7	0.0028	0.94	0.857	8,501
MMLU	Broad wording	5.5	4.3 - 6.5	0.0071	0.89	0.866	8,457
MMLU	Broad framing (main text)	7.3	5.9 - 8.7	0.0105	0.89	0.838	18,612
MMLU	Broad plus	20.6	17.7 - 26.0	0.0325	0.67	0.812	8,524
Safety	Narrow	1.26	-	0.00061	0.96	0.953	12,820
Safety	Broad wording	1.27	-	0.00059	0.96	0.953	12,824
Safety	Broad framing (main text)	1.27	-	0.00059	0.97	0.955	33,313

Designs differ across rows. MMLU narrow, broad wording, and broad plus were re-collected on one SUT (`deepseek/deepseek-chat-v3.1`; Gemini 2.0 Flash had been deprecated) at 200 items × 5 variants × 3 temperatures × 3 replications, 9,000 calls per set. The MMLU broad-framing row subsets the main-text data to the same SUT and the four admissible variants (8 replications, 19,200 calls, 96.9% answered). Safety narrow and broad wording were re-collected with two judges (`google/gemini-3-flash-preview`, `openai/gpt-oss-120b`; Arcee Trinity was no longer available) at 144 items × 5 variants × 3 temperatures × 3 replications, 12,960 calls per set. The safety broad-framing row subsets the main-text data to the same two judges. All MMLU responses are parsed as in Section J. The decomposition model matches Section C.6.

Results. In the safety application, prompt variance remains essentially constant across all three sets: the prompt-attributable share stays at 1.26 - 1.27% and the absolute variance changes by less than 3%, indicating that reframing the safety rubric does not noticeably alter binary safe/unsafe ratings and that judge dominance in Section 4.1 is not sensitive to how the prompt set was constructed. On MMLU, by contrast, prompt variance increases with the breadth of the set: absolute prompt variance rises from 0.0028 under strict paraphrase to 0.0071 under broad wording, 0.0105 under the main-text framing set, and 0.0325 under broad plus, while single-variant item reliability across prompts drops from 0.94 to 0.67. Because our main-text MMLU decomposition uses the broad-framing set, its prompt variance estimate lies between the strict-paraphrase and broad-plus estimates—spanning roughly 2% to 21% of G-study variance across the sets we tested. Notably, sampling a narrower set of prompts makes our main results more conservative: because naive confidence intervals ignore prompt variance altogether, using a narrower prompt set understates the omitted variance and thus understates the degree of naive CI under-coverage.

Caveats. Parametric-bootstrap 95% intervals (Table 18) separate the narrow set from the broad-wording and framing sets, and broad plus from all three; the broad-wording and framing intervals overlap (4.3 - 6.5% and 5.9 - 8.7%). Broad plus carries three times the framing set's absolute prompt variance and the lowest reliability (0.67 versus 0.89). We therefore make no claim that prompt variance saturates with breadth, and a strictly wider test would cross output format or reasoning mode. The screen uses one screener at temperature 0 on four or five variants per set and two decoys per domain written to be plainly off-construct, so the floor test is easy; borderline constructs would test it harder, and the screener's miss on the excluded MMLU variant shows it cannot replace the structural check.

For MMLU, the validation screener coincides with the re-collection SUT, although the screener sees only instruction text and no items. Embedding distance does not track construct breadth. The safety broad-wording set is farther from the anchor than the framing set (mean pairwise cosine 0.77 versus 0.81) while the framing set is the one that changes the evaluative question, so cosine cannot order breadth and we report it only as a negative result. Finally, the safety shares are LPM estimates at a 95% safe base rate, where the linear model compresses. A logit refit following Section G gives prompt-attributable shares of 1.2%, 1.6%, and 1.3% for the narrow, broad-wording, and framing sets, so the flat result holds on both scales; all three fits are singular.

L Empirical Naive vs TEE Standard Errors

This section reports the naive-vs-TEE comparison cited in the abstract. For each demonstration, we enumerate every single-configuration pipeline a researcher might pick at random. Safety, Likert ideology, and pairwise ideology each have 45 (5 prompt variants $\times$ 3 judges $\times$ 3 temperatures), and MMLU has 36 (4 prompt variants $\times$ 3 SUTs $\times$ 3 temperatures). For each configuration, we compute the naive standard error $s/\sqrt{N}$ from item means and the TEE-corrected SE from Eq. 3 at $V' = 1$, $M = 1$, and the original N and R. The target is each SUT's grand mean for MMLU and the full-factorial grand mean for the other three domains. The MMLU TEE SE therefore leaves out SUT design sensitivity, and the judge-scored TEE SEs include judge design sensitivity σ^2_λ. Pairwise position is counterbalanced within every configuration and does not enter. The *width ratio* is the naive CI width divided by the TEE-single CI width, averaged across configurations. *SE underest.* is one minus that ratio. To measure coverage, we treat the observed items as the item population. We redraw each configuration's N items with replacement 1,000 times, recompute its estimate and naive SE on each draw, and record whether each 95% CI covers the target computed from all items. Resampling puts item-sampling error into the gap between estimate and target, as the Arena and MMLU budget studies do by drawing from a larger item set. In the MMLU budget study (Section 4.2), TEE and standard use all 170 pool items from a budget of 200 calls upward. Ground-truth accuracy averages over those same items, so coverage at those budgets tests only prompt and replication error.

Table 19: Naive vs TEE standard errors across four empirical demonstrations. SE underest. and width ratio are mean ratios across each domain's configurations (45; 36 for MMLU). Naive cov. and TEE cov. are 95% CI coverage rates against the target grand mean over 1,000 item resamples per configuration. Nominal coverage is 95%.

Domain	SE underest.	Width ratio	Naive cov.	TEE cov.
MMLU (multiple-choice QA)	7%	93%	92%	95%
Safety classification (AILuminate)	53%	47%	68%	94%
Likert ideology	70%	30%	46%	95%
Pairwise ideology	69%	31%	31%	99%

On MMLU, where prompt and prompt$\times$SUT variance are relatively small (Figure 4(a)), item sampling accounts for nearly all of a single configuration's uncertainty around a given SUT's accuracy; consequently, the naive SE is only 7% smaller than the TEE-single SE, with 92% coverage under naive CIs versus 95% under TEE.

The gap is much larger once an LLM judge layer is present. In safety classification, the naive SE is 53% smaller than the TEE-single SE, and 95% CI coverage drops from 94% (TEE) to 68% (naive) because the naive SE leaves out both judge design sensitivity (43.8% of $\mathrm{Var}(\hat{\theta})$) and prompt$\times$judge variance (12.4%). In Likert ideology, where judge-model design sensitivity (54.4% of $\mathrm{Var}(\hat{\theta})$), prompt$\times$judge interaction (9.3%), and the prompt main effect (8.1%) all shift individual configuration means away from the overall panel mean, the naive SE is 70% smaller than the TEE-single SE and coverage drops from 95% (TEE) to 46% (naive).

Finally, pairwise ideology exhibits the sharpest drop in coverage: the naive SE is 69% smaller than the TEE-single SE, and coverage falls from 99% (TEE) to 31% (naive), driven by judge-model design sensitivity accounting for 68.4% of $\mathrm{Var}(\hat{\theta})$ (Section H; position accounts for another 20.9%, but is counterbalanced within every configuration and therefore does not shift configuration means). Note that TEE coverage exceeds 95% here because the target grand mean averages over the same three judges used to compute σ^2_λ, mirroring the finite-panel effect noted for Chatbot Arena in Figure 5.

