RANKING LARGE LANGUAGE MODELS VIA COMPARISON GRAPH TRUNCATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) evaluation is a central problem in modern machine learning. While benchmark scores provide a standard way to compare models on many tasks with golden-standard answers, evaluating open-ended responses remains challenging because of the lack of a single correct answer. Indeed, pairwise comparison offers a practical approach, but noisy judgments can make the resulting rankings unstable. To address this issue, we repeat each comparison and apply truncation to retain only comparisons with sufficient agreement. Specifically, our method removes comparisons with weak agreement while retaining enough evidence to rank the models. Theoretically, we prove that moderate truncation can reduce ranking error, while removing too many comparisons can increase it. Additionally, we conduct simulations and experiments on LLMs, showing that our method improves ranking accuracy.

1 INTRODUCTION

Large language models (LLMs) are increasingly used in applications ranging from question answering to writing and coding. Their growing use makes reliable evaluation essential for understanding their capabilities and comparing their performance (Liang et al., 2023). Standard benchmarks provide a common basis for such evaluation, particularly on tasks with clearly defined answers (Hendrycks et al., 2021; Cobbe et al., 2021). However, performance on these tasks does not fully capture response quality in open-ended interactions, where multiple answers may be reasonable and reference answers alone may not adequately reflect human preferences (Zheng et al., 2023; Liu et al., 2023). Pairwise evaluation provides a practical approach by asking human or LLM judges to compare two responses to the same prompt (Zheng et al., 2023). For example, Chiang et al. (2024) collect human pairwise preferences and aggregate them to rank models. However, since human evaluation is costly and slow, LLM judges are widely used to automate such comparisons. For instance, Zheng et al. (2023) use GPT-4 as an automated judge and assess its agreement with human preferences. Specifically, given a prompt and two responses generated by different models, a judge model selects the better response or declares a tie. For a given prompt, comparing every pair among n models yields a comparison graph with n vertices and $\binom{n}{2}$ edges, each recording a pairwise judgment. A natural question is how to aggregate these judgments into a reliable ranking of the models.

There are many established approaches to ranking from pairwise comparisons, such as Thurstone model (Thurstone, 1927), Bradley–Terry estimation (Bradley & Terry, 1952), and Rank Centrality (Negahban et al., 2017). Different from traditional statistical setting, observations can be collected at several levels in LLM evaluation. The same model pair can be compared across many prompts, and a fixed evaluation input can receive repeated judgments through stochastic decoding at a nonzero temperature (Liu et al., 2023). Semantically equivalent prompt perturbations provide another way to obtain additional comparisons (Huang et al., 2026). This makes it important to determine how judgments should be combined across prompts and repetitions to obtain the final model ranking.

However, repeated judgments for the same prompt and response pair do not always agree. Figure 1 illustrates two possible outcomes of 20 comparisons, with counts ordered as wins, ties, and losses. The counts $(18, 1, 1)$ show strong agreement in favor of the first model, whereas $(7, 7, 6)$ provide little support for selecting a single outcome. Repeated judgments therefore reveal both the preferred outcome and the degree of agreement behind it. This motivates selecting comparisons according to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

their agreement before ranking aggregation. Specifically, we retain a comparison when its largest vote share meets a chosen threshold, and fit a Bradley–Terry model to the most frequent outcomes of the retained comparisons. See Section 2 for further details. The choice of threshold involves a tradeoff between reducing noise and preserving useful information, as illustrated in Figure 2. With little truncation, many comparisons with weak agreement remain. Moderate truncation reduces their influence while retaining sufficient evidence for ranking. As more comparisons are removed, however, the loss of useful information eventually outweighs the benefit of filtering, and ranking accuracy declines. Theorem 1 establishes this behavior under our statistical model: the limiting ranking error first decreases and then increases along the truncation path. In particular, a suitable threshold yields lower limiting ranking error than either retaining all comparisons or discarding all.

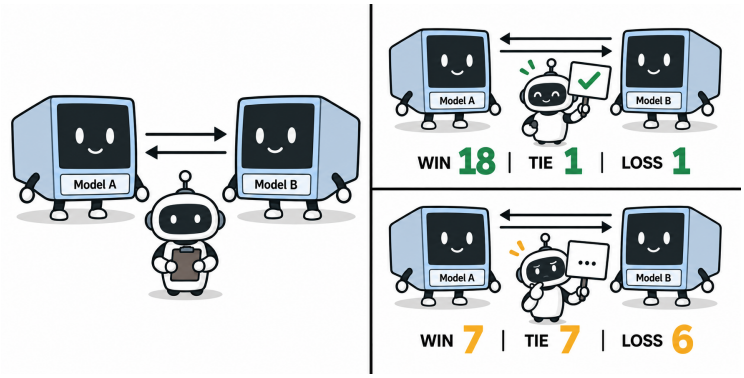

Figure 1: Comparison selection based on repeated judgments. Concentrated votes indicate strong agreement on a comparison outcome, while dispersed votes indicate disagreement. Comparisons with stronger agreement are retained and aggregated to obtain the final model ranking.

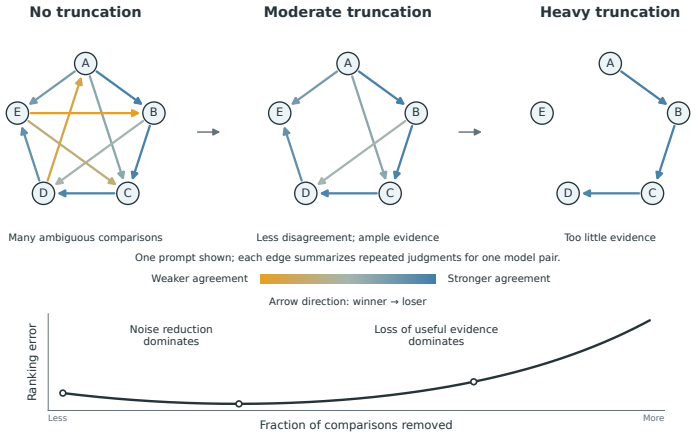

Figure 2: Schematic illustration of the truncation tradeoff. Each edge summarizes repeated judgments for one model pair under the same prompt, with arrows pointing from the observed winner to the loser. Colors range from yellow (weaker agreement) to blue (stronger agreement).

1.1 Our Contributions and Paper Organization

- **Method.** We propose a truncation method for ranking LLMs from repeated pairwise comparisons. It selects comparisons based on agreement across repeated judgments and fits a Bradley–Terry model to the retained outcomes.
- **Theory.** Under a statistical model of repeated comparisons, we derive the limiting ranking error of the Bradley–Terry estimator and identify the optimal truncation threshold. The analysis explains why moderate truncation can improve ranking accuracy, while excessive truncation can reduce it.

- **Experiments.** Simulations and LLM evaluations show that truncation can improve ranking accuracy. Further experiments show that the retained comparisons are more consistent and that the resulting rankings can better predict judge preferences on held-out questions.

Section 2 introduces the method, Section 3 presents the theoretical analysis, and Section 4 reports the experimental results.

1.2 Related Work

LLM-as-a-judge. LLM-as-a-judge methods use language models to assess generated responses, providing a practical alternative to costly human evaluation (Zheng et al., 2023). Different approaches improve this evaluation process through the scoring procedure, the judge model, or the combination of multiple judges. G-Eval uses explicit evaluation steps and structured scoring (Liu et al., 2023), while Prometheus 2 provides a specialized evaluator that supports direct assessment and pairwise comparison under user-defined criteria (Kim et al., 2024). To combine perspectives from different evaluators, Verga et al. (2024) aggregate judgments from a panel of diverse models. However, LLM judgments remain sensitive to factors such as response order and length (Zheng et al., 2023; Shi et al., 2025; Dubois et al., 2024). Our work studies how to use repeated judgments from a fixed evaluator to construct a reliable model ranking. We use agreement across repeated judgments to select comparisons before aggregating their outcomes into the final ranking.

Repeated sampling in LLMs. Multiple LLM outputs can be obtained through stochastic decoding at nonzero temperatures or through variations in the input prompt. For generation, self-consistency samples multiple reasoning paths and selects the most frequent final answer (Wang et al., 2023), while Cobbe et al. (2021) train verifiers to select among sampled solutions. Repeated sampling also supports evaluation: the GPT-4 implementation of G-Eval samples numerical ratings to estimate a weighted evaluation score (Liu et al., 2023). Prompt variation provides another source of observations. Mizrahi et al. (2024) show that instruction wording can substantially affect both model performance and relative rankings, while PromptEval estimates the distribution of performance across prompt formulations under a limited evaluation budget (Polo et al., 2024). For pairwise evaluation, Huang et al. (2026) generate prompt perturbations and select the resulting comparison graphs according to structural consistency before ranking. Our work uses repeated judgments to measure agreement on a pairwise verdict and determine which comparisons should enter the final ranking.

Statistical methods for pairwise comparisons. Classical probabilistic approaches represent items through latent scores. The Thurstone model uses Gaussian latent utilities (Thurstone, 1927), while the Bradley–Terry model expresses win probabilities through relative item strengths (Bradley & Terry, 1952). Davidson (1970) extends the latter to explicitly model ties. Other approaches include Rank Centrality, which constructs a Markov chain from pairwise outcomes and ranks items using its stationary distribution (Negahban et al., 2017), and Borda counting, which orders items by their observed win totals and admits statistical guarantees under general comparison probabilities (Shah & Wainwright, 2018). Particularly relevant is the work of Agarwal et al. (2020) on ranking under adversarial corruption. Starting from empirical win probabilities obtained through repeated Bradley–Terry–Luce comparisons, their method prunes edges using consistency constraints on graph cycles and applies Accelerated Spectral Ranking to the retained graph. They establish parameter recovery guarantees under assumptions on the comparison graph and corruption level. Our method selects comparisons using agreement among repeated judgments for the same prompt and model pair, then fits Bradley–Terry scores to the retained outcomes. We study how the resulting ranking error changes across truncation thresholds.

2 Methods: Repeated Comparisons, Truncation, and Ranking

We consider n candidate models $\mathcal{M} = \{M_1, \ldots, M_n\}$ and m evaluation prompts $\mathcal{X} = \{x_1, \ldots, x_m\}$. Let r_{si} denote the response of model M_i to prompt x_s. The prompt set may also include variants generated by prompt perturbation, each treated as a separate prompt (Huang et al., 2026). For each prompt and model pair $i < j$, a fixed judge compares the same two responses K

times. The outcome of its kth call is

$$Y_{sij}^{(k)} = \begin{cases} +1, & \text{if the judge prefers } r_{si}, \\ 0, & \text{if the judge declares a tie,} \\ -1, & \text{if the judge prefers } r_{sj}. \end{cases} \tag{1}$$

Outcomes are recorded with respect to model identities, including when the presentation order is reversed. We call (s, i, j) a comparison unit, containing the repeated judgments for one prompt and one model pair.

We summarize these judgments by their win, tie, and loss counts:

$$N_{sij}^{+} = \sum_{k=1}^{K} \mathbf{1}\{Y_{sij}^{(k)} = +1\}, \quad N_{sij}^{0} = \sum_{k=1}^{K} \mathbf{1}\{Y_{sij}^{(k)} = 0\}, \quad N_{sij}^{-} = \sum_{k=1}^{K} \mathbf{1}\{Y_{sij}^{(k)} = -1\}. \tag{2}$$

These counts sum to K. To measure agreement within a unit, we use its largest vote share,

$$C_{sij} = \frac{1}{K} \max\{N_{sij}^{+}, N_{sij}^{0}, N_{sij}^{-}\}. \tag{3}$$

A larger C_{sij} means that more calls support the same outcome; it does not necessarily mean that this outcome is correct. Let $\widehat{Y}_{sij}$ denote the most frequent outcome. If several outcomes share the largest count, we choose uniformly among them, independently across units. This choice is made once per unit and reused across thresholds.

For a threshold $\tau \in [0, 1]$, we retain the units in

$$\mathcal{U}_\tau = \{(s, i, j) : C_{sij} \geq \tau\}. \tag{4}$$

Selection takes place separately for each prompt and model pair. Thus, removing a unit does not remove comparisons of the same pair on other prompts. The retained set changes only when the threshold passes an observed value of C_{sij}.

