Dropping Just a Handful of Preferences Can Change Top Large Language Model Rankings

Jenny Y. Huang [*1 2] **Yunyi Shen** [*1 2] **Dennis Wei** [2] **Tamara Broderick** [1 2]

Abstract

We propose a method for evaluating the robustness of a widely used LLM ranking system—the Bradley–Terry ranking system—to dropping a worst-case very small fraction of evaluation data. Our approach is computationally fast and easy to adopt. When we apply our method to matchups from two popular human-preference platforms, Chatbot Arena and MT-Bench, we find that the Bradley–Terry rankings of top-performing models are remarkably sensitive to the removal of a small fraction of evaluations. Our framework also identifies the specific evaluations most responsible for such ranking flips, allowing for inspections of these influential preferences. We observe that the rankings derived from MT-Bench preferences are notably more robust than those from Chatbot Arena, likely due to MT-bench's use of expert annotators and carefully constructed prompts. Finally, we find that rankings based on crowdsourced human-evaluated systems are just as sensitive as those based on LLM-as-a-judge evaluations, where in both, dropping as little as 0.02% of the total evaluations in the dataset can change the top-ranked model.

1. Introduction

Open evaluation platforms like Chatbot Arena have, in large part due to their openness, become a gold standard for assessing the capabilities of leading LLMs via human preference. These open platforms are now widely used by top LLM developers and companies to evaluate and design new models and benchmarks (Chiang et al., 2024a; Singh et al., 2025; Grattafiori et al., 2024; Hui et al., 2024; White et al., 2024).

[*]Equal contribution [1]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA [2]MIT-IBM Watson AI Lab, Cambridge, MA, USA. Correspondence to: Jenny Y. Huang <jhuang9@mit.edu>.

ICML 2025 Workshop on Models of Human Feedback for AI Alignment, Vancouver, Canada. PMLR 267, 2025.

Such platforms rely on crowdsourced pairwise battles and human votes to compute model rankings (Lee et al., 2023; Bai et al., 2022). At the heart of these preference-based evaluation and alignment pipelines is the Bradley–Terry (BT) model (Bradley & Terry, 1952; Sun et al., 2025), which is widely used to rank LLMs based on human feedback (Chiang et al., 2024a), train reward models for RLHF (Ouyang et al., 2022; Touvron et al., 2023; Xu et al., 2024), and route incoming queries to the most appropriate LLM or inference-time scaling strategy (Damani et al., 2024).

Recent controversies, however, have raised concerns about the trustworthiness of popular LLM evaluation platforms like Chatbot Arena (Chiang et al., 2024a). For instance, Min et al. (2025) demonstrate that Chatbot Arena is vulnerable to vote-rigging: by injecting just a few hundred manipulated votes (out of 1.7 million), attackers can significantly change the top model rankings. Singh et al. (2025) further identify issues such as data leakage and private testing practices that allow large, proprietary model developers to selectively report the best-performing versions of their models.

We study a slightly different type of untrustworthiness of LLM-based ranking systems in this work. That is: *"Will the top rankings from LLM-evaluation platforms change upon dropping a very small fraction of the human (or AI) preference evaluations?"* If it turns out that the answer is yes, this raises concerns about the generalizability and stability of the rankings produced by such systems. One might worry about whether these rankings, the learned human preferences from these systems, actually generalize (Broderick et al., 2020).

This question motivates the need for a systematic way to assess the robustness of top rankings in BT-based evaluation systems to worst-case data dropping. However, no such method currently exists (beyond a brute-force combinatorial search over all possible small subsets of data) to test whether the model rankings on LLM evaluations systems are robust to the removal of a very small fraction of adversarially-chosen evaluations.[1]

In order to avoid this computationally intractable combinatorial search, we turn to a recent line of works from statistics

[1]This combinatorial search procedure is computationally infeasible for large-scale platforms like Chatbot Arena.