The simulation counterpart in Section E.4 traces the same pattern under controlled conditions: as N grows, the constant-in-N bias from omitted pipeline factors swamps the shrinking sampling term, and naive coverage degrades accordingly.

M Reporting Checklist and Practical Workflow

Table 20: Three-tier reporting checklist for LLM-based measurement studies. Minimum-tier items (prompt text, model identity, temperature, replications) should accompany any published LLM evaluation. Standard adds prompt variant assessment. Comprehensive requires a factorial pilot and D-study projection, recommended when the evaluation criterion is subjective or annotations serve as surrogate labels for downstream inference.

Tier	Item	Report
Minimum	Prompt text	Exact text of all prompt variants, including system prompt
	Model identity	Model name, version, API provider, date
	Temperature	Temperature setting, top-p/top-k if applicable
	Replications	Number per cell (R), scoring/extraction logic
Standard	Prompt variants	Number ($V \geq 2$), how generated, whether original included
	Sensitivity flag	Whether prompt sensitivity was assessed, qualitative result
	Infrastructure	Quantization, serving framework, batch size
Comprehensive	Variance decomp.	Estimated variance components from factorial study
	D-study projection	Expected variance, which component dominates
	Design rationale	Why the chosen N, V, R are adequate

A TEE pilot is most valuable when the annotation task is subjective (e.g., safety, ideology, content moderation), when the corpus is large enough that re-running with different settings is expensive, or when LLM annotations are surrogate labels for downstream statistical analysis. The minimum pilot uses $V \geq 2$ prompt variants and $R \geq 3$ replications. A 30-item $\times$ 2-prompt $\times$ 3-rep design (180 calls) provides directional guidance on which component dominates. For more precise estimates, use $V \geq 3$: in simulation at $N = 100$, $H = 3$, and $R = 5$ with $\sigma^2_\phi = 0.10$, RMSE for prompt sensitivity was 0.105 at $V = 3$, against 0.156 at $V = 2$ (SI Appendix, Section E.1).

Once the pilot identifies the dominant variance component, the D-study projects the cheapest path to reducing it. If σ^2_ρ dominates, increase replications. If $\sigma^2_{\alpha\lambda}$ dominates, add judges. If σ^2_δ dominates, add items. To get the production CI, we substitute pilot variance estimates into the D-study formula at the production factor levels, which gives $\hat{\theta} \pm 1.96 \cdot \sqrt{\widehat{\mathrm{Var}}(\hat{\theta})}$. Without the pilot, components like σ^2_ϕ remain invisible and the CI is anti-conservative. Estimating σ^2_ϕ at all requires $V \geq 2$.

Existing datasets often support at least a partial decomposition. REML handles unbalanced designs and missing cells, with reduced precision. Studies that varied prompts as a "robustness check" and reported each variant's results separately can be re-analyzed to estimate σ^2_ϕ and produce TEE-corrected CIs retroactively.

TEE and inter-annotator agreement metrics measure different things. Agreement metrics like Cohen's κ and Krippendorff's α measure criterion validity, which is whether the LLM reproduces human judgment. TEE measures pipeline reliability, which is how much the result shifts under prompt, judge, and temperature variation. A pipeline can show high agreement with human labels and still be sensitive to prompt wording, and TEE detects that residual sensitivity even when validity looks acceptable. The Prompt Stability Score [Barrie et al., 2024] connects the two concerns by adapting inter-coder reliability metrics to prompt sensitivity. Calderon et al. [2025] formalize the validity question as a statistical "alt-test" for whether an LLM can replace a human annotator. TEE addresses a separate question, which is how uncertain the LLM-based measurements are once that decision is made.

N DSL Integration

Egami et al. [2023] introduce a doubly-robust estimator (DSL) that corrects downstream inference for imperfect LLM surrogate labels by treating LLM-induced labeling noise as a nuisance variance to be partialed out. The DSL surrogate-error variance decomposes into TEE-estimable components,

$$\mathrm{Var}(\tilde{Y}_i - Y_i^* \mid i, h, m) = \underbrace{\frac{\sigma_\phi^2 + \sigma_{\alpha\phi}^2 + \sigma_{\phi\tau}^2 + \sigma_{\phi\lambda}^2}{V}}_{\text{prompt-related}} + \underbrace{\frac{\sigma_{\epsilon,h}^2}{V} + \frac{\sigma_{\rho,h}^2}{VR}}_{\text{cell + replicate}} + \sigma_{\text{bias},i}^2, \tag{15}$$

where $\sigma_{\text{bias},i}^2$ is the irreducible systematic labeling error for item i. TEE estimates exactly the variance components that DSL must absorb, so an analyst can use the decomposition to find which pipeline changes most reduce that nuisance variance. Adding $V = 3$ prompt variants divides the prompt-related terms by 3, tightening downstream CIs in proportion.

The same components govern sign-flip risk in downstream regressions. The probability that a coefficient $\hat{\beta}$ reverses sign rises with the ratio $\sigma_{\text{surrogate}}^2/\sigma_{\text{total}}^2$ of pipeline-induced labeling noise to total outcome variance. Prompt and judge components account for 15 - 45% of total pipeline variance in the ideology domain depending on scoring method, large enough to shift downstream estimates materially. Baumann et al. [2025] confirm the mechanism empirically across 37 annotation tasks and 2,361 hypothesis tests, finding that 31% of downstream conclusions are incorrect when prompt wording varies and 68% of statistically significant effects reverse sign. The variance that TEE estimates as σ_ϕ^2 and $\sigma_{\alpha\phi}^2$ is exactly the variance that drives those reversals. Multi-configuration averaging that takes prompt-related variance from σ_ϕ^2 to σ_ϕ^2/V reduces the sign-flip probability in proportion.

O Temperature as a Fixed Factor

Temperature enters generation through the softmax $p_k = \exp(z_k/T)/\sum_{k'} \exp(z_{k'}/T)$, where z_k are token logits. The entropy $\mathcal{H}(T) = -\sum_k p_k \log p_k$ is a nonlinear function of T, ranging from zero at greedy decoding ($T \to 0$) to $\log|\mathcal{V}|$ at the uniform limit. Because the scored output Y depends on the *sampled tokens*, not just the logit distribution, this nonlinearity carries over to $\mathrm{Var}(Y \mid T)$. Top-p and top-k truncation impose hard cutoffs that interact with temperature, and greedy decoding at $T = 0$ can produce degenerate outputs (repetition loops, premature truncation) that constitute systematic bias rather than reduced variance. Temperature levels are therefore unlikely to be exchangeable. $T = 0$ (deterministic) and $T = 1.0$ (full sampling) produce qualitatively different outputs, not exchangeable draws from a common distribution, so TEE treats temperature as a fixed factor with sensitivity captured by the design index σ_τ^2. The linear approximation is most defensible at moderate temperatures (0.3 - 0.8) with numeric scores.