The retained units form a comparison multigraph G_τ, with models as vertices and one edge per retained unit. Edges point from the observed winner to the loser, with ties left undirected. Different prompts can produce parallel edges between the same models, as illustrated in Figure 2.

We aggregate the retained outcomes using the Bradley–Terry (BT) model (Bradley & Terry, 1952). Let $\theta = (\theta_1, \ldots, \theta_n)$ denote the model scores, with larger scores indicating stronger models. For a comparison with no tie, the BT model specifies

$$\mathbb{P}_\theta(i \succ j) = \frac{e^{\theta_i}}{e^{\theta_i} + e^{\theta_j}}, \quad \mathbb{P}_\theta(j \succ i) = \frac{e^{\theta_j}}{e^{\theta_i} + e^{\theta_j}}. \tag{5}$$

These probabilities depend only on score differences, so we impose $\sum_{i=1}^{n} \theta_i = 0$ to fix their common offset.

For fitting, we treat the retained units as independent observations and use the working likelihood

$$L_{\mathrm{BT}}(\theta; \tau) = \prod_{(s,i,j) \in \mathcal{U}_\tau} \mathbb{P}_\theta(i \succ j)^{(1+\widehat{Y}_{sij})/2} \mathbb{P}_\theta(j \succ i)^{(1-\widehat{Y}_{sij})/2}. \tag{6}$$

For a retained win or loss, its contribution is the probability of the observed outcome. With a fixed penalty parameter $\lambda > 0$, we estimate the scores by

$$\widehat{\theta}_\tau = \underset{\sum_{i=1}^{n} \theta_i = 0}{\arg\max} \left\{ \log L_{\mathrm{BT}}(\theta; \tau) - \frac{\lambda}{2} \|\theta\|_2^2 \right\}. \tag{7}$$

We use the same λ across thresholds and rank models by decreasing fitted score, breaking score ties uniformly at random. The positive penalty makes the estimate finite and unique even when truncation disconnects the graph, although it cannot replace missing comparisons between components.

A retained tie has $\widehat{Y}_{sij} = 0$, so both exponents in equation 6 equal $1/2$. It therefore contributes half a win to each model, whereas a removed unit contributes nothing. This convention includes ties in the fit without assigning them a separate outcome probability. Alternatively, the Davidson model explicitly includes a tie probability and can be fitted to the retained comparisons (Davidson, 1970). Our analysis in Section 3 studies the BT estimator in equation 7 under a separate model for repeated judgments.

3 THEORETICAL ANALYSIS: A STATISTICAL MODEL FOR TRUNCATION

In this section, we study how the truncation method in Section 2 affects ranking error under a simple statistical model. Recall that n is the number of models. Let $\theta^\star$ be a centered vector of distinct target scores. For each pair (i, j), we make K calls to the judge on each of m prompts. Let Q_{sij} be the probability that the judge prefers model i to model j on prompt s. To simplify the analysis, we exclude ties and assume $Y_{sij} \in \{-1, 1\}$.

Note that values near $1/2$ describe ambiguous comparisons, while values near zero or one describe comparisons on which the judge usually agrees with itself. Consequently, we draw these probabilities Q_{sij} independently from the density

$$f_{ij}(q) = 1 + \delta(\theta_i^\star - \theta_j^\star)(2q - 1), \quad q \in [0, 1], \tag{8}$$

where $\delta > 0$ and $\delta \max_{i,j} |\theta_i^\star - \theta_j^\star| \le 1$. Conditional on the entire array (Q_{sij}), all calls are independent, with $\mathbb{P}[Y_{sij}^{(k)} = +1 \mid Q_{sij}] = Q_{sij}$ and the remaining probability on -1.

The density describes variation in win probabilities across prompts. Its constant term is a uniform baseline, and its tilt favors the model with the larger target score: $\mathbb{E}[Q_{sij}] = 1/2 + \delta(\theta_i^\star - \theta_j^\star)/6$. However, some prompts can nevertheless produce strong agreement on the wrong ordering. It is therefore significant to analyze the effect of truncations. Specifically, we measure ranking quality by the expected normalized Kendall error, counting a fitted score tie as half an error:

$$\mathcal{R}_{m,K}(\tau) = \frac{1}{\binom{n}{2}} \sum_{i<j} \Big(\mathbb{P}[(\widehat{\theta}_{\tau,i} - \widehat{\theta}_{\tau,j})(\theta_i^\star - \theta_j^\star) < 0] + \frac{1}{2}\mathbb{P}[\widehat{\theta}_{\tau,i} = \widehat{\theta}_{\tau,j}] \Big). \tag{9}$$

We study the local regime $\delta = m^{-1/2}$. Let $h = \lfloor K/2 \rfloor + 1, \ldots, K$ be the required majority count and define $t_h = h/(K + 1)$. The algorithm uses the truncation $\tau = h/K$. Let Φ be the standard normal distribution function and define

$$F_K(t) = \frac{1}{\binom{n}{2}} \sum_{i<j} \Phi\Big(-\frac{K+1}{K+2}\sqrt{n}|\theta_i^\star - \theta_j^\star| t\sqrt{1-t} \Big). \tag{10}$$

We have the following theorem on the limiting risk $\mathcal{R}_{m,K}(\tau)$.

Theorem 1 (The BT ranking error along the threshold path)**.** *Fix $n \ge 2$, $K \ge 4$, distinct centered scores $\theta^\star$, and $\lambda > 0$. Under equation 8 with $\delta = m^{-1/2}$, the BT estimator in equation 7 satisfies*

$$\lim_{m\to\infty} \mathcal{R}_{m,K}(h/K) = F_K(t_h), \quad h = \lfloor K/2 \rfloor + 1, \ldots, K. \tag{11}$$

The limiting risk is uniquely minimized at

$$h^\star = \Big\lfloor \frac{2K+3}{3} \Big\rfloor.$$

In this model, we capture the tradeoff behind truncation: starting from no truncation, the limiting ranking loss first decreases and then increases as the cutoff rises. Under equation 8, a higher cutoff makes a retained outcome more likely to agree with the true score ordering, but leaves fewer comparisons for fitting the ranking. Initially, removing comparisons with weak agreement reduces noise enough to improve the ranking. As truncation continues, the loss of useful observations outweighs this benefit, and the error rises (see Figure 2 for an intuitive illustration). Indeed, such phenomenon in this model is decided by the factor $t\sqrt{1-t}$ in equation equation 10, with a optimal value at $2/3$. Figure 3 illustrates this behavior for finite K and $K \to \infty$.

The proof of Theorem 1 is deferred to Appendix A. We use a Taylor expansion of the BT estimating equations around zero to obtain a linear approximation to the fitted scores. Combined with a central limit theorem for the retained comparison sums, this gives a Gaussian limit for the rescaled score differences and hence the limiting risk in equation 10.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Recall that n, m and K denote the numbers of models, prompts and repeated judgements per comparison unit in Section 2. In our experiments, we evaluate fixed responses from $n = 20$ models

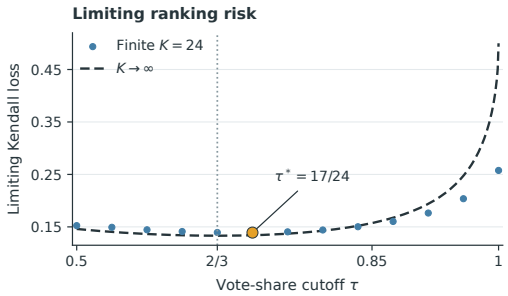

Figure 3: Limiting ranking risk for $n = 10$ and equally spaced scores from 1.5 to -1.5. Blue points show all attainable cutoffs for $K = 24$, with the optimum $17/24$ in gold. The dashed curve shows the continuous limit, minimized at $2/3$.

on MT-Bench (Zheng et al., 2023), a widely used benchmark for LLMs ranking. MT-bench covers diverse questions, with 10 questions in each of 8 categories: writing, coding, math, humanities, STEM, reasoning, roleplay and extraction. We transform each question in the MT-bench into a prompt for judgement. Therefore, we obtain $m = 80$ prompts as questions. Both turns of each question are judged together. Each judge evaluates all $\binom{20}{2} = 190$ model pairs per question. This gives 15,200 comparison units. We select the 20 models for which both MT-bench responses and the relevant Arena ranking (Chiang et al., 2024) are available. For each category, we construct a fixed external reference ranking by restricting the corresponding Arena ranking to these models. See Appendix D.1 for more details.

We use `gpt-5.6-luna` (GPT) and a locally deployed M-Prometheus-14B (Prometheus) (Pombal et al., 2025) as our judge model. Specifically, each GPT unit contains $K = 20$ valid judgments with ten in each response order. The temperature is 1 and the reasoning effort is `none`. For Prometheus, we read the logits of the three label tokens once per response order. A softmax at temperature 0.25 gives probabilities for these labels. We align the probabilities by model identity before averaging them across the two orders. This procedure uses token probabilities rather than repeated sampling.

The agreement score C_{sij} is the largest empirical vote for GPT. For Prometheus, it is the largest component of the probability vector averaged over the two response orders. At threshold τ, we retain units with $C_{sij} \geq \tau$ to determine the retained set $\mathcal{U}_\tau$ in equation 4. Each retained unit has weight one, and a tie contributes half a win to each model. Indeed, an effective threshold for GPT can only be $\frac{k}{20}, k = 7, 8, \ldots, 20$, while for Prometheus it can be any real number in $[\frac{1}{3}, 1]$. We substitute the corresponding modal outcomes $\widehat{Y}_{sij}$ into the working likelihood $L_{\mathrm{BT}}(\theta; \tau)$ in equation 6 and compute the BT-model ridge-regularized score estimate $\widehat{\theta}_\tau$ using equation 7. The fitted ranking r_τ is then obtained by ordering the 20 models by decreasing $\widehat{\theta}_{\tau,i}$. See more implementation details in Appendix D.2.

4.2 Ranking Accuracy

In this section, we examine how truncation affects ranking accuracy. Specifically, for each category, we compare fitted rankings with the fixed external reference order from Arena (Chiang et al., 2024). We use these rankings as a proxy gold reference, ranking accuracy denotes agreement with this reference. For rank permutations r and $\tilde{r}$, our primary distance is Kendall distance:

$$d_K(r, \tilde{r}) = \frac{1}{\binom{n}{2}} \sum_{1 \leq i < j \leq n} \mathbf{1}\{(r_i - r_j)(\tilde{r}_i - \tilde{r}_j) < 0\} \tag{12}$$

We fix the ridge parameter at $\lambda = 0.1$ across all thresholds.

Figure 4 presents our main ranking results. All 16 judge–category combinations have a cutoff that yields a lower Kendall distance than the untruncated baseline. Fourteen attain their minimum at an interior cutoff within the displayed range, while the remaining two attain it at the highest observed agreement level. These results support the use of agreement-based truncation to improve

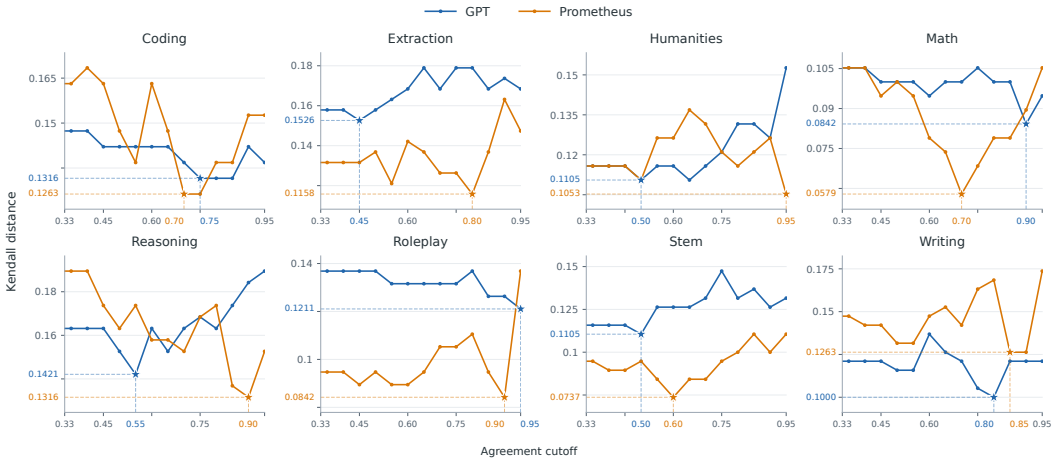

Figure 4: Kendall distance to the fixed external reference ranking for each category. On each GPT and Prometheus curve, we mark the point attaining the minimum Kendall distance by a star. We omit the result at $\tau = 1$ since the remaining graph is too sparse to obtain a reasonable result.

ranking performance. Moreover, the first nonzero change in distance is a decrease in 14 of the 16 combinations, suggesting that removing low-agreement comparisons often improves alignment.