and theoretical computer science that design algorithms for assessing whether data analyses are robust to dropping a small, worst-case fraction of data points (Broderick et al., 2020; Kuschnig et al., 2021; Moitra & Rohatgi, 2022; Freund & Hopkins, 2023; Shiffman et al., 2023; Nguyen et al., 2024; Huang et al., 2024; Rubinstein & Hopkins, 2025). One such method, the Approximate Maximum Influence Perturbation (AMIP), estimates how much a statistic of interest could change if a worst-case subset of the data were dropped (Broderick et al., 2020). We extend these ideas to develop a fast approximation method for assessing the robustness of rankings from LLM evaluation systems to worst-case data-dropping. We apply our method to two widely used sets of human preference matchups: Chatbot Arena and MT-Bench (Chiang et al., 2024a; Zheng et al., 2023). Although our methods are largely based on the ranking system used in Chatbot Arena and MT-Bench, there are a few differences (including tie-handling and post-processing) between the raw BT scores we analyze and the scores implemented on these platforms; see Section 3.1 for more details. We note that a previous work in computational biology (Shiffman et al., 2023) studies the sensitivity of gene set enrichment analyses to dropping just a few cells and that, importantly, this work studies the same form of worst-case data-dropping robustness of rankings. However, while this previous work studies robustness in the ranking of p-values (specifically, p-values derived using the hypergeometric test), our work studies the robustness the rankings derived from the BT model (which can be thought of as elements in the coefficient vector of a logistic regression model).

In Section 2, we formalize the setup for assessing worst-case data-dropping robustness in BT-based ranking systems, and in Section 3 we introduce a computationally efficient method for assessing this form of robustness in practice. In Section 4, we apply our robustness assessment method to investigate the robustness of LLM preference data.

Our code is publicly available at `https://github.com/JennyHuang19/IsRankingRobust`, including all scripts required to run our robustness assessing method and to reproduce the results presented in this paper.

2. Setup

Human preference data. We consider a preference-based ranking system akin to Chatbot Arena (Chiang et al., 2024a). There are in total M language models. Any user can submit a prompt to be answered by a pair of language models, say model i and j for $i, j \in [M]$ with $i \neq j$. The user then determines if the response from model i is better than that of model j. Suppose there are in total N such comparisons; the data can be seen as a collection of tuples $\{(i_n, j_n, y_n)\}_{n=1}^{N}$, with $y_n \in \{0, 1\}$ being an indicator for whether in the nth match, model i_n is preferred over model j_n. From a collection of preference data, the goal is to rank the language models.

Ranking with the Bradley–Terry model. The Bradley–Terry (BT) model is a classical statistical model used to rank players from binary preference data that is used by Chatbot Arena. In this model, each player (e.g., language model), i, is associated with a *BT score*, θ_i, and the preferences are assumed to be generated as

$$y_n \sim \text{Bernoulli}(\sigma(\theta_{i_n} - \theta_{j_n})). \tag{1}$$

Note, since the "winning" probability depends on the difference between two players rather than on their raw scores, the scores are identified only up to a constant additive term. There are different ways to avoid this identifiability problem (Wu et al., 2022). We adopted the method of setting up the first player's score to be 0. We can cast this model as a logistic regression with a specially-structured design matrix. We denote the corresponding "design" vector of the nth comparison, $x_n \in \{-1, 0, 1\}^M$, a vector encoding which two players are being compared. That is, if the game is between players i and j, then x_n has a 1 in the ith element, a -1 in the jth element, and 0 otherwise. Using this structure, we can rewrite the model as a logistic regression model with $M - 1$ parameters corresponding to the scores of the players, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M) \in \mathbb{R}^M$ with $\theta_1 = 0$,

$$y_n \sim \text{Bernoulli}(\sigma(x_n^\top \boldsymbol{\theta})). \tag{2}$$

We fit the BT-model (i.e., estimate $\boldsymbol{\theta}$) by maximum likelihood,

$$\begin{aligned}\hat{\boldsymbol{\theta}} := \underset{\boldsymbol{\theta}:\theta_1=0}{\arg\min} \sum_{n=1}^{N} \big(&- y_n \log \sigma(x_n^\top \boldsymbol{\theta}) \\ &- (1 - y_n) \log(1 - \sigma(x_n^\top \boldsymbol{\theta}))\big).\end{aligned} \tag{3}$$

Finally, we define the *rank* of a model as its position in the sorted list of models, $(\widehat{\theta}_{(1)}, \ldots, \widehat{\theta}_{(M)})$, ordered by their scores in descending order, so that $\widehat{\theta}_{(1)}$ corresponds to the top-ranked model.