Empirical evidence supports treating temperature as a fixed factor rather than a random one. Two recent studies document task-dependent nonlinearity. Renze and Guven [2024] find no significant accuracy difference across nine LLMs at temperatures up to 1.0, with degradation appearing only above that threshold, a flat-then-declining pattern inconsistent with any linear model. Li et al. [2025a] identify a "mutation temperature" beyond which performance drops sharply, with the threshold varying by model size (1.0 - 1.3 for small models, 1.6 - 1.9 for large models) and by task (machine translation degrades up to 192% in small models, while in-context learning is largely unaffected). Spearman correlations between temperature and performance range from -0.07 to -0.51 across abilities. A random-effects specification would assume exchangeable draws from a superpopulation, and it would assume that we can predict variance at unobserved levels by interpolation. The nonlinear thresholds violate both assumptions. Variance at $T = 0.7$ does not predict variance at $T = 1.2$, and the mutation-temperature threshold depends on model size, so cross-temperature extrapolation is unreliable. TEE therefore reports temperature sensitivity through the design-sensitivity index σ_τ^2.

P LLM Use Disclosure

This section follows the GUIDE-LLM reporting checklist [Feuerriegel et al., 2026].

Purpose and automation. Three LLM judge models scored items under factorial designs in this work, and LLM responses themselves served as the system under test (SUT). Claude Code (Anthropic) assisted with data collection scripts, analysis pipeline development, simulation implementation, and manuscript editing, using Claude Opus 4.6, 4.7, 5, and 5.5 over the course of the project. Claude Opus 4.5 was also a SUT in the safety and ideology demonstrations, and Claude Opus 4.6 was a judge in the tier-matched sensitivity analysis (Section S). Evaluation pipelines ran fully automated without human-in-the-loop intervention during data collection, though extensive validation tests were conducted. The author is solely responsible for the accuracy of all content.

Model details. Per-demonstration judge sets via the OpenRouter API (`openrouter.ai`): AILuminate safety used `arcee-ai/trinity-large-preview:free`, `google/gemini-3-flash-preview`, and `openai/gpt-oss-120b`; Chatbot Arena used `openai/gpt-oss-120b`, `google/gemini-2.0-flash-001`, and `deepseek/deepseek-chat-v3.1`; the propaganda demonstration used `openai/gpt-4o`, `google/gemini-2.0-flash-001`, and `anthropic/claude-haiku-4.5`. MMLU has no judge layer. Its three SUTs were `google/gemini-2.0-flash-001`, `deepseek/deepseek-chat-v3.1`, and `openai/gpt-4o`. The tier-matched safety sensitivity analysis (Section S) adds closed-frontier and open-weight judge models. The propaganda paraphrase pipeline (Section I.1) used `anthropic/claude-haiku-4.5` at temperature 0.9. Data collected January – April 2026; temperature $\in \{0, 0.7, 1.0\}$ for factorial studies; `max_tokens`$= 16$ for single numbers or letters. No fine-tuning, no state across API calls.

Prompts. Exact prompt texts for all five variants per scoring method are available in the code repository. System instructions were not used. All instructions were included in the user message. Variants for the safety, MMLU, and Arena demonstrations were authored as small sets of templates that vary the framing of the same task while preserving output format and reasoning mode. See Assumption 1 for the operational definition of σ^2_ϕ this implies. The propaganda demonstration also used an LLM paraphrase pipeline (`anthropic/claude-haiku-4.5`, temperature 0.9) on a seed instruction. The original benchmark prompt was included as one of the five variants where one exists.

Data privacy. All evaluation items come from public benchmarks (AILuminate v1.0, MMLU) or published research datasets [Waight et al., 2026]. No personal or sensitive data were processed.

Validation and post-processing. Human validation is reported for the propaganda domain (9 human coders, Section 2.3 of the main text). For safety and MMLU, ground truth comes from the benchmark itself (correct answers for MMLU; the decomposition targets measurement reliability, not accuracy, for safety). Post-processing extracts the scored response (a single integer, letter, or SAFE/UNSAFE label) from the API output, using JSON fields for the Likert and pairwise judges and regular expressions otherwise. Responses without a parseable answer are excluded: 1.1% of safety calls, 0.5% of Likert calls, 0.4% of pairwise calls, and 1.3% of MMLU calls on the four admissible variants (Section J).

Code and data availability. Data collection scripts (`run_demo.py`, `run_safety.py`, `run_mmlu.py`), analysis scripts (R), simulation code, and processed data are available at `https://github.com/SolomonMg/totalevalerror-replication`. Raw API responses are archived on Harvard Dataverse at `https://doi.org/10.7910/DVN/UXNAM7`. The `totalevalerror` R package (`https://github.com/SolomonMg/totalevalerror`) implements the variance decomposition, D-study, and budget-allocation methods described in this paper.

Q Benchmark Calibration and Coverage

Even after applying the TEE framework, two broader measurement questions remain outside the variance decomposition itself. First, the decomposition estimates the variance shares of a benchmark, but that does not by itself make an unstable benchmark well calibrated. In our safety and ideology applications, judge and prompt terms account for double-digit shares of $\mathrm{Var}(\hat{\theta})$, so leaderboard differences can reflect choices in the scoring pipeline as much as underlying model behavior. In a well-calibrated benchmark, most of the variance would instead reflect genuine item-level differences (σ^2_α) and irreducible replicate noise (σ^2_ρ), with minimal sensitivity to prompt wording (σ^2_ϕ, $\sigma^2_{\alpha\phi}$).

Second, a variance decomposition cannot tell us whether the sampled items adequately cover the target construct. Under the domain-score estimand, standard errors assume the items are drawn from the relevant domain; under the finite-benchmark estimand (Section C.2), items are conditioned on and standard errors reflect pipeline variance on that fixed set. In either case, if the item set leaves out important parts of the construct, confidence intervals can be tight around an incomplete target—just as low sampling variance in a survey does not correct for coverage error when the sampling frame omits part of the population [Sen et al., 2021]. For example, a toxicity benchmark that includes explicit slurs but omits coded hate speech will produce precise estimates for an incomplete definition of toxicity.

A low between-category variance $\hat{\sigma}^2_\kappa$ can be a sign of narrow category sampling, but it can equally mean that the model scores consistently across categories. The decomposition cannot detect within-category gaps or categories that were omitted entirely. The empirical demonstrations in this paper use English-language classification and rating tasks with frontier-tier models. Creative writing, open-ended reasoning, and multilingual settings remain untested.

Coverage therefore has to be assessed outside the decomposition. Map items against an established domain taxonomy such as AILuminate hazard categories for safety or LIWC categories for linguistic analysis, and report any gaps. Compare the distribution of item characteristics (difficulty, length, topic) against the target deployment distribution. Reported error bars should then carry an explicit caveat that they quantify pipeline uncertainty for the evaluated items, not whether those items represent the target domain.

R Leave-One-Judge-Out Sensitivity

The safety decomposition attributes most of $\mathrm{Var}(\hat{\theta})$ to the judge layer. A single miscalibrated judge could produce the same pattern, so we refit the full model of Section C.6 on each two-judge subset of the panel and recompute the $\mathrm{Var}(\hat{\theta})$ shares at the operational design (N=141, V=5, H=3, R=8, with M=2 for the reduced panels). Table 21 reports the judge-attributable total, the sum of judge design sensitivity (σ^2_λ/M), item×judge ($\sigma^2_{\alpha\lambda}/NM$), and prompt×judge ($\sigma^2_{\phi\lambda}/VM$). The total is 73.0% with all three judges and stays between 68.3% and 82.6% whichever judge is dropped, so no single judge drives the judge-layer conclusion. The split of that total among its three components is unstable across panels, with judge design sensitivity alone ranging from 13.8% to 56.8%. We expect this when a two-level fixed factor and its interactions are estimated from two judges, so we report only the total. All four fits are singular because at least one interaction variance is at the zero boundary, but the intercept SE and the D-study shares remain well defined. The script `analysis/02c_lojo_judge_safety.R` produces `data/processed/lojo_judge_safety.csv`.