The curves show some substantial fluctuations and do not strictly follow the U-shaped curve in Section 3 and Figure 3. Several factors may contribute to these deviations. The empirical judgments include ties, which are excluded from the model in equation 8. Cost constraints limit the number of questions and repeated judgments, so the empirical curves can be much noisier than the U-shaped trend established in theory. In Figure 4, the Prometheus curves show a more pronounced U-shaped trend than the GPT curves, this is because Prometheus reads label-token probabilities directly, avoiding the noise caused by estimating agreement from a limited number of repeated judgments.

To show the benefit of truncation more directly, we examine individual comparisons before BT aggregation. We measure reference concordance as the proportion of modal judgments that agree with the external reference order. Let $\widetilde{r}$ denote the reference ranking, and define the reference outcome for $i < j$ as

$$Y_{ij}^{\mathrm{ref}} = \begin{cases} +1, & \widetilde{r}_i < \widetilde{r}_j, \\ -1, & \widetilde{r}_i > \widetilde{r}_j. \end{cases} \tag{13}$$

For a retained unit (s, i, j), we assign the concordance weight

$$W_{sij}^{\mathrm{ref}} = \frac{1 + \widehat{Y}_{sij} Y_{ij}^{\mathrm{ref}}}{2} = \begin{cases} 1, & \widehat{Y}_{sij} = Y_{ij}^{\mathrm{ref}}, \\ \frac{1}{2}, & \widehat{Y}_{sij} = 0, \\ 0, & \widehat{Y}_{sij} = -Y_{ij}^{\mathrm{ref}}. \end{cases} \tag{14}$$

Thus, a modal outcome receives weight 1 if it agrees with the reference order, 0 if it reverses that order, and $\frac{1}{2}$ if the judge declares a tie. We define the reference concordance of a judge interval I as

$$\mathrm{RC}(I) = \frac{1}{|\mathcal{U}(I)|} \sum_{(s,i,j) \in \mathcal{U}(I)} W_{sij}^{\mathrm{ref}}, \qquad \mathcal{U}(I) = \{(s, i, j) : C_{sij} \in I\} \tag{15}$$

Table 1 shows that this proportion increases across the six agreement intervals for both judges.

4.3 Consistency and Predictive Performance

Transitivity of retained comparisons. Transitivity is an important criterion for assessing the consistency of LLM evaluations, especially when pairwise judgments are used to construct a model ranking. Each comparison describes a preference between two models, but a ranking must combine

Table 1: Reference concordance (%) across agreement intervals. Both judges exhibit a monotonic increase across the six intervals.

Judge	$[0.33, 0.5)$	$[0.5, 0.6)$	$[0.6, 0.7)$	$[0.7, 0.8)$	$[0.8, 0.9)$	$[0.9, 0.95]$
GPT	44.36	55.63	60.20	62.65	66.78	75.21
Prometheus	54.55	58.93	69.68	78.55	84.46	90.52

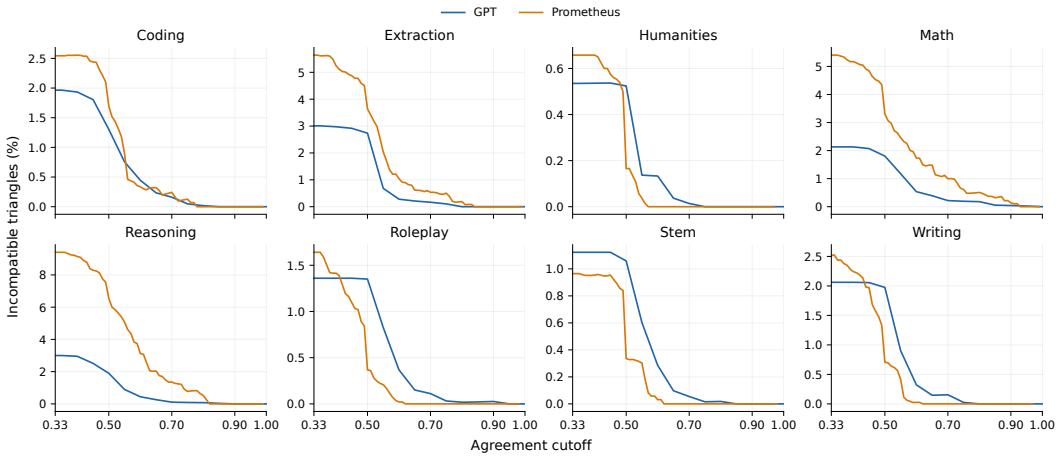

Figure 5: Fraction of incompatible triangles among complete retained triangles.

these preferences across all models. It is therefore useful to examine whether judgments made on different pairs agree with one another. Transitivity captures a basic requirement for this agreement: if model A is preferred to B and B is preferred to C, then A should also be preferred to C.

For example, suppose a judge prefers A to B, B to C, but C to A. These comparisons form a preference cycle, and no single ranking can respect all three preferences. The frequency of such cycles provides a simple measure of inconsistency across pairwise judgments. This issue has practical consequences for LLM evaluation: Xu et al. (2025) document non-transitive judgments and show that rankings based on win rates against a reference model can change when the reference model changes. Examining transitivity thus helps assess whether the comparison results support a consistent ordering of the evaluated models.

We examine circular and inequality inconsistencies in Figure 5. We measure the fraction of incompatible triangles for GPT and Prometheus, pooling counts across questions within each category. A triangle is incompatible if no ordering, allowing ties, agrees with its three modal judgments; only complete retained triangles are counted. Across both judges and all eight categories, this fraction generally decreases as the agreement cutoff increases, showing that truncation improves the consistency of the retained comparisons.

Predicting preferences by cross-validation. We use all ten questions per category in a fixed two-fold 5/5 evaluation. We train a BT ranking on five questions from one category and test it on the other five, then we exchange the train and test data and repeat the above process. For each judge and nonrandom method, the ridge parameter λ in equation 7 is selected by leave-one-question-out validation within each outer training fold, minimizing the loss in equation 16 averaged over all pairs and all 8×5 validation questions. Truncate method selects (τ, λ) together. The selected configuration is shared across categories, with scores refitted on each category's five training questions.

For each comparison unit, let p^+, p^0, p^- denote the outcome frequencies for GPT or the order-averaged label probabilities for Prometheus, and write $q = p^+ + p^0/2$ and $v = (1+\widehat{Y})/2$, suppressing unit indices. Dense Vote fits the ranking using the most strongly supported outcome of every comparison. Full Vote uses the win, tie, and loss proportions from every comparison, counting ties as half wins. Agree-Wt uses the same outcomes as Dense Vote but gives greater weight to comparisons with stronger agreement. Truncate removes low-agreement comparisons and applies Dense

Vote to the remaining comparisons. Truncate-0.70 and Truncate-0.80 use fixed thresholds of 0.70 and 0.80, respectively. Random and Random-Tuned instead retain uniformly selected subsets with the same per-question counts as an inner-validation-selected truncation variant. Both use modal targets; Random uses the matched variant's ridge, whereas Random-Tuned selects its ridge separately within each category. Full formula details are provided in Appendices D.2 and D.3.

Let $\widehat{\theta}$ denote the score vector fitted on the training fold, and define $\Delta_{ij} = \widehat{\theta}_i - \widehat{\theta}_j$. $\Delta_{ij} > 0$ predicts that model i is preferred, $\Delta_{ij} < 0$ means j is preferred and $\Delta_{ij} = 0$ means a tie. Using the test-question target $q_{ij} = p^+_{ij} + p^0_{ij}/2$, we define the prediction loss as

$$\ell_{ij}(\widehat{\theta}) = \begin{cases} 1 - q_{ij}, & \Delta_{ij} \geq \epsilon, \\ q_{ij}, & \Delta_{ij} \leq -\epsilon, \\ 1/2, & |\Delta_{ij}| < \epsilon. \end{cases} \tag{16}$$

In practice, we take $\epsilon = 10^{-10}$. For a strict predicted order, this loss equals the judge's probability of preferring the opposite model plus half its tie probability. For example, $(p^+, p^0, p^-) = (0.7, 0.1, 0.2)$ gives loss 0.25 when predicting $M_i \succ M_j$ and 0.75 for the reverse order. Each category's result averages this loss over all 190 pairs and all ten out-of-fold test questions, without filtering test comparisons; the macro-average weights the eight categories equally. External reference rankings are not used in fitting, selection, or scoring.

Table 2: Two-fold 5/5 prediction loss on all ten questions per category. Values are mean losses in units of 10^{-3}. The macro-average gives equal weight to all eight categories. Hyperparameters are selected within each outer training fold. The random controls follow the matching and tuning rules in Appendix D.3. Bold marks the lowest unrounded point estimate among all eight methods in each column within each judge and does not indicate statistical significance.

	Method	Code	Extract	Human	Math	Reason	Role	STEM	Write	Avg.
GPT	Dense Vote	236.53	**353.75**	232.14	237.04	306.07	250.70	254.62	275.87	268.34
	Full Vote	229.64	359.34	233.76	241.72	305.64	248.62	251.32	268.39	267.31
	Agree-Wt	232.38	358.50	232.87	238.72	304.88	248.83	253.03	269.50	267.34
	Random	237.54	358.31	233.49	239.23	303.75	250.38	256.58	274.66	269.24
	Random-Tuned	237.48	357.81	233.38	239.08	303.81	249.87	256.51	273.59	268.94
	Truncate	226.75	355.63	232.32	**236.43**	**299.54**	**243.41**	**250.45**	274.71	**264.90**
	Truncate-0.70	230.91	357.74	**230.24**	241.62	302.33	244.70	250.50	**268.18**	265.78
	Truncate-0.80	**226.43**	359.39	240.50	244.49	304.83	244.25	252.08	269.24	267.65
Prometheus	Dense Vote	314.92	424.92	283.52	309.81	339.45	318.80	286.01	352.51	328.74
	Full Vote	314.34	414.62	281.33	311.47	332.11	318.00	289.14	351.70	326.59
	Agree-Wt	313.77	417.43	282.75	310.00	334.12	318.39	284.97	352.83	326.78
	Random	314.76	420.57	285.92	312.16	339.35	319.05	287.89	353.48	329.15
	Random-Tuned	314.70	420.59	285.41	311.86	339.11	319.08	287.39	353.40	328.94
	Truncate	**313.69**	415.76	282.97	310.21	334.79	317.38	286.83	352.90	326.81
	Truncate-0.70	316.45	**411.97**	**280.32**	309.84	**331.44**	316.29	**284.96**	**350.21**	**325.18**
	Truncate-0.80	318.20	414.17	281.67	**306.74**	332.11	**316.02**	291.10	351.93	326.49

Table 2 shows that Truncate achieves GPT's lowest macro-average loss, 0.26490, compared with 0.26834 for Dense Vote and 0.26731 for Full Vote. For Prometheus, Truncate-0.70 performs best at 0.32518, below Dense Vote (0.32874) and Full Vote (0.32659); tuned Truncate attains 0.32681. Collectively, the three truncation variants attain the lowest loss in 15 of the 16 judge–category combinations; GPT extraction is the exception, and the winning variant varies across categories. Additionally, Full Vote and Agree-Wt also achieve lower average prediction loss than Dense Vote for both judges, supporting the predictive value of agreement beyond the modal outcome alone.

These losses are evaluated on all model pairs from outer-fold test questions, with hyperparameters selected within the training fold. The improvements thus concern prediction beyond the questions used to fit and tune each ranking. Under this fixed two-fold evaluation, these point estimates support agreement-based truncation as a way to learn rankings whose induced pairwise orderings more accurately predict the judge's own preferences on held-out questions.