In the rest of this section, we turn to the goal of determining whether there exists a small fraction of data (e.g., matchups) that we can drop to change the ordering of the estimated BT scores.

Setup for Data-Dropping. We study whether dropping a small fraction $\alpha \in (0, 1)$ (e.g., $\alpha = 0.01$) of the preference data can change the ordering of the estimated BT scores. Broderick et al. (2020) define the *Maximum Influence Perturbation* as the largest possible change induced in a quantity of interest by removing at most $100\alpha\%$ of the data.

Let w_n denote a weight on the nth data point, and collect these into a vector $w := (w_1, ..., w_N)$. Define the weighted estimator as

$$\hat{\boldsymbol{\theta}}(w) := \underset{\boldsymbol{\theta}(w):\theta_1(w)=0}{\arg\min} \sum_{n=1}^{N} w_n\big(- y_n \log\sigma(x_n^\top \boldsymbol{\theta}) - (1-y_n)\log(1-\sigma(x_n^\top \boldsymbol{\theta}))\big). \tag{4}$$

Setting $w = 1_N$ (the all-ones vector) recovers the BT scores computed on the full data (e.g., the original arena), while setting $w_n = 0$ corresponds to dropping the nth data point (e.g., a matchup). We define the set of all weight vectors corresponding to dropping at most an α-fraction of the data as follows:

Definition 2.1 (Feasible Drop Set)**.** Let $W_\alpha := \{w \in \{0,1\}^N : \sum_{n=1}^N (1-w_n) \leq \alpha N\}$ be the set of all binary weight vectors indicating subsets where at most $100\alpha\%$ of the data has been dropped.

Two-Player Arena. We begin by analyzing the robustness of an arena involving just two players (e.g., LLMs): player i and player j. Without loss of generality, we assume[2] that player i has the higher estimated BT score on the full data:

$$\widehat{\theta}_i(1_N) \geq \widehat{\theta}_j(1_N).$$

We are interested in whether this ordering can be reversed by dropping at most an α-fraction of the data.

Definition 2.2 (Top-1 Data-Dropping Robustness in Two–Player Arenas)**.** An arena consisting of players i and j is said to be *top-1 robust at level* α if there does *not* exist a data weighting $w \in W_\alpha$ such that the BT scores reverse under reweighting:

$$\left\{ w \in W_\alpha : \widehat{\theta}_i(w) < \widehat{\theta}_j(w) \right\} = \emptyset. \tag{5}$$

To generalize this setup beyond a two-player arena, we introduce more notation.

M-Player Arena. We now extend this notion to arenas with M players, for any $M \geq 2$. Let $\mathcal{T}(w) := \{\widehat{\theta}_i(w)\}_{i=1}^M$ denote the set of BT scores under weighting w.

Definition 2.3 (Top-k Set)**.** The *top-k set* under full data is defined as the set of players whose scores rank among the top k:

$$\mathcal{K}_{\mathcal{T}(1_N)} := \left\{ \widehat{\theta}_i(1_N) : \text{rank}\left[\widehat{\theta}_i(1_N); \mathcal{T}(1_N)\right] \leq k \right\}. \tag{6}$$

Definition 2.4 (Top-k Data-Dropping Robustness in M-Player Arenas)**.** An arena is *top-k robust at level* α if no α-fraction subset of data can be dropped to change the top-k set. That is,

$$\left\{ w \in W_\alpha : \mathcal{K}_{\mathcal{T}(1_N)} \neq \mathcal{K}_{\mathcal{T}(w)} \right\} = \emptyset. \tag{7}$$

[2]If this assumption does not hold, the identities of i and j can be swapped.