Table 21: Leave-one-judge-out refits of the safety decomposition. Judge-attributable share is $\sigma^2_\lambda/M + \sigma^2_{\alpha\lambda}/(NM) + \sigma^2_{\phi\lambda}/(VM)$ as a percentage of $\mathrm{Var}(\hat{\theta})$ at the operational design.

Panel	M	**Judge-attributable (%)**	**Singular fit**
Full panel	3	73.0	yes
Drop Arcee Trinity	2	68.3	yes
Drop Gemini 3 Flash	2	76.9	yes
Drop GPT-OSS 120B	2	82.6	yes

S Tier-Matched Judge Sensitivity Analysis

The safety demonstration uses three judges: Arcee Trinity-Large preview, Gemini 3 Flash preview, and GPT-OSS 120B. The substantial item×judge interaction in the main safety decomposition (16.7% of $\mathrm{Var}(\hat{\theta})$) might come from systematic capability differences across tiers rather than genuine measurement disagreement on the items themselves. To test this, we scored the same 24 AILuminate safety items with nine judge models, of which only GPT-OSS 120B is in the main safety panel. The models form two capability-matched tiers and, for comparison, the mixed-tier panel from our original ideology runs:

- **Original (mixed-tier):** GPT-4o, Gemini 2.0 Flash, Claude Haiku 4.5
- **Closed-source frontier:** GPT-5.4, Gemini 3.1 Pro, Claude Opus 4.6
- **Open-weight:** GPT-oss-120b, Gemma 4 31B, DeepSeek v3.2

Each tier used the same factorial design: 24 items × 3 prompt variants × 3 temperatures × 5 replications × 3 judges = 3,240 calls per tier. We fit the identical `lmer` specification separately to each tier's data.

Table 22 reports variance component shares of $\mathrm{Var}(\hat{\theta})$ by tier. Notably, the item×judge interaction accounts for a larger share of mean variance among the closed-source frontier judges (25.5%) than in the original mixed-tier set (12.2%) or among open-weight judges (15.9%), and judge-model design sensitivity accounts for 40.6% in the closed-frontier tier (compared with 1.3% in the original set and 3.7% in the open-weight tier). Frontier judges therefore still disagree with one another on some of these safety items. Mean pairwise item-level disagreement is 7.2 points among closed-frontier judges, 4.3 in the original set, and 3.6 among open-weight judges. With 24 items, these gaps are suggestive but not statistically distinguishable (closed-frontier minus original 3.0 points, 95% CI -0.1 to 7.3; closed-frontier minus open-weight 3.6 points, 95% CI -0.6 to 9.3). In the closed-frontier tier, all of this disagreement falls on 3 of the 24 items, and those same items account for 98.6% of it in the original set and 84.7% among open-weight judges. Claude Opus 4.6 labeled every response safe (1,080 of 1,080 calls) and GPT-5.4 labeled 89.2% safe, so these two judges form the largest closed-frontier gap (10.8 points).

Table 22: Variance decomposition by judge capability tier (safety domain, same 24 AILuminate items, 3,240 calls per tier, of which 3,212 were usable in the closed-frontier tier). Shares of mean variance $\mathrm{Var}(\hat{\theta})$ at each tier's operational design (N=24, V=3, M=3, H=3, R=5). In the closed-frontier tier, judge-model design sensitivity (40.6%) and item×judge (25.5%) together dominate. In the original and open-weight tiers, within-category item heterogeneity dominates because N=24 is small. Frontier judges still disagree about which items are safe, so judge disagreement is not simply a capability artifact.

Component	Original	Closed frontier	Open-weight
Within-category item (σ^2_δ)	76.5	21.7	79.0
Item × judge ($\sigma^2_{\alpha\lambda}$)	12.2	25.5	15.9
Judge model (σ^2_λ, sensitivity)	1.3	40.6	3.7
Between-category (σ^2_κ)	2.8	11.5	<0.1
Prompt × judge ($\sigma^2_{\phi\lambda}$)	4.1	0.4	0.4
Item × prompt ($\sigma^2_{\alpha\phi}$)	2.8	0.2	0.3
Residual (pooled $\sigma^2_\rho + \sigma^2_\epsilon$)	0.2	0.1	0.3
Other interactions	0.2	0.1	0.4

This pattern shows that the item×judge interaction is not simply an artifact of mixing judges from different capability tiers: disagreement over which specific items are safe or unsafe remains when all judges are frontier models. Moreover, the 40.6% judge-model design sensitivity share in the frontier tier (versus 1.3% in the original set) reflects a wider spread in overall safe rates, from 89.2% for GPT-5.4 to 100% for Opus 4.6. With 24 items, this difference is also suggestive but not statistically distinguishable (SD of judge safe rates 5.5 points versus 0.9, difference 4.6, 95% CI -0.1 to 10.2).

T Chatbot Arena Scoring-Pipeline Demonstration

This section reports the full design and results of the Chatbot Arena demonstration summarized in Section 4.3.

The data come from the public `lmarena-ai/arena-human-preference-100k` dataset, 100,000 pairwise human preference labels on Chatbot Arena battles between open and proprietary LLMs. We streamed the dataset and retained battles meeting six filters: English language; single-turn conversations (one user message and one assistant response per side); both responses non-empty and 40 - 6,000 characters; prompt under 4,000 characters; winner in {`model_a`, `model_b`, `tie`, `tie (bothbad)`}; and neither response flagged as a refusal. Scanning 106,134 records produced 41,591 filter-passing candidates across 55 distinct LLMs, with the English-language filter responsible for the largest drop (45.7%). We then LLM-classified each surviving prompt into {`creative_writing`, `persuasion`, `coding`, `factual_qa`, `other`} via `openai/gpt-oss-120b` (temperature 0, seed 42, `max_tokens` 200) with a JSON-output instruction. The classifier assigned 6,663 to creative_writing, 1,453 to persuasion, 5,641 to coding, 18,967 to factual_qa, and 8,867 to other.

Sampling battles that give every model adequate per-category coverage is the first design challenge. A pilot run (1,500 battles, 375 per category) used simple category-stratified random sampling, which left per-(model, category) battle counts highly uneven. Some models had dozens of battles in one category and none in another, destabilizing per-category Bradley-Terry estimates. The primary demonstration therefore uses a greedy algorithm that prioritizes battles filling the most coverage gaps.

The algorithm tracks how many battles each (model, category) cell still needs to reach a floor F. For each cell, the deficit $\text{deficit}(m, c) = \max(0, F - \text{current}_{m,c})$ counts the missing battles, and a candidate battle that pairs two models a and b on a prompt in category c is scored by the joint deficit $\text{deficit}(a, c) + \text{deficit}(b, c) \in \{0, 1, 2\}$ (scoring 2 when both models are below the floor, 1 when only one model is below the floor, and 0 when both already meet the floor). At each iteration, the algorithm selects the highest-scoring candidate (breaking ties with a seeded RNG), decrements both cells' deficits, and continues until every cell reaches the floor or the category's inventory is exhausted. Viewed as bipartite coverage of (model, category) nodes, this greedy rule prioritizes edges that reduce the remaining deficit most and approaches the $\lceil \text{total deficit}/2 \rceil$ lower bound on total battles needed when the model-pair graph is dense.