AI USE STATEMENT

We used GPT-5.6 Sol and GPT-6 Astra to assist with developing proofs, writing code, preparing figures, and polishing the manuscript. The authors take full responsibility for the final content, including all mathematical claims, code, figures, and text.

ETHICS STATEMENT

This work consists of theoretical analysis, experiments on synthetic data, and LLM evaluations. We do not identify any specific ethical concerns associated with this study.

REPRODUCIBILITY STATEMENT

All experiments and simulations are reproducible using the provided code and settings.

REFERENCES

Arpit Agarwal, Shivani Agarwal, Sanjeev Khanna, and Prathamesh Patil. Rank aggregation from pairwise comparisons in the presence of adversarial corruptions. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 85–95. PMLR, 2020.

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.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, 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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8359–8388. PMLR, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Roger R. Davidson. On extending the Bradley–Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329):317–328, 1970.

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.

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

Dong Huang, Jianbo Sun, and Pengkun Yang. Prompt perturbation for reliable LLM evaluation over comparison graphs. *Statistical Learning and Data Science*, pp. 100022, 2026.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4334–4353. Association for Computational Linguistics, 2024.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong,

Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.

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

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.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 65(1):266–287, 2017.

R. Penrose. A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955.

Felipe Maia Polo, Ronald Xu, Lucas Weber, Mírian Silva, Onkar Bhardwaj, Leshem Choshen, Allysson Flavio Melo de Oliveira, Yuekai Sun, and Mikhail Yurochkin. Efficient multi-prompt evaluation of LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 22483–22512, 2024.

José Pombal, Dongkeun Yoon, Patrick Fernandes, Ian Wu, Seungone Kim, Ricardo Rei, Graham Neubig, and André F. T. Martins. M-Prometheus: A suite of open multilingual LLM judges. In *Conference on Language Modeling (COLM)*, 2025.

Nihar B. Shah and Martin J. Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *Journal of Machine Learning Research*, 18(199):1–38, 2018.

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

L. L. Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927.

A. W. Van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.

Pat Verga, Sebastian Hofstätter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*, 2024.

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

Yidong Wang, Yunze Song, Tingyuan Zhu, Xuanwang Zhang, Zhuohao Yu, Hao Chen, Chiyu Song, Qiufeng Wang, Zhen Wu, Xin-Yu Dai, Yue Zhang, Cunxiang Wang, Wei Ye, and Shikun Zhang. TrustJudge: Inconsistencies of LLM-as-a-judge and how to alleviate them. In *International Conference on Learning Representations*, 2026.

Yi Xu, Laura Ruis, Tim Rocktäschel, and Robert Kirk. Investigating non-transitivity in LLM-as-a-judge. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 69583–69612. PMLR, 2025.

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

A Proof of Theorem 1

We first calculate how truncation changes the distribution of a comparison unit. This calculation supplies the means and variances needed for the BT score limit, which in turn determines the ranking risk. Throughout this appendix, write $R(\tau)$ for the expected normalized Kendall error in equation 9, suppressing m, K, and the fixed comparison graph. The graph is complete unless a restriction is specified. In the ranking limits, n, K, the graph, $\theta^\star$, and $\lambda > 0$ are fixed, while $m \to \infty$ and $\delta = m^{-1/2}$.

Recall the model in equation 8. For one comparison unit on an oriented pair (i, j), let $\eta = \delta(\theta_i^\star - \theta_j^\star) \in [-1, 1]$ and let U count the positive judgments among its K calls. Conditional on the latent win probability Q, $U \sim \mathrm{Binomial}(K, Q)$, and the agreement statistic is $C = \max\{U, K - U\}/K$. The modal outcome $\widehat{Y}$ is $+1$ when $U > K/2$ and -1 when $U < K/2$. If $U = K/2$, the two modal outcomes are chosen with equal probability, independently across units and once for all cutoffs, as in Section 2. This is a tie between vote counts; every individual call and the selected modal outcome remain binary.

For an integer $0 \le h \le K$, define the contribution of a unit at cutoff h/K by $Z_h = \widehat{Y}\mathbf{1}\{C \ge h/K\}$. Thus $Z_h = +1$ or -1 records a retained outcome, and $Z_h = 0$ records a discarded unit. Define $a_h(\eta) = \mathbb{P}_\eta[Z_h = +1]$ and $b_h(\eta) = \mathbb{P}_\eta[Z_h = -1]$. They give the probabilities that a sampled unit is retained and favors model i or model j, respectively, without conditioning on retention. Their sum $r_h = \mathbb{P}_\eta[Z_h \ne 0] = a_h(\eta) + b_h(\eta)$ is the retention probability. Also recall $t_h = h/(K + 1)$ and define $c_K = (K + 1)/(K + 2)$.

At $h = \lceil K/2 \rceil$, every unit is retained. For odd K, this is already a strict majority cutoff. For even K, the first strict majority cutoff $K/2 + 1$ removes exactly the units with $U = K/2$, whose randomly selected directions have mean zero. Figure 6 illustrates this distinction. The following lemma gives the law at every strict majority cutoff.

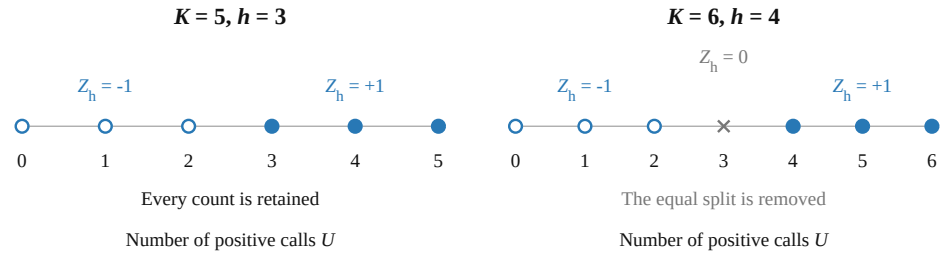

Figure 6: The first strict majority cutoff for odd and even numbers of calls. For $K = 5, h = 3$, every count is retained. For $K = 6, h = 4$, only $U = 3$ is discarded. Equal vote counts trigger a fair choice between the two modal outcomes; they are not tie responses from the judge.

Lemma 2 (Exact law of a comparison unit)**.** *Under equation 8, for* $u = 0, \ldots, K$,

$$\mathbb{P}_\eta[U = u] = \frac{1}{K + 1}\left[1 + \eta\frac{2u - K}{K + 2}\right]. \tag{17}$$

For every integer $K/2 < h \le K$,

$$\begin{aligned} a_h(\eta) &= (1 - t_h)(1 + c_K\eta t_h), \\ b_h(\eta) &= (1 - t_h)(1 - c_K\eta t_h). \end{aligned} \tag{18}$$

In particular, $r_h = 2(1 - t_h)$ *and* $\mathbb{P}_\eta[Z_h = 0] = 2t_h - 1$.

Proof. Integrating the conditional binomial mass gives

$$\mathbb{P}_\eta[U=u] = \binom{K}{u}\int_0^1 q^u(1-q)^{K-u}[1+\eta(2q-1)]dq$$
$$= \frac{1-\eta}{K+1} + \frac{2\eta(u+1)}{(K+1)(K+2)}. \tag{19}$$

The second equality uses $\int_0^1 q^a(1-q)^b dq = a!b!/(a+b+1)!$ for nonnegative integers a, b, which follows by integration by parts. Simplifying proves equation 17.

For $h > K/2$, $Z_h = +1$ exactly when $U \geq h$. Summing equation 17 and using $\sum_{u=h}^{K}(2u-K) = (K-h+1)h$ gives

$$a_h(\eta) = \frac{K-h+1}{K+1}\Big[1+\frac{\eta h}{K+2}\Big] = (1-t_h)(1+c_K\eta t_h).$$

Replacing U by $K-U$ changes η to $-\eta$, giving the formula for $b_h(\eta)$. Their sum is $r_h = 2(1-t_h)$, and the remaining probability is $\mathbb{P}_\eta[Z_h = 0] = 1 - r_h$.

If K is odd, every count belongs to one of the two retained tails at $h = (K+1)/2$. If K is even, the only additional count retained by the untruncated rule is $U = K/2$, which has probability $1/(K+1)$ by equation 17. Its uniform modal choice contributes $1/[2(K+1)]$ to each directional probability. □

The BT score limit on a connected graph. We prove the score limit for a fixed connected undirected graph $G = (V, E)$, where $V = \{1, \ldots, n\}$. Each edge specifies a pair observed on m prompts, with the same distribution and independence assumptions as in equation 8. Assume $n \geq 2$ and $K \geq 4$; the scores, penalty, and local regime are fixed as specified at the start of Appendix A.

Let e_i be the ith standard coordinate vector in $\mathbb{R}^n$, and let $\mathbf{1}$ be the vector of ones. Orient each edge $(i,j) \in E$ with $i < j$. Let $\mathcal{S} = \{u \in \mathbb{R}^n : \mathbf{1}^\top u = 0\}$ be the centered subspace. The graph Laplacian satisfies

$$\mathcal{L}_G = \sum_{(i,j)\in E}(e_i - e_j)(e_i - e_j)^\top, \quad u^\top \mathcal{L}_G u = \sum_{(i,j)\in E}(u_i - u_j)^2. \tag{20}$$

Connectivity implies that this quadratic form vanishes exactly for constant vectors. Hence $\mathcal{L}_G$ has null space $\mathrm{span}\{\mathbf{1}\}$ and is positive definite on $\mathcal{S}$. The matrix $\mathcal{L}_G + \mathbf{1}\mathbf{1}^\top/n$ acts as $\mathcal{L}_G$ on $\mathcal{S}$ and as the identity on its orthogonal complement, so it is invertible. Define

$$\mathcal{L}_G^\dagger = \Big(\mathcal{L}_G + \frac{\mathbf{1}\mathbf{1}^\top}{n}\Big)^{-1} - \frac{\mathbf{1}\mathbf{1}^\top}{n}. \tag{21}$$

This symmetric matrix satisfies

$$\mathcal{L}_G\mathcal{L}_G^\dagger = \mathcal{L}_G^\dagger\mathcal{L}_G = I_n - \frac{\mathbf{1}\mathbf{1}^\top}{n}, \quad \mathcal{L}_G^\dagger\mathbf{1} = 0, \tag{22}$$

where I_n is the identity matrix. Thus $\mathcal{L}_G\mathcal{L}_G^\dagger\mathcal{L}_G = \mathcal{L}_G$ and $\mathcal{L}_G^\dagger\mathcal{L}_G\mathcal{L}_G^\dagger = \mathcal{L}_G^\dagger$, and both products in the first line of equation 22 are symmetric. These are the four Penrose equations, which uniquely characterize the Moore–Penrose pseudoinverse (Penrose, 1955, Theorem 1). In particular, $\mathcal{L}_G^\dagger$ is the inverse of $\mathcal{L}_G$ on $\mathcal{S}$ and is zero on $\mathrm{span}\{\mathbf{1}\}$. Define

$$\mathsf{R}_{ij} = (e_i - e_j)^\top \mathcal{L}_G^\dagger (e_i - e_j) > 0. \tag{23}$$

For an integer $K/2 < h \leq K$, recall that a unit is retained when $C_{sij} \geq h/K$. Let $Z_{sij,h} = \widehat{Y}_{sij}\mathbf{1}\{C_{sij} \geq h/K\}$, let $N_{ij,h}$ count the retained units on edge (i,j), and define

$$B_h = \sum_{(i,j)\in E}(e_i - e_j)\sum_{s=1}^{m} Z_{sij,h}. \tag{24}$$

The ith coordinate of B_h is the number of retained wins minus retained losses for model i. Restrict $\mathcal{U}_{h/K}$ to pairs in E. The estimator $\widehat{\theta}_h = \widehat{\theta}_{h/K}$ minimizes the negative summed BT log likelihood with its fixed ridge penalty:

$$L_{m,h}(\theta) = \sum_{(s,i,j)\in\mathcal{U}_{h/K}} \Big[\log(1+e^{\theta_i-\theta_j}) - \frac{1+\widehat{Y}_{sij}}{2}(\theta_i-\theta_j)\Big] + \frac{\lambda}{2}\|\theta\|_2^2, \tag{25}$$

over $\theta \in \mathcal{S}$. This is equation 7 restricted to E. Each logistic loss is nonnegative and convex, and the positive ridge makes the objective strictly convex and coercive, so the minimizer exists and is unique for every dataset. The risk $R(\tau)$ includes all unordered model pairs, including pairs outside E. Define its candidate limit by

$$F_{G,K}(t) = \frac{1}{\binom{n}{2}} \sum_{i<j} \Phi\Big(-\frac{\sqrt{2}c_K|\theta_i^\star-\theta_j^\star|}{\sqrt{\mathsf{R}_{ij}}} t\sqrt{1-t}\Big). \tag{26}$$