Equation (7) is nontrivial to directly verify; to directly check, we have to perform two combinatorial operations, namely (1) recalculate these sets and (2) test out dropping all possible small-fraction subsets of the arena.

In Section 3, we show that verifying whether Equation (7) holds can be reduced to checking the robustness of a series of pairwise comparisons. Specifically, top-k robustness as defined in Definition 2.4 can be checked by checking whether there exists a reweighting $w \in W_\alpha$ such that the ranking of any pair (i, j) such that i is inside and j is outside the top-k set flips. We then can test if such flipping can happen by a continuous approximation of the discrete weight w. We detail this procedure in Section 3.

3. Proposed method

Recall that our goal is to evaluate the robustness of the rankings induced by a BT-model $\{\widehat{\theta}_{(1)}, ..., \widehat{\theta}_{(M)}\}$ when a small fraction of matches (e.g., evaluations) is removed from the arena. To this end, we introduce a method based on checking the robustness of pairwise BT score differences.

We begin by showing that a top-k set can be characterized by considering a set of pairwise comparisons in Proposition 3.1.

Proposition 3.1. *Suppose we have M distinct real numbers, $\mathcal{T}(w) := \{\widehat{\theta}_i(w)\}_{i=1}^M$. Suppose a set $\mathcal{S} \subset \mathcal{T}(w)$ satisfies $|\mathcal{S}| = k$. Suppose it is the case that $\forall\ \widehat{\theta}_i(w) \in \mathcal{S}$ and $\forall\ \widehat{\theta}_j(w) \in \mathcal{T}(w) \setminus \mathcal{S}$, we have that $\widehat{\theta}_i(w) > \widehat{\theta}_j(w)$. Then, it must be that $\mathcal{S}$ is the top-k set, or $\mathcal{S} = \mathcal{K}_{\mathcal{T}(w)}$.*

Proof. We first show that $\mathcal{S} \subset \mathcal{K}_{\mathcal{T}(w)}$. Suppose that $\widehat{\theta}_i(w) \in \mathcal{S}$. By assumption, we have that $\forall\ \widehat{\theta}_j(w) \in \mathcal{T}(w)\backslash\mathcal{S}, \widehat{\theta}_i(w) > \widehat{\theta}_j(w)$. Since $|\mathcal{T}(w)\backslash\mathcal{S}| = M-k$, there must exist at least $(M-k)$ values in $\mathcal{T}(w)$ that are smaller than $\widehat{\theta}_i(w)$. This must mean that $\text{rank}(\widehat{\theta}_i(w); \mathcal{T}(w)) \leq k$, so $\widehat{\theta}_i(w) \in \mathcal{K}_{\mathcal{T}(w)}$ as needed.

We next show that $\mathcal{K}_{\mathcal{T}(w)} \subset \mathcal{S}$ by contradiction. Suppose there exist a $\widehat{\theta}_j(w)$ such that $\widehat{\theta}_j(w) \in \mathcal{K}_{\mathcal{T}(w)}$ but $\widehat{\theta}_j(w) \notin \mathcal{S}$. Since $\widehat{\theta}_j(w) \notin \mathcal{S}$, then $\widehat{\theta}_j(w) \in \mathcal{T}(w) \setminus \mathcal{S}$. This means that $\forall \widehat{\theta}_i(w) \in \mathcal{S}$ we have $\widehat{\theta}_i(w) > \widehat{\theta}_j(w)$, and since $|\mathcal{S}| = k$, this implies that $\text{rank}(\widehat{\theta}_j(w); \mathcal{T}(w)) > k$, contradicting the assumption $\widehat{\theta}_j(w) \in \mathcal{K}_{\mathcal{T}(w)}$. ◻

Using the result from Proposition 3.1, we come up with a greedy algorithm to test whether the top-k set is robust to worst-case data-dropping.