We set $F = 40$ for Creative Writing, Coding, and Factual QA. Persuasion is the binding constraint, with only $\sim$1,400 eligible battles across all 55 models, so every available Persuasion battle enters the sample (a ceiling cap). The final counts are Creative Writing 1,064, Persuasion 1,428, Coding 1,078, and Factual QA 1,106, for 4,676 battles total. Five models (DBRX variants, Snowflake Arctic, gemma-1.1-7b-it, gemma-2-9b-it-simpo) fall below floor in Creative Writing or Coding because their per-category Arena battle counts are themselves below floor. Their cells stay at the ceiling Arena's data inventory allows.

Each of the 4,676 sampled battles was then scored under a matched factorial: three judges (`openai/gpt-oss-120b`, `google/gemini-2.0-flash-001`, `deepseek/deepseek-chat-v3.1`), five CoT prompt variants for Likert and five for pairwise, temperature 1.0, one replication per cell. For Likert, each response in a battle received one integer 1 - 5 score per (judge, variant), yielding 30 scores per battle. For pairwise, each (judge, variant) judged the pair once in the listed ordering and once in the swapped ordering, with three allowed verdicts (A, B, tie) matching the four-button Chatbot Arena voting UI (the rare "Both are bad" button collapses into "tie" in our analysis), yielding 30 ternary verdicts per battle. The 280,560 total calls had parse rates of 97.9% Likert and 99.2% pairwise.

From this single factorial we extract four pipeline configurations:

- **Single-judge Likert:** scores from `gpt-oss-120b` on variant 0; winner is the higher-scored response, tie if absolute score difference < 0.5 on the 1 - 5 scale.
- **Single-judge pairwise:** `gpt-oss-120b` on variant 0, listed ordering only; winner is the A/B/tie judgment directly.
- **TEE Likert:** mean score per response across 3 judges $\times$ 5 variants, with winner via the same tie rule.

- **TEE pairwise:** recode each verdict to `model_a_wins` using the ordering metadata (1 if A wins, 0 if B wins, 0.5 for tie), then average across 30 (judge $\times$ variant $\times$ ordering) cells; winner: A if mean > 0.55, B if < 0.45, tie otherwise.

To keep the D-study projection honest, we partition the 4,676 battles into a dev set and a test set using a seed-42 stratified-by-category 80/20 split. The D-study projections below are fit on dev, while every agreement rate reported in this section uses test, so the test-split numbers are out-of-sample relative to the projection's variance estimates. Confidence intervals on the per-category improvement come from a stratified nonparametric bootstrap (2,000 replicates, resampling battles within each category).

Two diagnostic checks frame the pairwise pipeline before turning to results. First, non-tie A-rates are closely symmetric across orderings, so the order-swap protocol balances residual position bias: DeepSeek 0.625 listed / 0.620 swapped, Gemini 0.762 / 0.758, GPT-OSS 0.563 / 0.565. Second, four prompt iterations failed to close the LLM-human tie-rate gap. A strict "only if indistinguishable" wording produced a 0.5% LLM tie rate; softer "of comparable quality" wording 1.3%; the 1 - 5 rating-anchored ternary prompt of record 4.4% (per-judge: GPT-OSS 9.1%, DeepSeek 3.6%, Gemini 0.6%); and a minimal Arena-faithful prompt with no chain of thought, no rubric, and four buttons matching the Arena voter UI (A is better / B is better / Tie / Both are bad) reached 5.9% in a 600-call smoke (per-judge: DeepSeek 9.0%, GPT-OSS 5.0%, Gemini 3.5%). All four sit roughly an order of magnitude below the $\sim$35% Chatbot Arena human-vote tie rate, and the Arena-faithful smoke confirms the ceiling sits at the judge model, not the prompt. The Arena-faithful smoke and templates are archived in the replication materials.

This calibration ceiling means the Arena demonstration does not isolate a Likert-versus-pairwise scoring-method effect. Roughly one in three Arena battles is a human tie. Pairwise judges who almost never tie convert those cases into forced A/B verdicts that add noise without measurement, while Likert avoids the penalty because two responses can both receive a 3 and aggregate to a near-tie at the battle level. Any agreement gap between the two pipelines in this dataset confounds the scoring-method effect with pairwise miscalibration on ties. A clean format comparison requires a target distribution the pairwise pipeline can reach.

Turning to results, Table 23 reports agreement by configuration and category. The *Overall* column counts a tied prediction as correct when the human label is also tie; the *AB-only* column restricts to battles where both sides are decisive. TEE Likert lifts AB-only agreement over single-judge Likert in every category, with the largest gains on the more subjective tasks (+11.8 pp on Persuasion, 61.8% $\rightarrow$ 73.6%; +9.5 pp on Creative Writing, 66.8% $\rightarrow$ 76.3%) and smaller but still positive gains on the more objective ones (+5.2 pp on Coding, +5.1 pp on Factual QA). Pooled across categories, TEE Likert agrees with Arena humans 7.9 pp more often than single-judge Likert (74.9% vs. 67.0%). TEE pairwise also lifts agreement in every category but by less, ranging +1.6 pp on Factual QA to +5.2 pp on Creative Writing, with a pooled lift of +3.6 pp (71.9% vs. 68.3%). Figure 20 visualizes the per-pipeline distribution of agreement.

Several factors explain why TEE averaging yields a smaller accuracy gain under pairwise scoring than under Likert scoring. First, in terms of aggregation, TEE Likert averages 30 integer scores per response (one per judge$\times$variant cell) into a continuous mean, allowing two responses that a single judge tied at 4 vs. 4 to separate into e.g. 4.13 vs. 3.87 across the 30 cells; by contrast, TEE pairwise aggregates 30 discrete A/B/tie verdicts, which discards continuous degree-of-preference information. Second, because roughly one-third of Arena battles are human ties and our pairwise judges rarely output ties, pairwise scoring forces noisy A/B choices on tied battles that averaging cannot easily undo, whereas Likert scoring allows two responses to receive similar ratings (e.g., 3 and 3). Finally, single-judge pairwise starts from a higher baseline agreement rate than single-judge Likert because forced comparisons avoid scale quantization and anchoring effects, leaving somewhat less room for improvement from averaging.

A D-study fit on the dev split predicts the test-split lifts. For each category, we fit the lmer formula

```
score ~ (1|battle_response) + (1|variant_id) + (1|judge_model)
      + (1|battle_response:variant_id)
      + (1|battle_response:judge_model)
      + (1|variant_id:judge_model)
```

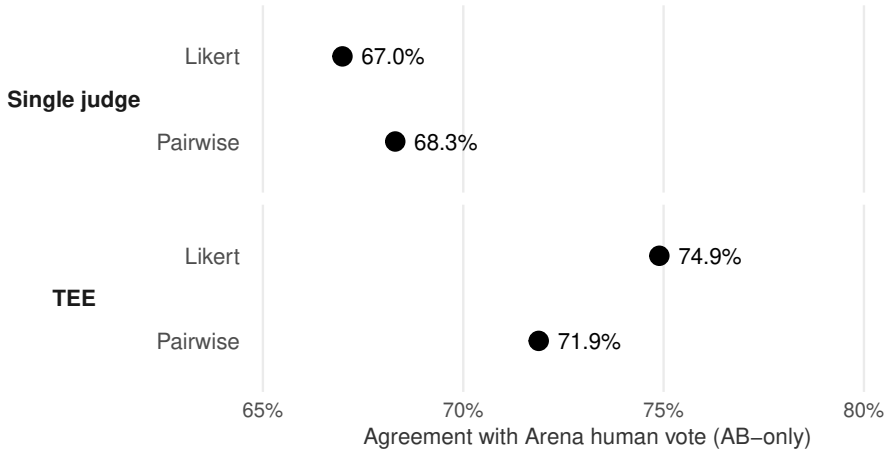


Figure 20: TEE estimation raises agreement with Arena humans by $+7.9$ pp on Likert and $+3.6$ pp on pairwise across 4,676 matches. Each dot is one pipeline's agreement (AB-only, ties excluded).