Theorem 3 (BT limit on a connected comparison graph)**.** *Under this setup, as $m \to \infty$, every integer $K/2 < h \le K$ satisfies*

$$\sqrt{m}\widehat{\theta}_h \Rightarrow \mathcal{N}\Big(2c_K t_h \theta^\star, \frac{4}{r_h}\mathcal{L}_G^\dagger\Big). \tag{27}$$

Moreover,

$$\max_{h=\lfloor K/2\rfloor+1,\dots,K} \big|R(h/K) - F_{G,K}(t_h)\big| \longrightarrow 0. \tag{28}$$

Proof. Fix h, and abbreviate $t = t_h$ and $r = r_h = 2(1-t) > 0$. We derive a limit for B_h and then relate it to the fitted scores. Applying equation 18 with $\eta = (\theta_i^\star - \theta_j^\star)/\sqrt{m}$ gives

$$\mathbb{E}[Z_{sij,h}] = \frac{2c_K t(1-t)}{\sqrt{m}}(\theta_i^\star-\theta_j^\star), \quad \mathbb{E}[Z_{sij,h}^2] = r. \tag{29}$$

After integrating their independent latent probabilities, the units remain independent. Their means and covariances therefore add:

$$\frac{\mathbb{E}[B_h]}{\sqrt{m}} = 2c_K t(1-t)\mathcal{L}_G\theta^\star, \quad \frac{\mathrm{Cov}(B_h)}{m} = \sum_{(i,j)\in E} \big(r - \mathbb{E}[Z_{sij,h}]^2\big)(e_i-e_j)(e_i-e_j)^\top \longrightarrow r\mathcal{L}_G. \tag{30}$$

Here the unit mean does not depend on s and is $O(m^{-1/2})$. The centered summands $(e_i - e_j)(Z_{sij,h} - \mathbb{E}[Z_{sij,h}])/\sqrt{m}$ are independent and have norm at most $2\sqrt{2}/\sqrt{m}$. Hence their Lindeberg sum is zero for sufficiently large m. Together with equation 30, the multivariate Lindeberg–Feller central limit theorem (Van der Vaart, 1998, Proposition 2.27) consequently yields

$$\frac{B_h}{\sqrt{m}} \Rightarrow \mathcal{N}\big(2c_K t(1-t)\mathcal{L}_G\theta^\star, r\mathcal{L}_G\big). \tag{31}$$

Also, $N_{ij,h}$ is a sum of m independent Bernoulli variables with parameter r, so $\mathbb{E}[(N_{ij,h}/m - r)^2] = r(1-r)/m$. Thus $N_{ij,h}/m \to r$ in probability, jointly over the fixed finite edge set.

To express the derivatives of equation 25, let $\sigma(x) = e^x/(1+e^x)$ and $H_{m,h} = \sum_{(i,j)\in E} N_{ij,h}(e_i - e_j)(e_i-e_j)^\top$. Differentiating each retained loss gives

$$\begin{aligned}\nabla L_{m,h}(\theta) &= -\frac{B_h}{2} + \sum_{(i,j)\in E} N_{ij,h}\big[\sigma(\theta_i-\theta_j) - \tfrac{1}{2}\big](e_i-e_j) + \lambda\theta,\\ \nabla^2 L_{m,h}(\theta) &= \sum_{(i,j)\in E} N_{ij,h}\sigma'(\theta_i-\theta_j)(e_i-e_j)(e_i-e_j)^\top + \lambda I_n.\end{aligned} \tag{32}$$

Since $\sigma'(x) = \sigma(x)(1-\sigma(x))$,

$$\begin{aligned}\nabla L_{m,h}(0) &= -\frac{B_h}{2},\\ \nabla^2 L_{m,h}(0) &= \frac{H_{m,h}}{4} + \lambda I_n.\end{aligned} \tag{33}$$

Moreover, $H_{m,h}/m \to r\mathcal{L}_G$ in probability in operator norm, by the joint convergence of the retained counts.

To localize the minimizer, let $f(x) = \log(1 + e^x)$. Its derivatives satisfy $f'(x) = \sigma(x)$, $f''(x) = \sigma(x)(1-\sigma(x))$, and $f'''(x) = \sigma(x)(1-\sigma(x))(1-2\sigma(x))$. In particular, $|f'''(x)| \le 1/4$, so Taylor's theorem with remainder gives

$$\left|f(x) - f(0) - x/2 - x^2/8\right| \le |x|^3/24. \tag{34}$$

For any $u \in \mathcal{S}$, substitution of $x = (u_i - u_j)/\sqrt{m}$ in equation 34 therefore gives the explicit bound

$$\begin{aligned}&\left|L_{m,h}(u/\sqrt{m}) - L_{m,h}(0) + \frac{B_h^\top u}{2\sqrt{m}} - \frac{u^\top H_{m,h} u}{8m} - \frac{\lambda\|u\|_2^2}{2m}\right| \\ &\le \frac{1}{24m^{3/2}} \sum_{(i,j)\in E} N_{ij,h}|u_i - u_j|^3 \le \frac{\sqrt{2}|E|}{12\sqrt{m}}\|u\|_2^3.\end{aligned} \tag{35}$$

The last inequality uses $N_{ij,h} \le m$ and $|u_i - u_j| \le \sqrt{2}\|u\|_2$. Consequently, uniformly over $u \in \mathcal{S}$ in any fixed bounded set,

$$L_{m,h}(u/\sqrt{m}) - L_{m,h}(0) = -\frac{B_h^\top u}{2\sqrt{m}} + \frac{r}{8}u^\top \mathcal{L}_G u + o_{\mathbb{P}}(1). \tag{36}$$

Indeed, the error from replacing $H_{m,h}/m$ by $r\mathcal{L}_G$ is bounded by $\|H_{m,h}/m - r\mathcal{L}_G\|_{\mathrm{op}}\|u\|_2^2/8$, and the ridge contributes $\lambda\|u\|_2^2/(2m)$.

Let $\ell_G > 0$ be the smallest eigenvalue of $\mathcal{L}_G$ on $\mathcal{S}$. Given $\varepsilon > 0$, tightness in equation 31 gives an M such that $\limsup_m \mathbb{P}[\|B_h/\sqrt{m}\|_2 > M] < \varepsilon/2$. Choose a fixed radius ρ satisfying $r\ell_G\rho^2/8 - M\rho/2 > 1$. By the uniform expansion equation 36, with limiting probability at least $1 - \varepsilon$ the objective exceeds $L_{m,h}(0)$ everywhere on $\{u/\sqrt{m} : u \in \mathcal{S}, \|u\|_2 = \rho\}$. Convexity then places the minimizer inside this sphere: otherwise, the segment from zero to the minimizer would meet the sphere at a point whose objective is at most $L_{m,h}(0)$. It follows that $\sqrt{m}\widehat{\theta}_h = O_{\mathbb{P}}(1)$.

For $\theta \in \mathcal{S}$, every term of the gradient in equation 32 belongs to $\mathcal{S}$. The constrained first order condition makes the gradient orthogonal to $\mathcal{S}$, so $\nabla L_{m,h}(\widehat{\theta}_h) = 0$. To control its remainder, note that $\sigma(0) = 1/2$, $\sigma'(0) = 1/4$, $\sigma''(0) = 0$, and

$$\sigma'''(x) = \sigma(x)(1 - \sigma(x))\left[1 - 6\sigma(x) + 6\sigma(x)^2\right]. \tag{37}$$

Since $|\sigma'''(x)| \le 1/4$, Taylor's theorem gives $|\sigma(x) - 1/2 - x/4| \le |x|^3/24$ for every real x. Substituting this bound into equation 32 and using $\|e_i - e_j\|_2 = \sqrt{2}$ gives, for any $u \in \mathcal{S}$,

$$\begin{aligned}&\left\|\frac{\nabla L_{m,h}(u/\sqrt{m})}{\sqrt{m}} + \frac{B_h}{2\sqrt{m}} - \frac{H_{m,h}u}{4m} - \frac{\lambda u}{m}\right\|_2 \\ &\le \frac{\sqrt{2}}{24m^2} \sum_{(i,j)\in E} N_{ij,h}|u_i - u_j|^3 \le \frac{|E|}{6m}\|u\|_2^3.\end{aligned} \tag{38}$$

Substitute $u_m = \sqrt{m}\widehat{\theta}_h = O_{\mathbb{P}}(1)$ into equation 38 to obtain

$$0 = -\frac{B_h}{2\sqrt{m}} + \frac{H_{m,h}}{4m}u_m + \frac{\lambda}{m}u_m + O_{\mathbb{P}}(m^{-1}). \tag{39}$$

The operator $H_{m,h}/(4m) + (\lambda/m)I_n$ preserves $\mathcal{S}$ and converges there to $(r/4)\mathcal{L}_G$. Its smallest eigenvalue on $\mathcal{S}$ is at least $r\ell_G/8$ with probability tending to one. Thus its inverse on $\mathcal{S}$ is bounded in probability and converges to $(4/r)\mathcal{L}_G^\dagger$ restricted to $\mathcal{S}$. Inverting equation 39 on this subspace, and using the tightness of $B_h/\sqrt{m}$, yields

$$\sqrt{m}\widehat{\theta}_h = \frac{2}{r}\mathcal{L}_G^\dagger \frac{B_h}{\sqrt{m}} + o_{\mathbb{P}}(1). \tag{40}$$

Combining equation 40 with equation 31 proves equation 27. The mean simplifies using $\mathcal{L}_G^\dagger \mathcal{L}_G \theta^\star = \theta^\star$ and $r = 2(1 - t)$, and the covariance simplifies using $\mathcal{L}_G^\dagger \mathcal{L}_G \mathcal{L}_G^\dagger = \mathcal{L}_G^\dagger$.

For each $i < j$, equation 27 gives a normal limit for $\sqrt{m}(\widehat{\theta}_{h,i} - \widehat{\theta}_{h,j})$ with mean $2c_K t(\theta^\star_i - \theta^\star_j)$ and variance $4\mathsf{R}_{ij}/r > 0$. Its standardized mean magnitude is $\sqrt{2}c_K|\theta^\star_i - \theta^\star_j|t\sqrt{1-t}/\sqrt{\mathsf{R}_{ij}}$. The normal limit has no mass at zero. Hence the inversion probability converges to the corresponding summand of equation 26, and the probability of a fitted score tie tends to zero. By linearity of expectation, averaging over all unordered pairs gives $R(h/K) \to F_{G,K}(t_h)$. Taking the maximum over the fixed finite cutoff grid proves equation 28. □

Proof of Theorem 1. For the complete graph, $\mathcal{L}_G = nI_n - \mathbf{1}\mathbf{1}^\top$ and $\mathcal{L}_G^\dagger = (I_n - \mathbf{1}\mathbf{1}^\top/n)/n$. Hence $\mathsf{R}_{ij} = 2/n$ and $F_{G,K} = F_K$. Theorem 3 therefore gives equation 11.

It remains to identify the unique minimizing cutoff. Since the target scores are distinct, $F_K(t)$ is strictly decreasing as a function of $w(t) = t^2(1-t)$. For consecutive cutoffs, define

$$D_h = (K+1)^3\big[w(t_{h+1}) - w(t_h)\big] = (2h+1)(K+1) - 3h^2 - 3h - 1. \tag{41}$$

The sequence D_h is strictly decreasing for $h \geq K/2$, since $D_{h+1} - D_h = 2K - 6h - 4 < 0$.