The idea here is to test the data-dropping robustness of all players in the top-k set against all players outside of the top-k set. If any one of these pairwise comparisons is non-robust, then the top-k set is non-robust.

Before that, we describe what it means for a given pair of player scores, $(\widehat{\theta}_i(w), \widehat{\theta}_j(w))$, to be data-dropping robust.

Pairwise Robust. Given a pair of teams, (i, j), we say that the scores for this pair, $(\widehat{\theta}_i(w), \widehat{\theta}_j(w))$, are robust to small-fraction data-dropping at level-α if

$$\{w \in W_\alpha : \widehat{\theta}_i(w) \leq \widehat{\theta}_j(w)\} = \emptyset. \tag{8}$$

Top-k Robust. Recall that an arena is top-k robust at level-α if there does not exist a reweighting, $w \in W_\alpha$, such that $\mathcal{K}_{\mathcal{T}(1_N)} \neq \mathcal{K}_{\mathcal{T}(w)}$ (see Equation (7)). Using the line of logic in Proposition 3.1, this is equivalent to showing that, $\forall\ (i, j)$ where $i \in \mathcal{K}_{\mathcal{T}(w)}$ and $j \notin \mathcal{K}_{\mathcal{T}(w)}$, the pair $(\widehat{\theta}_i(w), \widehat{\theta}_j(w))$ is robust. Namely, if every comparison (i, j) in this set of pairwise comparisons stays the same (after reweighting), then the top-k set also stays the same (see Proposition 3.1 for a detailed proof).

We now provide a method for checking the robustness of pairwise comparisons.[3]

Method for Checking Pairwise Robustness. In Equation (8), we are interested in checking whether there exists a small fraction of evaluations, $w \in W_\alpha$, that can be dropped to change the sign of a difference in BT scores. Without loss of generality, we will assume that the sign of the difference of BT scores fit to the full data is positive (e.g., $[\widehat{\theta}_i(1_N) - \widehat{\theta}_j(1_N)] > 0$, meaning that model i has a higher score than model j).

To evaluate the robustness of the sign of $[\widehat{\theta}_i(1_N) - \widehat{\theta}_j(1_N)]$ to dropping a small fraction of matches, we adopt a recently-developed method from the statistics literature known as the *Approximate Maximum Influence Perturbation* (Broderick et al., 2020) (see Appendix A for a more detailed discussion on how we adapt this method to our problem setup). This method approximates the maximal directional change in a statistic, e.g., $[\widehat{\theta}_i(1_N) - \widehat{\theta}_j(1_N)]$, that can result from dropping a worst-case subset of data points (in our case, evaluations) of size at most $\lfloor \alpha N \rfloor$. This method allows us to sidestep running an expensive combinatorial search over all data-subsets for the worst-case subset of matches to drop, a procedure that is computationally prohibitive for large LLM evaluation platforms like Chatbot Arena.

As the name suggests, AMIP approximates the *Maximum Influence Perturbation*, or the maximal change to a quantity-of-interest (e.g., $[\widehat{\theta}_i(1_N) - \widehat{\theta}_j(1_N)]$) that can be induced by dropping a small fraction of data points from a dataset (i.e., by performing the data analysis using a reweighting of the data in which the weight of a small subset of points is set to 0, $w \in W_\alpha$).

The optimization problem implied by the Maximum Influence Perturbation problem in our particular case is shown below

$$\max_{w \in W_\alpha} \left(\left[\widehat{\theta}_i(1_N) - \widehat{\theta}_j(1_N)\right] - \left[\widehat{\theta}_i(w) - \widehat{\theta}_j(w)\right] \right). \tag{9}$$