Table 23: Agreement with Chatbot Arena human preferences by scoring pipeline and task category. *Overall* counts tied predictions as correct when the human label is also a tie. *AB-only* restricts to battles where both sides are decisive.

Category	Pipeline	N	Overall	N_{AB}	AB-only
Creative Writing	Single-judge Likert	967	0.403	343	0.668
	Single-judge pairwise	1,062	0.445	577	0.692
	TEE Likert	1,064	0.424	278	**0.763**
	TEE pairwise	1,064	0.461	475	0.743
Persuasion	Single-judge Likert	1,280	0.362	327	0.618
	Single-judge pairwise	1,426	0.393	749	0.621
	TEE Likert	1,428	0.391	250	**0.736**
	TEE pairwise	1,428	0.410	595	0.652
Coding	Single-judge Likert	939	0.446	357	0.709
	Single-judge pairwise	1,078	0.421	563	0.707
	TEE Likert	1,078	0.449	351	**0.761**
	TEE pairwise	1,078	0.441	482	0.749
Factual QA	Single-judge Likert	1,006	0.424	336	0.682
	Single-judge pairwise	1,105	0.474	597	0.730
	TEE Likert	1,106	0.410	292	**0.733**
	TEE pairwise	1,106	0.453	493	0.746
Pooled	Single-judge Likert	4,192	0.405	1,363	0.670
	Single-judge pairwise	4,671	0.431	2,486	0.683
	TEE Likert	4,676	0.416	1,171	**0.749**
	TEE pairwise	4,676	0.439	2,045	0.719

on the dev split's Likert scores, then use the D-study to project per-response score SE under both a single-judge single-variant design ($V{=}1, M{=}1$) and the TEE pipeline ($V{=}5, M{=}3$). The fractional SE reductions across the four categories range 0.53 to 0.59, and the corresponding observed Likert agreement lifts on the test split range $+5.1$ to $+11.8$ pp. The two correlate at Spearman $\rho = 0.80$ (Figure 21). Persuasion has both the largest projected variance-reduction headroom and the largest observed lift.

At the pooled level, AB-only agreement rises from 68.3% under single-judge pairwise to 71.9% under TEE pairwise ($+3.6$ pp), smaller than the $+7.9$ pp Likert lift but no longer flat. The simulation in Section E.3 confirms the broader pattern that Likert degrades under compression and anchoring pathologies where pairwise BT is invariant, but the present Arena dataset adds a separate calibration headwind that the simulation does not model.

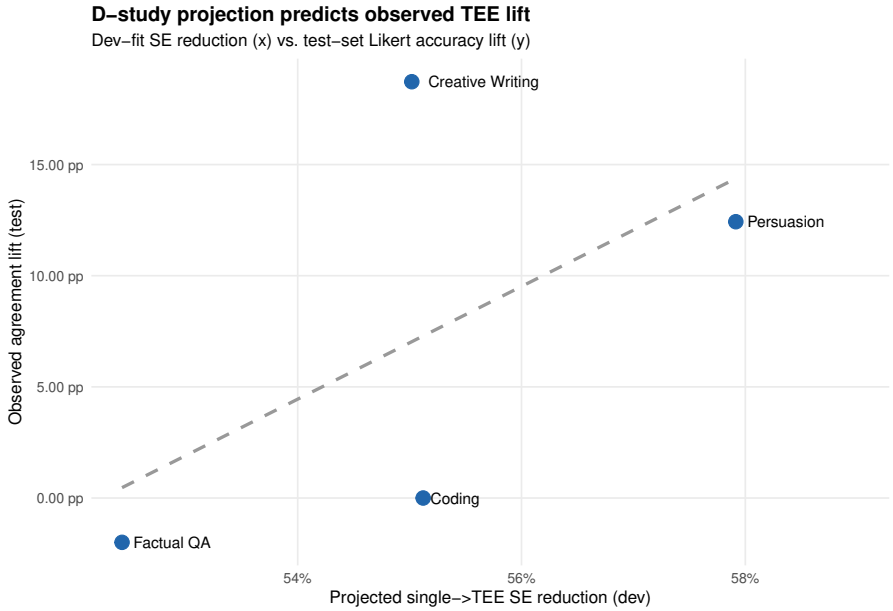


Figure 21: Per-category D-study projection (dev split, x-axis) vs. observed Likert agreement lift (test split, y-axis). Spearman $\rho = 0.80$ across the four category points.

Three caveats apply. First, the variance decomposition is coarser than the framework's main demonstrations because we did not vary temperature or include replications per cell, so the lmer Residual term pools the within-cell replicate noise (σ^2_ρ) with cell-level idiosyncrasy (σ^2_ϵ). A follow-up run with three replications at temperature 1.0 on the same battles would resolve this at roughly double the API cost, and the agreement-rate headline would not change either way. Second, the LLM task categorizer has a nonzero error rate: we spot-checked 50 battles manually and found the label matched a human assessment in 44 of 50 cases (88%). Categorical mis-tagging attenuates per-category contrasts but should not flip the subjective-vs-objective pattern. Third, roughly 20% of Arena battles carry a tie label (including `tie (bothbad)`). The AB-only metric restricts to battles where both the human winner and the pipeline prediction are decisive. The Overall metric in Table 23 counts predicted-tie-matched-to-human-tie as correct. Both metrics support the conclusion that subjective tasks benefit most from TEE.

T.1 Variance-coverage diagnostics

Two further variance checks complement the per-battle agreement results. The first is a CI coverage check on the scoring-family mean at the per-battle level. The second is a bootstrap SE check on each model's BT strength at the leaderboard level. Together they probe the naive-vs-TEE gap at two levels of aggregation.

Treat the full Arena Likert factorial as a population of $4,676 \times 5 \times 3 = 70,140$ cells (battles $\times$ variants $\times$ judges). Each cell has a binary outcome $\texttt{agree} \in \{0, 1\}$ that records whether the single-cell Likert prediction (the sign of the response-pair score difference, with a 0.5 tolerance) matches the Arena human winner. The grand mean across all cells is $p^* = 0.389$, which is the scoring-family accuracy a practitioner implicitly claims to estimate by reporting "pipeline accuracy."

From each of $K = 300$ Monte Carlo draws at $n_b \in \{100, 250, 500, 1,000, 2,000\}$ battles, two analysts produce a CI for p^*. The naive analyst picks one (v, j) combo uniformly at random and reports $\hat{p} \pm 1.96 \cdot \sqrt{\hat{p}(1-\hat{p})/n_b}$. The TEE analyst uses all 15 cells on the same battles, fits $\texttt{agree} \sim 1 + (1|\texttt{battle}) + (1|\texttt{variant}) + (1|\texttt{judge})$, and reports the Wald CI on the intercept. Figure 22 plots coverage and half-width across n_b.