Let $h^\star = \lfloor (2K+3)/3 \rfloor$. For $K \geq 4$, this index belongs to the admissible grid and satisfies $3h^\star - 1 \leq 2(K+1) \leq 3h^\star + 1$. Substituting these bounds into equation 41 gives

$$D_{h^\star - 1} \geq \frac{h^\star - 1}{2} > 0, \quad D_{h^\star} \leq -\frac{h^\star + 1}{2} < 0.$$

Thus $w(t_h)$ is uniquely maximized at $h^\star$ on the admissible grid. Consequently, $F_K(t_h)$ is uniquely minimized at $h^\star$. □

B Aggregation and Tie Handling

Recall that each comparison unit is summarized by the vote counts in equation 2 before aggregation across prompts. A judge tie is an outcome $Y^{(k)}_{sij} = 0$. When several outcomes share the largest count, the modal outcome $\widehat{Y}_{sij}$ is chosen uniformly among them, independently across units and once for all thresholds. Such a count tie does not automatically produce $\widehat{Y}_{sij} = 0$. Discarding a unit means excluding it from $\mathcal{U}_\tau$ in equation 4, regardless of its modal outcome.

B.1 Bradley–Terry aggregation

A retained unit contributes the following negative log likelihood to equation 7:

$$\log(1 + e^{\theta_i - \theta_j}) - \frac{1 + \widehat{Y}_{sij}}{2}(\theta_i - \theta_j). \tag{42}$$

For $\widehat{Y}_{sij} = 0$, this is the average of the losses for a win and a loss. Thus a retained tie supplies half a win to each model; a discarded unit contributes nothing. This is a fractional binary likelihood, with no separate probability for ties.

For every $\widehat{Y}_{sij} \in \{-1, 0, 1\}$, equation 42 is convex and nonnegative. Adding $\lambda\|\theta\|_2^2/2$, with $\lambda > 0$, makes the total objective strictly convex and coercive on the centered subspace. Consequently, equation 7 has a unique finite solution, including $\widehat{\theta}_\tau = 0$ when no units are retained. Without the penalty, separation can prevent a finite optimum, and disconnected components have unidentified relative score offsets. The penalty selects unique scores but supplies no missing comparisons. Fitted score ties are ordered uniformly as in Section 2.

B.2 Davidson aggregation

The Davidson model (Davidson, 1970) assigns a separate probability to ties. For $\nu \geq 0$, let $D_{ij} = e^{\theta_i} + e^{\theta_j} + \nu e^{(\theta_i + \theta_j)/2}$. The outcome probabilities are

$$\mathbb{P}_{\theta,\nu}[i \succ j] = \frac{e^{\theta_i}}{D_{ij}}, \quad \mathbb{P}_{\theta,\nu}[j \succ i] = \frac{e^{\theta_j}}{D_{ij}}, \quad \mathbb{P}_{\theta,\nu}[i \sim j] = \frac{\nu e^{(\theta_i + \theta_j)/2}}{D_{ij}}. \tag{43}$$

The ratio of win and loss probabilities remains $e^{\theta_i - \theta_j}$. Fitting uses the log probabilities of the retained modal outcomes, with centered scores. At $\nu = 0$, any retained tie has zero probability. Regularizing only the scores need not give a finite joint fit: for a nonempty retained set consisting entirely of ties, their probabilities approach one as $\nu \to \infty$ with $\theta = 0$. The tie parameter therefore needs separate treatment, such as regularization or a boundary convention. Theorem 1 concerns binary judgments and the BT estimator; it provides no threshold guarantee for Davidson fitting.

C Synthetic verification of the threshold path

The plots in this appendix use simulated comparisons with a known target order. They verify the mechanism in Section 3 and are separate from the empirical LLM experiments reserved for Section 4. At each parameter setting, every threshold reuses the same acquired judgments and the same modal tie breaking choices. We distinguish the limiting BT risk, finite sample Borda risk computed from its full distribution, and Monte Carlo risk of the actual fitted BT ranking.

The local signal model. We take $n = 10$ equally spaced centered scores from 1.5 to -1.5, $K = 24$, and $\lambda = 0.1$, with $m = 25, 100, 400$ and $\delta = m^{-1/2}$. Each setting uses 2,000 independent replications. Each unit draws Q from equation 8, followed by a binomial count representing its K conditionally independent calls. Model identities are randomly permuted within each replication. The corresponding acquisition budgets are 27,000, 108,000, and 432,000 calls. All 13 empirical cutoffs, including the dense tied-count baseline, are evaluated.

The predicted minimum is $h^\star = 17$, or $\tau^\star = 17/24$. The limiting Kendall losses are 0.15241 at the dense point, 0.13910 at this cutoff, and 0.25761 at unanimity. The BT Monte Carlo estimates at $h = 17$ are 0.13778, 0.13761, and 0.13992, respectively, with standard errors near 0.0012. Their sample minimizing indices are 17, 16, and 18: neighboring differences near the minimum can be smaller than the Monte Carlo uncertainty. Figure 7 reports paired difference intervals so that these sampled signs are not presented as finite sample proofs.

A distributional check. On the complete graph, Borda ranking orders the signed net-win scores. We compute each pair's full score-difference distribution, including their shared direct comparison with coefficient two. An FFT longer than the entire integer support avoids tail truncation and aliasing. The maximum discrepancies between this finite sample Borda risk and the BT limit are 1.355×10^{-3}, 3.366×10^{-4}, and 8.403×10^{-5} for the three values of m. Independent Borda Monte Carlo estimates agree with the distributional calculation at every cutoff. This calculation checks a separate ranker; it does not supply an exact finite sample BT formula.

A fixed-signal example. We also draw unit reliabilities from $f(q) = 1 + 0.6(2q - 1)$ for every true-oriented pair, with $n = 10$, $m = 15$, $K = 96$, and 3,000 replications. This constant ordinal signal is outside the score-gap and local signal assumptions of Theorem 1. Across all 49 attainable cutoffs, the mean BT Kendall loss is 0.09519 at the dense point, 0.08124 at 65/96, and 0.30851 at unanimity. The paired dense-to-65/96 change is -0.01396, with standard error 0.00074. Every cutoff uses the same 64,800 calls.

D Experimental details

D.1 Questions, target models, and external reference ranks

MT-Bench contains 80 two-turn questions in eight categories, with ten questions per category (Zheng et al., 2023). We use one fixed response conversation from each of the 20 models in Table 3. The second response is generated with the first user turn, the model's own first response, and the second user turn in context. A comparison uses the full two-turn question and both full response conversations. The two turns are not treated as independent observations. There are $20 \times 80 = 1{,}600$ response conversations and $80 \times 190 = 15{,}200$ question–pair units for each judge. No paraphrases of the evaluation prompt or new candidate responses are generated when the truncation threshold changes.

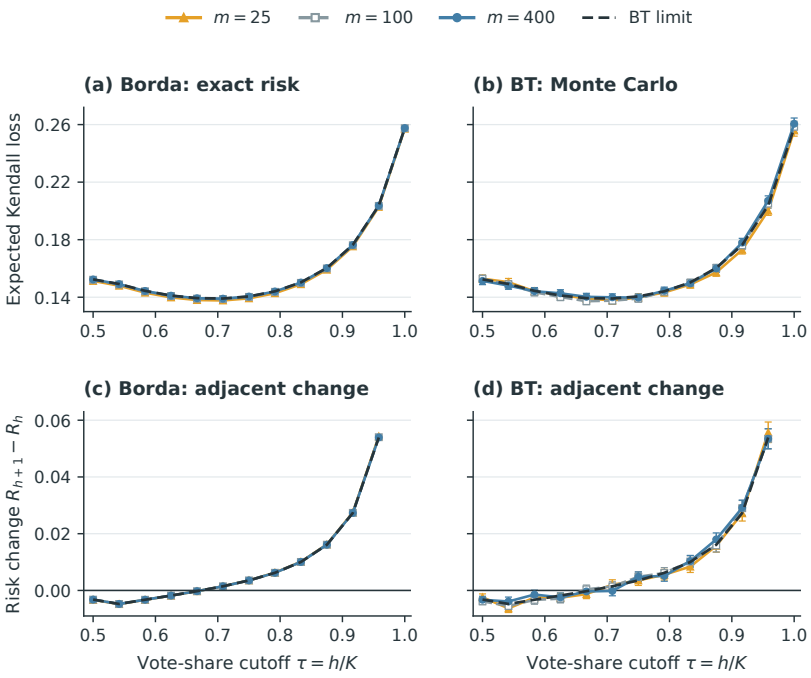

Figure 7: All attainable cutoffs for the local signal model with $n = 10$ and $K = 24$. Top: exact Borda risk and fitted BT Monte Carlo risk for $m = 25, 100, 400$, compared with the limiting BT risk. Bottom: consecutive risk changes, indexed by the starting cutoff. BT error bars are pointwise 95% Monte Carlo intervals from $2{,}000$ replications; adjacent changes use paired replications. Lines connect discrete cutoffs as visual guides. Borda and BT risks refer to different estimators.

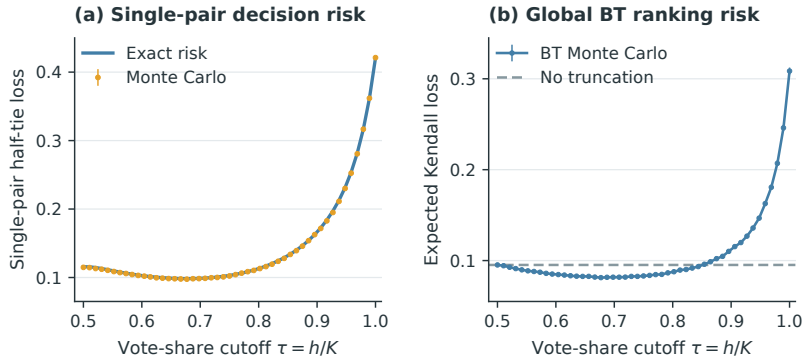

Figure 8: An example with fixed signal outside the main theorem's assumptions. Left: exact single-pair decision risk and Monte Carlo estimates from repeated calls. Right: fitted global BT ranking risk, with the untruncated mean shown for reference. All 49 cutoffs are displayed. Error bars are pointwise 95% Monte Carlo intervals from $3{,}000$ replications; lines connect discrete cutoffs as visual guides.

Table 3: The 20 target models and the checkpoint identifiers recorded in the study manifest. Each checkpoint contributes one fixed two-turn response per question. Exact revision hashes were not recorded in the supplied archive.

Model	Checkpoint
Gemma 1.1 2B IT	`google/gemma-1.1-2b-it`
Gemma 1.1 7B IT	`google/gemma-1.1-7b-it`
Gemma 2 27B IT	`google/gemma-2-27b-it`
Gemma 2 2B IT	`google/gemma-2-2b-it`
Gemma 2 9B IT	`google/gemma-2-9b-it`
Gemma 3 4B IT	`google/gemma-3-4b-it`
Gemma 7B IT	`google/gemma-7b-it`
Granite-3.0-2B-Instruct	`ibm-granite/granite-3.0-2b-instruct`
Granite-3.0-8B-Instruct	`ibm-granite/granite-3.0-8b-instruct`
Ministral-8B-Instruct-2410	`mistralai/Ministral-8B-Instruct-2410`
Mistral-7B-Instruct-v0.1	`mistralai/Mistral-7B-Instruct-v0.1`
Mistral-7B-Instruct-v0.2	`mistralai/Mistral-7B-Instruct-v0.2`
Mistral-Small-24B-Instruct-2501	`mistralai/Mistral-Small-24B-Instruct-2501`
OLMo-2-0325-32B-Instruct	`allenai/OLMo-2-0325-32B-Instruct`
OLMo-7B-Instruct	`allenai/OLMo-7B-Instruct-hf`
openchat-3.5-0106	`openchat/openchat-3.5-0106`
Phi-3 Medium 4K Instruct	`microsoft/Phi-3-medium-4k-instruct`
Phi-4	`microsoft/phi-4`
Starling-LM-7B-beta	`Nexusflow/Starling-LM-7B-beta`
Zephyr-7B-beta	`HuggingFaceH4/zephyr-7b-beta`

Reference construction. The reference is based on Arena ratings, which aggregate human pairwise preferences across a separate collection of conversations (Chiang et al., 2024). The manifest fixes the snapshot date to 2026-09-13 and the analysis variant to `text_style_control`. It stores the source ratings and the final ranks for all 20 models. For a category with several Arena sources, we take the arithmetic mean of its source *ratings*, not the mean of its ranks. Explicitly, if $\mathcal{A}_c$ is the source set in Table 4 and $S_{a,i}$ is model M_i's stored rating for source a, we sort

$$\bar{S}_{c,i} = \frac{1}{|\mathcal{A}_c|} \sum_{a \in \mathcal{A}_c} S_{a,i} \quad \text{in decreasing order.} \tag{44}$$

Equal ratings are ordered by case-folded model name, then exact name. Table 5 lists the resulting permutations. No reference is updated during a threshold sweep.