The key that makes the AMIP method fast for solving Equation (9) is that it approximates the impact of dropping a data point[4] by using a first-order approximation (e.g., influence functions and variants). In our case, this approximation amounts to replacing Equation (9) with

$$\max_{w \in W_\alpha} \sum_{n=1}^{N} (1 - w_n) \left(\frac{\partial \hat{\theta}_i(w)}{\partial w_n}\Big|_{w=1_N} - \frac{\partial \hat{\theta}_j(w)}{\partial w_n}\Big|_{w=1_N} \right). \tag{10}$$

Let e_j denote the jth standard basis vector and $\mathbf{X} \in \mathbb{R}^{N \times P}$ denote the design matrix. Let $\widehat{p}_n = \sigma(\hat{\theta}^\top x_n)$ and $\boldsymbol{V} = \mathrm{diag}(\{\widehat{p}_n(1 - \widehat{p}_n)\}_n)$. For logistic regression with an effect-size quantity of interest, θ_j, the formula for the influence score for the nth data point (Pregibon, 1981) is given by

$$\frac{\partial \hat{\theta}_j(w)}{\partial w_n}\Big|_{w=1_N} = e_j^\top (\mathbf{X}^\top \boldsymbol{V} \mathbf{X})^{-1} x_n \widehat{p}_n (1 - \widehat{p}_n) \left(\frac{y_n + 1}{2} - \widehat{p}_n \right). \tag{11}$$

Using these influence score approximations for dropping out single evaluations, we then find the $\lfloor \alpha N \rfloor$-sized subset of evaluations that correspond to the largest influence scores.[5] We denote the approximate solution to Equation (9) that is returned by AMIP as $\tilde{w} \in W_\alpha$ (i.e., the set of data weights that are 0 at indices of data points that AMIP chooses to drop and 1 elsewhere).

For a candidate pair of players, (i, j), we check whether $[\widehat{\theta}_i(\tilde{w}) - \widehat{\theta}_j(\tilde{w})] < 0$. In other words, we refit the BT-model upon leaving out the subset of impactful evaluations identified by AMIP and check whether leaving out this subset induces a sign change in the difference of BT scores for the pair, (i, j). We say that the BT scores for a pair of players, (i, j), are non-robust if the *sign* of the difference in scores *becomes negative* upon refitting under $\tilde{w}$, (i.e., if $[\widehat{\theta}_i(\tilde{w}) - \widehat{\theta}_j(\tilde{w})] < 0$).

[3] The code implementation of our method can be found at https://github.com/JennyHuang19/IsRankingRobust.

[4] By "impact" here, we mean the impact of data-dropping on some statistical quantity-of-interest, such as a regression coefficient, or a parameter, or a test prediction.

[5] In addition to the influence score approximation, our method allows for the use of another type of data-dropping approximation known as the Additive one-step Newton approximation (Park et al., 2023; Huang et al., 2024). For more details on data-dropping approximations, see Appendix A.

Method for Checking Top-k Robustness. We now describe how we can fold our check for pairwise robustness into an overall routine for checking for top-k robustness.

Recall from Section 3 that we can check Top-k robustness by checking pairwise robustness for every comparison (i, j) where $i \in \mathcal{K}_{\mathcal{T}(w)}$ and $j \notin \mathcal{K}_{\mathcal{T}(w)}$, the pair $(\widehat{\theta}_i(w), \widehat{\theta}_j(w))$ is robust. This amounts to checking the pairwise robustness for $k(M-k)$ pairs.

We do this by iterating over pairs of players. Not all pairs need to be checked since we only need to find one non-robust pair to render the set non-robust. To save on compute, we take a greedy approach and start with comparing the most closely-ranked pairs between the top-k ranked players and the remaining $M-k$ players, where "closeness" is quantified using the absolute difference in BT scores fit on the full data.[6] This is because the closest pairs are the most likely to exhibit data-dropping non-robustness. Upon finding any single pair that is pairwise non-robust at an α-level, the procedure terminates early and returns the corresponding players and the indices of the dropped evaluations. We say that an arena is α-level top-k robust if there does not exist a pair of players (i, j), where $i \in \mathcal{K}_{\mathcal{T}(w)}$ and $j \notin \mathcal{K}_{\mathcal{T}(w)}$, that are α-level pairwise non-robust.