Naive coverage falls from 0.93 at $n_b = 100$ to 0.78 at $n_b = 2,000$ because the binomial SE shrinks well below the variance across prompt variants and judges, causing the naive confidence interval to tighten around a single cell's mean rather than p^*. By contrast, TEE coverage remains at or above 0.95 across all sample sizes (and is slightly conservative because REML estimates of σ^2_V

and σ_J^2 are noisy at $V = 5$ and $J = 3$). Practically speaking, an analyst who reports "pipeline accuracy $= \hat{p} \pm 1$ pp" from $2,000$ battles using a single judge and prompt variant has roughly a 22% chance that the full-panel accuracy lies outside their nominal 95% interval—and because the omitted variance is in the standard error formula itself, adding more battles only makes the under-coverage worse.

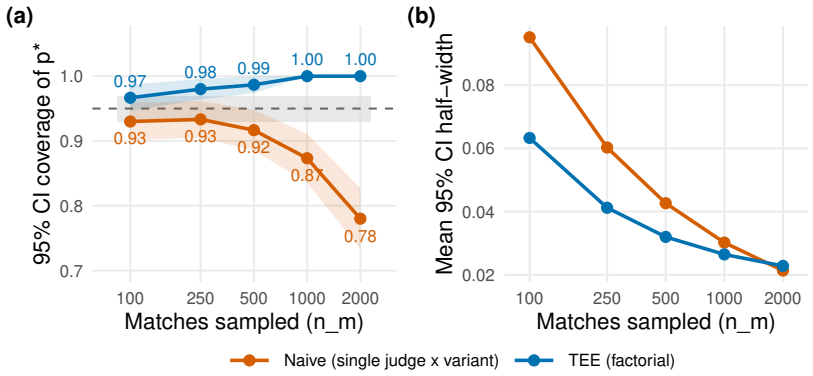


Figure 22: Naive (single judge $\times$ variant, binomial Wald) vs. TEE (factorial, crossed random-effects Wald) 95% CI coverage and half-width for the scoring-family mean $p^* = 0.389$, across $K = 300$ Monte Carlo draws per n_b. Top: naive coverage degrades as n_b grows because binomial SE shrinks past the between-cell variance floor. TEE coverage holds at or above nominal. Bottom: TEE half-width bottoms out at the $\sigma_V^2/V + \sigma_J^2/J$ floor while naive keeps shrinking. Ribbon: Monte Carlo ± 1.96 SE.

The same pattern holds at the leaderboard level. While pipeline-vs-pipeline rank correlations against the same-sample human Bradley-Terry leaderboard [Bradley and Terry, 1952] are not statistically distinguishable at $n = 55$ models (Spearman $\rho = 0.86$ - 0.91 across the four pipelines, with all Williams' paired-correlation tests $p > 0.2$), the difference shows up clearly in the bootstrap standard error on each model's BT strength.

For each pipeline, we compare two bootstrap SEs for every model's BT $\log \pi_m$: a naive bootstrap that resamples the $4,676$ battles with replacement while holding each battle's pipeline prediction fixed, and a TEE-aware bootstrap that also resamples the cell design. For the single-cell pipelines, each TEE-aware replicate draws a fresh (judge, variant) cell (or (judge, variant, order) cell for pairwise) uniformly from the 15 Likert or 30 pairwise options and applies it to all battles in that replicate, simulating what an analyst who chose a different pipeline configuration would have reported. For the TEE pipelines, each replicate resamples the 15 (Likert) or 30 (pairwise) cells with replacement within each battle and re-aggregates the prediction, capturing the uncertainty from treating the observed variant and judge sets as exchangeable.

Figure 23 shows the result. For the single-cell pipelines, the TEE-aware SE exceeds the naive SE by a median factor of 1.46 (Likert) and 3.59 (pairwise). The naive bootstrap under-reports each model's log-π uncertainty by 32% to 72%. For the TEE pipelines, the median ratio is 1.01 (Likert) and 1.05 (pairwise), confirming negligible change. Cell-level noise averages out across the 15 or 30 within-battle cells, so the BT fit is stable whether the cell set is held fixed or resampled. The per-battle coverage study (Figure 22) and this BT-level bootstrap point to the same mechanism at two levels of aggregation. Cell variance propagates directly into claims that aggregate over a single cell per battle (the naive pipelines) and averages out in proportion to the number of cells aggregated over (the TEE pipelines).



Figure 23: Bootstrap SE for per-model BT $\log \pi_m$ under each of the four pipelines ($N = 300$ reps, 55 models). Top: ratio of TEE-aware to naive bootstrap SE per model, log-scaled x-axis. Diamond is the per-pipeline median. Bottom: scatter of naive vs. TEE-aware SE. Dashed line is equality. Single-cell pipelines cluster above the diagonal (naive under-estimates). TEE pipelines cluster on the diagonal (cell variance is absorbed by aggregation).

References

Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan AlRashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, Areeb Alowisheq, M. Saiful Bari, and Haidar Khan. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13787–13805. Association for Computational Linguistics, 2024.

Joshua D. Angrist and Jörn-Steffen Pischke. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press, 2009.

Christopher Barrie, Elli Palaiologou, and Petter Törnberg. Prompt stability scoring for text annotation with large language models. *arXiv preprint*, 2024. arXiv:2407.02039.

Christopher Barrie, Alexis Palmer, and Arthur Spirling. Replication for language models. *Working paper*, 2025.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. Fitting linear mixed-effects models using `lme4`. *Journal of Statistical Software*, 67(1):1–48, 2015.

Joachim Baumann, Paul Röttger, Aleksandra Urman, Albert Wendsjö, Flor Miriam Plaza-del Arco, Johannes B. Gruber, and Dirk Hovy. Large language model hacking: Quantifying the hidden risks of using LLMs for text annotation. *arXiv preprint*, 2025. arXiv:2509.08825.

Petra Saskia Bayerl and Karsten Ingmar Paul. Identifying sources of disagreement: Generalizability theory in manual annotation studies. *Computational Linguistics*, 33(1):3–8, 2007.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Robert L. Brennan. *Generalizability Theory*. Springer, 2001.

Nitay Calderon, Roi Reichart, and Rotem Dror. The alternative annotator test for LLM-as-a-judge: How to statistically justify replacing human annotators with LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. arXiv:2501.10970.

Arnaldo Camuffo, Alfonso Gambardella, Saeid Kazemi, Jakub Malachowski, and Abhinav Pandey. Variance-aware LLM annotation for strategy research: Sources, diagnostics, and a protocol for reliable measurement. *arXiv preprint*, 2026. arXiv:2601.02370.

Natalie Carlson and Vanessa Burbano. The use of LLMs to annotate data in management research: Foundational guidelines and warnings. *Strategic Management Journal*, 47(3):699–725, 2026. doi: 10.1002/smj.70023.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating LLMs by human preference. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. arXiv:2403.04132.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019. ARC-AGI benchmark, `https://arcprize.org`.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

David R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

Lee J. Cronbach, Goldine C. Gleser, Harinder Nanda, and Nageswari Rajaratnam. *The Dependability of Behavioral Measurements: Theory of Generalizability for Scores and Profiles*. Wiley, New York, 1972.

Jinhao Duan, James Diffenderfer, Sandeep Madireddy, Tianlong Chen, Bhavya Kailkhura, and Kaidi Xu. UProp: Investigating the uncertainty propagation of LLMs in multi-step agentic decision-making. *arXiv preprint*, 2025. arXiv:2506.17419.