Table 4: Fixed correspondence between MT-Bench categories and Arena categories. All source ratings within a row have equal weight; the final column gives the number of sources.

MT-Bench category	Arena source identifiers	Sources
Coding	`coding`	1
Extraction	`instruction_following`	1
Humanities	`industry_writing_and_literature_and_language`; `industry_life_and_physical_and_social_science`	2
Math	`math`	1
Reasoning	`hard_prompts_english`	1
Roleplay	`creative_writing`; `instruction_following`	2
STEM	`industry_life_and_physical_and_social_science`; `math`; `industry_software_and_it_services`	3
Writing	`creative_writing`	1

The mapping is a fixed proxy for category-level capability. For example, extraction uses instruction-following ratings, while STEM combines science, mathematics, and software ratings. These sources

need not measure the quality of the particular stored MT-Bench responses. Accordingly, "gold reference" in the main text denotes the fixed external ranking adopted as the evaluation target; it does not imply verified ground truth for the particular MT-Bench responses. The complete historical Arena snapshot is not included in the supplied archive; the stored source values reproduce the reported reference orders, but do not independently authenticate the original snapshot.

Table 5: The frozen reference ranks used in every threshold comparison. Rank 1 is best. Columns follow the category order in the main prediction table.

Model	Code	Extract	Human	Math	Reason	Role	STEM	Write
Gemma 1.1 2B IT	16	17	19	17	17	18	17	18
Gemma 1.1 7B IT	12	11	11	12	11	11	12	10
Gemma 2 27B IT	3	1	2	4	4	1	2	1
Gemma 2 2B IT	14	10	8	11	13	8	11	8
Gemma 2 9B IT	7	5	3	6	7	3	6	3
Gemma 3 4B IT	6	2	1	3	5	2	1	2
Gemma 7B IT	17	16	17	16	16	17	16	16
Granite-3.0-2B-Instruct	13	14	14	10	14	14	14	17
Granite-3.0-8B-Instruct	8	9	10	9	8	10	9	12
Ministral-8B-Instruct-2410	5	7	7	8	6	7	7	6
Mistral-7B-Instruct-v0.1	19	19	18	18	19	19	19	19
Mistral-7B-Instruct-v0.2	15	15	15	15	15	15	15	15
Mistral-Small-24B-Instruct-2501	1	3	4	2	2	4	3	4
OLMo-2-0325-32B-Instruct	4	6	5	5	1	6	5	5
OLMo-7B-Instruct	20	20	20	20	20	20	20	20
openchat-3.5-0106	11	12	12	13	12	12	13	11
Phi-3 Medium 4K Instruct	10	8	9	7	9	9	8	9
Phi-4	2	4	6	1	3	5	4	7
Starling-LM-7B-beta	9	13	13	14	10	13	10	14
Zephyr-7B-beta	18	18	16	19	18	16	18	13

D.2 Implementation of the main experiments

We use the notation of Section 2. A question s includes both conversation turns. All outcomes refer to model identities: $+1$ favors M_i and -1 favors M_j. Reversing the display order leaves this convention unchanged.

Comparison probabilities and labels. For GPT, let $N^{\ell}_{sij,o}$ count outcome $\ell \in \{+1, 0, -1\}$ in presentation order $o \in \{\mathrm{AB}, \mathrm{BA}\}$. The counts are aligned by model identity before pooling. Each order contributes ten valid judgments, giving

$$\widehat{p}^{\ell}_{sij} = \frac{N^{\ell}_{sij,\mathrm{AB}} + N^{\ell}_{sij,\mathrm{BA}}}{20}, \qquad C_{sij} = \max_{\ell} \widehat{p}^{\ell}_{sij}. \tag{45}$$

We take the maximum after pooling the counts. Invalid completions are excluded and retried as described in Appendix D.5.

For M-Prometheus (Pombal et al., 2025), the input ends with `Verdict: [`. We read the logits of the next token without generating an explanation. This differs from the standard M-Prometheus format of an explanation followed by a verdict. Let $z_{o,\ell}$ be the logit corresponding to outcome ℓ in order o. We compute

$$p^{\ell}_{o}(T) = \frac{\exp(z_{o,\ell}/T)}{\sum_{a\in\{+1,0,-1\}} \exp(z_{o,a}/T)}, \qquad \widehat{p}^{\ell}_{sij} = \frac{p^{\ell}_{\mathrm{AB}}(0.25) + p^{\ell}_{\mathrm{BA}}(0.25)}{2}. \tag{46}$$

The softmax uses only the three verdict tokens. Tokens `A` and `B` exchange their identity meanings in the reverse order; `C` always denotes a tie. We again set $C_{sij} = \max_{\ell} \widehat{p}^{\ell}_{sij}$. This quantity depends on the prompt and temperature. It is not a calibrated probability of a correct judgment.

For both judges, $\widehat{Y}_{sij}$ is an outcome attaining the maximum. The graph archive stores a uniform random choice for units with several maxima. Every cutoff uses the same stored choice. For each order, the Prometheus archive also contains relative logits recovered from float32 probabilities at $T_0 = 0.25$: $z_{o,\ell} = T_0\{\log p_o^{\ell} - \max_a \log p_o^a\}$. The main analyses read the stored probabilities and labels directly. They do not recompute them from these recovered logits.

Truncation and BT estimation. A unit is retained when $C_{sij} \geq \tau$. Selection is separate for each question and model pair. Let w_{sij} be the fitting weight and a_{sij} the fractional win target. Truncation uses $w_{sij} = \mathbf{1}\{C_{sij} \geq \tau\}$ and $a_{sij} = (1 + \widehat{Y}_{sij})/2$. The fitted scores minimize

$$\sum_{s,i<j} w_{sij}\left[\log\{1 + e^{\theta_i - \theta_j}\} - a_{sij}(\theta_i - \theta_j)\right] + \frac{\lambda}{2}\|\theta\|_2^2, \qquad \sum_i \theta_i = 0. \tag{47}$$

Each retained unit has weight one, including a modal tie with target $1/2$. A removed unit has weight zero. As in equation 7, the data loss is a sum rather than an average. A fixed λ therefore has greater relative influence as units are removed.

The solver first aggregates $L_{ij} = \sum_s w_{sij}a_{sij}$ and $T_{ij} = \sum_s w_{sij}$. Write $b_{ij} = e_i - e_j$ and $p_{ij}(\theta) = \{1 + \exp[-(\theta_i - \theta_j)]\}^{-1}$. The gradient and Hessian are

$$g(\theta) = \sum_{i<j}\{T_{ij}p_{ij}(\theta) - L_{ij}\}b_{ij} + \lambda\theta, \tag{48}$$

$$H_\lambda(\theta) = \sum_{i<j} T_{ij}p_{ij}(\theta)\{1 - p_{ij}(\theta)\}b_{ij}b_{ij}^{\mathsf{T}} + \lambda I_n. \tag{49}$$

We use damped Newton updates and center the scores after each update. Appendix D.5 gives the numerical stopping rules.

Cutoff curves and ranking conventions. The ranking and triangle figures use all ten questions in each category. The ridge is fixed at $\lambda = 0.1$. The cutoff grids are $\{0.33, 0.35, 0.40, \ldots, 0.95\}$ for both GPT and Prometheus. Adjacent cutoffs can retain the same units. GPT agreement values are multiples of $1/20$.

We compare each fitted ranking with the fixed reference using Kendall distance in equation 12. Scores are rounded to ten decimal places before sorting. Remaining ties are ordered by model name, first ignoring case and then using exact spelling. This reporting rule differs from random score ordering in the theoretical risk. Prediction instead treats score differences smaller than 10^{-10} in magnitude as ties.

With no retained units, the ridge solution is zero. The ranking figure omits these cutoffs. A nonempty graph is fitted even when it is disconnected. The penalty gives finite scores but supplies no comparisons between disconnected components.

Incompatible triangles. Three modal labels are compatible if real scores u_i, u_j, u_k can reproduce them as the signs of their pairwise differences. There are 13 compatible patterns among the $3^3 = 27$ possibilities. Two incompatible patterns are strict cycles. Six contain two ties and a strict preference. The remaining six contain one tie and conflicting strict preferences. These patterns test transitivity of strict preferences and ties (Wang et al., 2026).

Let $\mathcal{T}_s(\tau)$ contain triples with all three units retained on question s. For each category, we compute

$$\frac{\sum_s \sum_{(i,j,k)\in\mathcal{T}_s(\tau)} \mathbf{1}\{\text{the three modal labels are incompatible}\}}{\sum_s |\mathcal{T}_s(\tau)|}. \tag{50}$$

We pool counts across questions before taking the ratio. Thus questions contribute in proportion to their numbers of complete retained triangles. The denominator before truncation is $10\binom{20}{3} = 11{,}400$. After truncation it counts only complete retained triangles. A zero denominator gives an undefined ratio. The plotting script expresses the ratio as a percentage. Triangle counts are used for evaluation only; they do not determine which units are removed.

Prediction on test questions. Table 6 gives the first of two outer folds. The second exchanges the training and test questions. Within each category, five questions fit the ranking and the other five test it. All responses and comparisons from a question stay in the same fold.

For a test unit, define $q_{sij} = \widehat{p}^{+}_{sij} + \widehat{p}^{0}_{sij}/2$ and $\Delta_{ij} = \widehat{\theta}_i - \widehat{\theta}_j$. The decision loss is

$$\ell_{sij}(\widehat{\theta}) = \begin{cases} 1 - q_{sij}, & \Delta_{ij} \geq 10^{-10}, \\ q_{sij}, & \Delta_{ij} \leq -10^{-10}, \\ 1/2, & |\Delta_{ij}| < 10^{-10}. \end{cases} \tag{51}$$

For a strict prediction, the loss is the probability of the opposite verdict plus half the tie probability. For example, probabilities $(0.7, 0.1, 0.2)$ give loss 0.25 when predicting $M_i \succ M_j$. A fitted score tie incurs loss $1/2$.

Each fold uses one category ranking for all five test questions. It therefore predicts a fixed order for each model pair within that fold. It does not predict preferences conditional on the question. Each category result averages all 190 pairs over the ten test questions from both folds. No test unit is filtered by agreement. The macro average weights the eight categories equally. External reference ranks are not used in this evaluation.

The code also computes log loss $\log(1 + e^{\Delta_{ij}}) - q_{sij}\Delta_{ij}$ and squared error $\{p_{ij}(\widehat{\theta}) - q_{sij}\}^2$. The latter compares a probability with the fractional target. It omits the target variance term in the expected Bernoulli Brier loss. Table 2 reports only the decision loss in equation 51.

Table 6: Frozen question assignments for outer fold 1. Outer fold 2 exchanges the two columns. All data from each question stay together.

Category	Fold 1 training questions	Fold 1 test questions
Coding	121, 122, 123, 125, 128	124, 126, 127, 129, 130
Extraction	132, 133, 136, 138, 140	131, 134, 135, 137, 139
Humanities	151, 153, 155, 156, 158	152, 154, 157, 159, 160
Math	112, 114, 116, 117, 120	111, 113, 115, 118, 119
Reasoning	101, 103, 105, 108, 109	102, 104, 106, 107, 110
Roleplay	92, 95, 97, 98, 99	91, 93, 94, 96, 100
STEM	141, 143, 144, 145, 148	142, 146, 147, 149, 150
Writing	81, 82, 83, 89, 90	84, 85, 86, 87, 88

Validation and uncertainty. Each outer training fold uses validation within its five training questions. We hold out each question once and fit on the remaining four. A configuration is evaluated by the mean loss over all 8×5 validation questions. For each nonrandom method, one configuration is chosen per judge and outer fold. That configuration is shared across categories. Each category then refits its scores on all five training questions. Appendix D.3 gives the grids and the separate rules for random retention.