Runtime. We provide a fast procedure for assessing the robustness of preference-based ranking systems. For example, we tested our method on historical preference datasets released by the Chatbot Arena project and hosted on Hugging Face (Chiang et al., 2024a). Specifically, we ran top-1 and top-k robustness on a dataset of size around 50,000 evaluations in under 3 minutes on a personal computer equipped with an Apple M1 Pro CPU at 3200 MHz and 16 GB of RAM.

3.1. Deviations from LLM-Arena Computations in Practice

While our methods are based largely on the ranking system used in Chatbot Arena and MT-Bench, there are a few differences between the raw BT scores we analyze and the scores implemented on the platform. The first difference, which concerns the filtering of ties, may affect the resulting rankings, whereas the second difference, an affine transformation, does not. We provide more information on these differences next.

The primary difference is that our analysis is conducted on tie-free data (we discard any matchups that were annotated as ties). In contrast, the Chatbot Arena leaderboard calculation (Bradley–Terry model) (Chiang et al., 2024b) incorporates ties using a weighted BT model. Specifically, in this weighted version, each pairwise comparison between models i and j contributes a row to the estimation procedure, with the weight on each row corresponding to the number of times that matchup occurred. Ties are treated by splitting their weight evenly across both directions. While our tie-free rankings differ from those used in practice by Chatbot Arena, omitting ties is fairly common in practice for analyses done on the platform; for example, many of the plots on the Chatbot Arena Overview Page (Chiang et al., 2024c) are based on tie-free data, and the primary BT model analyzed in the original Chatbot Arena paper also omits ties (Chiang et al., 2024a).[7]

A second difference lies in postprocessing. Chatbot Arena applies a linear transformation to the learned BT scores. They use $\texttt{SCALE} = 400$, $\texttt{INIT\_RATING} = 1{,}000$, and a further shift $\texttt{ANCHOR\_SHIFT}$ to produce the displayed scores:

$$\texttt{ELO}_n = \texttt{SCALE} \cdot \widehat{\theta}_n + \texttt{INIT\_RATING} + \texttt{ANCHOR\_SHIFT}.$$

The final constant ($\texttt{ANCHOR\_SHIFT}$) shifts all the $\texttt{ELO}_n$ scores relative to a specific reference model. Chatbot Arena uses `mixtral-8x7b-instruct-v0.1` as the anchor, assigning it a fixed score of 1,114 and adjusting all other scores accordingly. We note that the affine transformation does not affect model rankings because it is a strictly monotonic transformation.

4. Experiments

Our analysis reveals that dropping as little as 0.02% of the evaluation data can flip the top-ranked model in popular LLM evaluation platforms, that crowdsourced human-evaluated systems are about as non-robust as AI-evaluated systems, and that the responses of dropped evaluations appear similar in content. Henceforth, for convenience, we use "robustness" as shorthand for robustness of a system's top-k ranking to dropping a small fraction, α, of the data.

4.1. Data and Setup.

Chatbot Arena. Chatbot Arena is a crowdsourced platform where users engage in conversations with two chatbots at the same time and rate their responses based on personal preferences (Zheng et al., 2023). This benchmark contains a total number of 57,477 evaluations.[8] Figure 1 presents the Bradley–Terry scores of the top-20 models on Chatbot Arena.

MT-Bench. MT-Bench is a benchmark composed of open-ended questions designed to assess a chatbot's ability to

[6]The robustness of the relative ranking of two players is correlated with the proximity of their BT scores as seen in Figure 6.

[7]Chiang et al. (2024a) introduce variations of the original model that does handle ties.

[8]We conduct our analysis on historical datasets released by the Chatbot Arena project (Chiang et al., 2024a) and hosted on Hugging Face; specifically, we use the arena-human-preference-55k dataset, the chatbot-arena-llm-judges dataset, and the mt-bench-human-judgments dataset.