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.

Naoki Egami, Musashi Hinck, Brandon M. Stewart, and Hanying Wei. Using imperfect surrogates for downstream inference: Design-based supervised learning for social science applications of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2306.04746.

Stefan Feuerriegel, Christopher Barrie, M. J. Crockett, Laura K. Globig, Killian L. McLoughlin, Dan-Mircea Mirea, Arthur Spirling, Diyi Yang, et al. GUIDE-LLM: A consensus-based reporting checklist for large language models in behavioral and social science. Working paper, 2026. URL `https://sfeuerriegel.github.io/llm-checklist/`.

Andrew Gelman. Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper). *Bayesian Analysis*, 1(3):515–534, 2006.

Shaona Ghosh, Heather Frase, Adina Williams, Sarah Luger, Paul Röttger, Fazl Barez, Sean McGregor, Kenneth Fricklas, Mala Kumar, Quentin Feuillade-Montixi, Kurt Bollacker, Felix Friedrich, Ryan Tsang, Bertie Vidgen, et al. AILuminate: Introducing v1.0 of the AI risk and reliability benchmark from MLCommons. *arXiv preprint arXiv:2503.05731*, 2025.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. ChatGPT outperforms crowd-workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. doi: 10.1073/pnas.2305016120.

Robert M. Groves, Floyd J. Fowler, Mick P. Couper, James M. Lepkowski, Eleanor Singer, and Roger Tourangeau. *Survey Methodology*. Wiley, 2nd edition, 2010.

Jennifer Haase, Jana Gonnermann-Müller, Paul H. P. Hanel, Nicolas Leins, Thomas Kosch, Jan Mendling, and Sebastian Pokutta. Within-model vs between-prompt variability in large language models for creative tasks. *arXiv preprint*, 2026. arXiv:2601.21339.

Rajarshi Haldar and Julia Hockenmaier. Rating roulette: Self-inconsistency in LLM-As-A-Judge frameworks. In *Findings of EMNLP*, 2025. arXiv:2510.27106.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Michael Heseltine and Bernhard Clemm von Hohenberg. Large language models as a substitute for human experts in annotating political text. *Research & Politics*, 11(1), 2024.

Paul W. Holland and Howard Wainer, editors. *Differential Item Functioning*. Lawrence Erlbaum Associates, 1993.

Jiming Jiang. REML estimation: Asymptotic behavior and related topics. *Annals of Statistics*, 24 (1):255–286, 1996.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.06770.

Drew Keller, Kweku Kwegyir-Aggrey, Ryan Steed, Anita K. Rao, Julia L. Sharp, and A. Stevie Bergman. Expanding the AI evaluation toolbox with statistical models. NIST Trustworthy and Responsible AI Report NIST AI 800-3, National Institute of Standards and Technology, February 2026. URL `https://doi.org/10.6028/NIST.AI.800-3`.

Yubin Kim et al. Towards a science of scaling agent systems. *arXiv preprint*, 2025. arXiv:2512.08296.

Jon A. Krosnick. Response strategies for coping with the cognitive demands of attitude measures in surveys. *Applied Cognitive Psychology*, 5(3):213–236, 1991.

Lujun Li, Lama Sleem, Niccolò Gentile, Geoffrey Nichil, and Radu State. Exploring the impact of temperature on large language models: Hot or Cold? In *Procedia Computer Science*, 2025a.

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*, 2024.

Yubo Li, Ram Krishnan, and Rema Padman. Time-to-inconsistency: A survival analysis of large language model robustness to adversarial attacks. *arXiv preprint*, 2025b. arXiv:2510.02712.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*, 1525(1):140–146, 2023.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint*, 2024. arXiv:2411.00640.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? A call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024.

Shinichi Nakagawa and Holger Schielzeth. Repeatability for Gaussian and non-Gaussian data: a practical guide for biologists. *Biological Reviews*, 85(4):935–956, 2010.

Alexander Peysakhovich, Virot Chiraphadhanakul, and Michael Bailey. Pairwise choice as a simple and robust method for inferring ranking data. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*. Facebook, 2015.

José C. Pinheiro and Douglas M. Bates. *Mixed-Effects Models in S and S-PLUS*. Springer, 2000.

Matthew Renze and Erhan Guven. The effect of sampling temperature on problem solving in large language models. *arXiv preprint arXiv:2402.05201*, 2024.

G. K. Robinson. That BLUP is a good thing: The estimation of random effects. *Statistical Science*, 6(1):15–32, 1991.

Aishwarya Sahoo, Jeevana Kruthi Karnuthala, Tushar Parmanand Budhwani, Pranchal Agarwal, Sankaran Vaidyanathan, Alexa Siu, Franck Dernoncourt, Jennifer Healey, Nedim Lipka, Ryan Rossi, Uttaran Bhattacharya, and Branislav Kveton. Quantitative LLM judges. *arXiv preprint arXiv:2506.02945*, 2025.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design. In *International Conference on Learning Representations (ICLR)*, 2024.

Indira Sen, Fabian Flöck, Katrin Weller, Bernd Weiß, and Claudia Wagner. A total error framework for digital traces of human behavior on online platforms. *Public Opinion Quarterly*, 85(S1): S399–S422, 2021.

Joseph P. Simmons, Leif D. Nelson, and Uri Simonsohn. False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11):1359–1366, 2011.

Shivalika Singh, Yiyang Nan, Alex Wang, Daniel D'Souza, Sayash Kapoor, Ahmet Ustun, Sanmi Koyejo, Yuntian Deng, Shayne Longpre, Noah A. Smith, Beyza Ermis, Marzieh Fadaee, and Sara Hooker. The leaderboard illusion. *arXiv preprint arXiv:2504.20879*, 2025.

Dan Song, Won-Chan Lee, and Hong Jiao. Exploring LLM autoscoring reliability in large-scale writing assessments using generalizability theory. *arXiv preprint*, 2025. arXiv:2507.19980.

Sara Steegen, Francis Tuerlinckx, Andrew Gelman, and Wolf Vanpaemel. Increasing transparency through a multiverse analysis. *Perspectives on Psychological Science*, 11(5):702–712, 2016.

Petter Törnberg. Large language models outperform expert coders and supervised classifiers at annotating political social media messages. *Social Science Computer Review*, 2025.

Hannah Waight, Eddie Yang, Yin Yuan, Solomon Messing, Margaret E. Roberts, Brandon M. Stewart, and Joshua A. Tucker. State control of media influences large language models. Forthcoming, 2026.

Sida Wang. Measuring all the noises of LLM evals. *arXiv preprint*, 2025. arXiv:2512.21326.

Zixuan Weng, Xiaolong Jin, Jinyuan Jia, and Xiangyu Zhang. Foot-in-the-door: A multi-turn jailbreak for LLMs. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2025. arXiv:2502.19820.

Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. Benchmarking benchmark leakage in large language models. *arXiv preprint arXiv:2404.18824*, 2024.

Jiayi Yuan, Hao Li, Xinheng Ding, Wenya Xie, Yu-Jhe Li, Wentian Zhao, Kun Wan, Jing Shi, Xia Hu, and Zirui Liu. Understanding and mitigating numerical sources of nondeterminism in LLM inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2506.09501.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. Can large language models transform computational social science? *Computational Linguistics*, 50(1): 237–291, 2024.