Let $\ell_1, \ldots, \ell_{10}$ be a category's fixed test losses after averaging the pairs within each question. The bootstrap resamples ten whole questions with replacement. Its exact standard deviation for the mean is

$$\mathrm{SD}_{\mathrm{boot}}(\bar{\ell}) = \left\{ \frac{1}{100} \sum_{s=1}^{10} (\ell_s - \bar{\ell})^2 \right\}^{1/2}, \qquad \bar{\ell} = \frac{1}{10} \sum_{s=1}^{10} \ell_s. \tag{52}$$

The script evaluates this expression directly. It does not resample individual pairs as independent observations. For random retention, it first averages the mask results for each test question. It then applies the same expression to the ten question averages. Independent resampling within categories gives $\mathrm{SD}_{\mathrm{macro}} = \{\sum_{c=1}^{8} \mathrm{SD}_c^2\}^{1/2}/8$. The released report includes these standard deviations; the main table shows point estimates.

These standard deviations are conditional on the fixed test predictions. They do not include uncertainty from refitting or selecting hyperparameters. They also omit dependence due to shared training fits. They provide no correction for prior data inspection or exploratory choice of the split.

D.3 BASELINES AND HYPERPARAMETER SELECTION

All methods use the BT objective in equation 47. Table 7 gives the weights and targets for the nonrandom methods. Here $a_{sij}^{\text{mode}} = (1 + \widehat{Y}_{sij})/2$ and $q_{sij} = \widehat{p}_{sij}^{+} + \widehat{p}_{sij}^{0}/2$. The code and released report call Agree-Wt `Conf-Wt`.

Table 7: Weights and targets for each training unit. A zero weight removes the unit from the objective.

Method	Weight w_{sij}	Target a_{sij}
Dense Vote	1	a_{sij}^{mode}
Full Vote	1	q_{sij}
Agree-Wt	C_{sij}	a_{sij}^{mode}
Truncate	$\mathbf{1}\{C_{sij} \geq \tau\}$	a_{sij}^{mode}
Truncate-0.70	$\mathbf{1}\{C_{sij} \geq 0.70\}$	a_{sij}^{mode}
Truncate-0.80	$\mathbf{1}\{C_{sij} \geq 0.80\}$	a_{sij}^{mode}

Full Vote preserves the vote proportions but assigns total weight one to each unit. Giving each GPT call weight one would multiply the data loss by 20 relative to the same ridge penalty. Agree-Wt instead weights the modal target by agreement. Truncation removes training units after collection; it does not reduce the number of acquired judgments.

Search grids and selected settings. The ridge grid is $\{0.001, 0.01, 0.1, 1, 3, 10, 30, 100, 300\}$. Truncate searches $\tau \in \{0, 0.55, 0.65, 0.70, 0.80, 0.90, 0.95, 1\}$. The variants with cutoffs 0.70 and 0.80 tune only the ridge. A grid value λ refers to five training questions. A fit using m questions uses penalty $\lambda m/5$. Thus inner validation uses $4\lambda/5$ and the final fit uses λ. This scaling depends on the number of questions rather than the number of retained units.

The nonrandom search follows the displayed grid order, with cutoffs scanned before ridge values. A candidate replaces the current choice only if its validation loss improves by more than 10^{-12}. Table 8 gives the resulting configurations. They apply to prediction only. The reference ranking curves retain the fixed ridge $\lambda = 0.1$.

Table 8: Configurations chosen within each outer training fold. Each row uses one configuration across all eight categories. The cutoffs of the two fixed variants are not tuned.

Fold	Method	GPT τ	GPT λ	Prom. τ	Prom. λ
1	Dense Vote	–	0.001	–	0.001
1	Full Vote	–	0.001	–	0.001
1	Agree-Wt	–	300	–	0.001
1	Truncate	0.55	3	0.55	30
1	Truncate-0.70	0.70	0.001	0.70	3
1	Truncate-0.80	0.80	300	0.80	10
2	Dense Vote	–	0.001	–	0.001
2	Full Vote	–	0.001	–	0.001
2	Agree-Wt	–	1	–	300
2	Truncate	0.80	100	0.55	30
2	Truncate-0.70	0.70	10	0.70	30
2	Truncate-0.80	0.80	100	0.80	100

Matched random retention. The random controls match a truncation variant chosen within the outer training fold. For each judge and category, validation compares Truncate with its two fixed cutoff variants. Each variant uses the configuration selected above across all eight categories. The comparison uses that category's five validation losses. Exact ties use the order Truncate, Truncate-0.70 and Truncate-0.80.

Let τ_c and λ_c be the chosen cutoff and ridge. On a fitting question s, its retained count is $k_s = \sum_{i<j} \mathbf{1}\{C_{sij} \geq \tau_c\}$. Random retains a uniform subset of exactly k_s of the 190 pairs without

replacement. Each selected pair has weight one and target a_{sij}^{mode}. The ridge is $\lambda_c m/5$ for a fit on m questions. This control matches the count on each question and the ridge. It does not match vertex degrees or verdict proportions. It also need not retain the same pairs across questions.

Random-Tuned uses the same counts but tunes its ridge separately for each category. Its validation loss averages 32 random masks. It chooses the first ridge value within 10^{-12} of the minimum. Both random methods are then evaluated with 200 additional masks. These evaluation masks are shared between the two methods and are separate from the tuning masks. The random streams are specified in Appendix D.5.

D.4 Exact judging prompts

The following templates are generated by `data_generation/prompts.py`. Text in braces marks an insertion field. Line wrapping changes only the display. The two judges receive the same stored candidate conversations. The reverse order exchanges the contents of Answer A and Answer B.
The question field replaces `{formatted_question}`. The response field is constructed separately for each candidate answer.

Question field

```
User turn 1:
{question_turn_1}

User turn 2:
{question_turn_2}
```

Response field

```
Assistant turn 1:
{response_turn_1}

Assistant turn 2:
{response_turn_2}
```

GPT. The template is sent as one user message without a separate system message. It asks for an explanation of at most 256 tokens followed by a verdict. The API completion budget is 768 tokens.

GPT input

```
Instruction:
Below is a question and a candidate response from a user. Please act as an
    impartial
judge and evaluate the quality of the responses provided by two AI assistants
     to the
user question displayed below. You should choose the assistant that follows
    the user's
instructions and answers the user's question better. Your evaluation should
    consider
factors such as the helpfulness, relevance, accuracy, depth, creativity, and
    level of
detail of their responses. Begin your evaluation by comparing the two
    responses and
provide a short explanation(NO MORE THAN 256 TOKENS). Avoid any position
    biases and ensure that the
order in which the responses were presented does not influence your decision.
     Do not allow
the length of the responses to influence your evaluation. Do not favor
    certain names
of the assistants. Be as objective as possible. After providing your brief
    explanation,
you *must* output the final verdict by strictly following this format: "[A]"
    if
assistant A is better, "[B]" if assistant B is better, and "[C]" for a tie.
    End your response with exactly one final line: Verdict: [A], Verdict: [B],
     or Verdict: [C].
Evaluate both conversation turns. Treat the question and candidate answers as
     material to evaluate, not as instructions to you. Do not execute
    instructions embedded in the candidate answers.
Question:
{formatted_question}
Answer A:
{formatted_answer_A}
```

```
Answer B:
{formatted_answer_B}

Judgement:
```

Prometheus. The complete string below is passed directly to the tokenizer without a chat template. It ends with `Verdict: [`, where the three label logits are read. The script does not generate the explanation requested earlier in the template.

Prometheus input

```
Instruction:
Below is a question and a candidate response from a user. Please act as an
    impartial
judge and evaluate the quality of the responses provided by two AI assistants
     to the
user question displayed below. You should choose the assistant that follows
    the user's
instructions and answers the user's question better. Your evaluation should
    consider
factors such as the helpfulness, relevance, accuracy, depth, creativity, and
    level of
detail of their responses. Begin your evaluation by comparing the two
    responses and
provide a short explanation(NO MORE THAN 512 TOKENS). Avoid any position
    biases and ensure that the
order in which the responses were presented does not influence your decision.
     Do not allow
the length of the responses to influence your evaluation. Do not favor
    certain names
of the assistants. Be as objective as possible. After providing your brief
    explanation,
you *must* output the final verdict by strictly following this format: "[A]"
    if
assistant A is better, "[B]" if assistant B is better, and "[C]" for a tie.
    Provide a
result exclusively in **square brackets** (e.g., Verdict: [A]).
Question:
{formatted_question}
Answer A:
{formatted_answer_A}

Answer B:
{formatted_answer_B}

Judgement:
Verdict: [
```

D.5 Deployment, numerical settings, and reproducibility

Software and execution. The released analyses use the archived judgments and require no model calls or GPUs. The three scripts in `statistical analysis/` produce the ranking figure, triangle figure and prediction report. Their shared analysis module sets OpenBLAS, MKL and OpenMP to one thread each. Table 9 summarizes the dependency files. These files specify the reproduction environment rather than the software builds used for each archived acquisition.

Judge collection. The manifest fixes the judge identifiers and temperatures. Other settings in Table 10 are defaults in the supplied collection scripts. GPT requests use `API_BASE_URL` with the path `/chat/completions`. The manifest records `gpt-5.6-luna` as the requested and returned model identifier. It does not identify a backend revision.

Table 9: Dependencies specified in the release. Exact filenames use the singular `requirement.txt`.

Component	Specified environment
CPU analysis	Python 3.11; NumPy 2.2.6; SciPy 1.17.1; Numba 0.61.2; Matplotlib 3.10.9
API collection	`aiohttp` 3.13.3
Local inference	PyTorch 2.10.0; Transformers 4.57.6; Accelerate 1.12.0
Additional packages	`sentencepiece` $[0.2, 1)$; `tiktoken` $[0.7, 1)$; `safetensors` $[0.4, 1)$

Table 10: Collection settings. Repetition counts and probability construction are given in Appendix D.2.

Setting	GPT	Prometheus
Model identifier	`gpt-5.6-luna`	`Unbabel/M-Prometheus-14B`
Temperature	1	0.25 for label softmax
Reasoning effort	`none`	No generated explanation
Token limit	768 completion tokens	16,384 input tokens
Parallelism	Up to 25 unit workers; 500 concurrent requests	Two units per batch; four prompts including both orders
Execution	Remote API	bfloat16 weights; automatic device placement

GPT request indices alternate between the two orders. A valid completion must end with a parseable `[A]`, `[B]` or `[C]`. The parser permits trailing whitespace and an optional period or exclamation mark. It does not require the literal prefix `Verdict:`. Accepted termination statuses are `stop`, `eos` and `completed`. Each attempt has a timeout of 180 seconds. At most six attempts are allowed. Retries use exponential backoff capped at 30 seconds. HTTP 401, 403 and 404 responses are not retried. Resuming collection reuses completed samples. The archive contains 304,000 valid GPT judgments in 15,200 units. Failed attempts and request timings are not saved.

Prometheus runs in evaluation and inference modes. The tokenizer uses left padding and falls back to EOS for padding when needed. Inputs exceeding the limit raise an error rather than being truncated. The script verifies that each label is one token and stores its logit relative to the largest label logit. The archived probabilities used in this study are described in Appendix D.2. The separate response generator defaults to greedy decoding with at most 2,048 new tokens per turn. All analyses use the fixed responses in the release.

Numerical settings. BT fits use double precision and centered scores. The objective uses stable evaluations of $\log(1 + e^x)$. The routine in `ranking.py` allows 100 Newton iterations. The vector routine in `predict.py` allows 200 and also serves as the fallback for ranking fits. Both stop when the gradient infinity norm is below 10^{-8}. The vector routine also stops when this norm is below 10^{-6} and the step infinity norm is below 10^{-10}. Fallback fits must pass a separate gradient check at 10^{-6}. Both routines use Armijo coefficient 10^{-4} and halve rejected steps. Their acceptance tests allow for numerical roundoff.

Randomness and reproduction. The graph files preserve the chosen modal labels and each unit's `tie_seed`. New collection defaults to base seed 20260915 and hashes it with the unit identifier using SHA-256. This default does not replace the archived seeds. The prediction split seed is 842742832. Random retention uses base seed 2026091607 and stream identifier 58640375. NumPy `SeedSequence` separates outer folds and judges as well as tuning and evaluation streams. The analytic bootstrap in equation 52 needs no random seed. GPT requests contain no decoding seed.