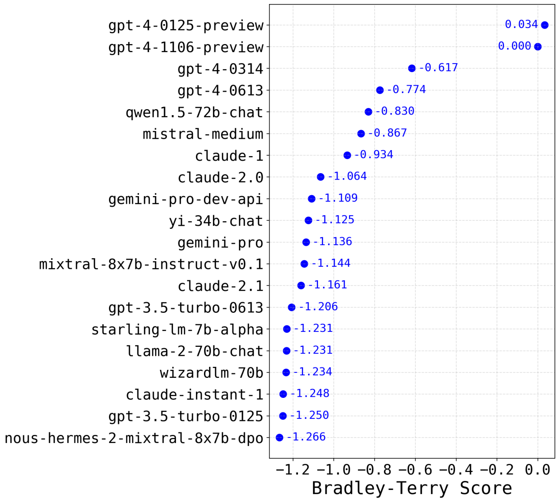


Figure 1. The top-20 model rankings in Chatbot Arena.

engage in multi-turn conversation and follow instructions (Zheng et al., 2023). It is also constructed to distinguish between models based on reasoning and mathematics (Zheng et al., 2023). This benchmark is much smaller because it is handcrafted using 58 expert-level human labelers; it contains 3,355 total evaluations. In contrast to Chatbot Arena, these labelers are mostly graduate students, so they are considered more skilled than average crowd workers. Figure 5 presents the BT scores of the models in the arena.

Experimental Setup. Using historical data from each platform, publicly available on Hugging Face, we construct a design matrix based on model-vs-model comparisons, treating each evaluation as a binary outcome.

Using this design (for more details on this setup, see Section 2), we compute the model rankings based on full-data BT scores (see Figure 3a for the top-10 ranked models). We then run our method to check for top-k robustness, for $k \in \{1, 3, 5, 10, 20\}$ (see Section 3 for more details on this procedure).

4.2. Sensitivity of Chatbot Arena and MT-Bench

Dropping less than 0.05% of the evaluation data is sufficient to change the top-1 and top-5 model rankings in Chatbot Arena. MT-Bench, however, requires over 3% of the data to be removed in order to change the top ranked model and over 4% to change one of the models in the top-5.

We find Chatbot Arena to be incredibly sensitive to data-dropping. In particular, we find that dropping only around 0.02% of the evaluations changes the top-ranked model (GPT-4-0125) to GPT-4-1106. We then find that dropping 20 (0.035% of) evaluations can change one of the models in the top-5 rankings (the 5th and 6th-ranked models changed). Surprisingly, GPT-4-0125 and GPT-4-1106 participated in the most matchups across the entire arena, as shown in Figure 4, suggesting that data-dropping sensitivity cannot be attributed to a small sample size alone.

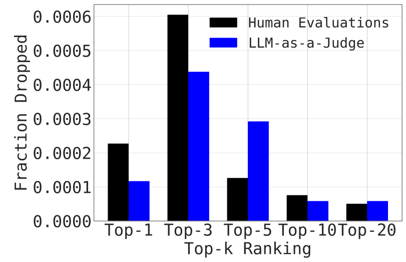


Figure 2. Each bar shows the percentage of data points in Chatbot Arena that are sufficient to be dropped in order to demote the BT score of a model inside the top-k to outside of the top-k. Results are shown for $k \in \{1, 3, 5, 10, 20\}$. The black bars denote results for human evaluators while the blue bar indicates results for LLM-as-a-judge evaluators.

We find MT-Bench Arena to be more robust than Chatbot Arena (see Table 1). Here, dropping 83 out of 2,575 (3.22% of) evaluations changes the top model from GPT-4 to Claude-v1. We then find that dropping 110 (4.27% of) matchups can change one of the models in the top-5 rankings (again, the 5th and 6th ranked models changed).

There are several reasons that may lead MT-Bench to be much more robust than Chatbot Arena, such as the use of expert annotators and the higher quality of prompts (Zheng et al., 2023).

4.3. Humans vs. LLM-as-a-Judge

Within both arenas (Chatbot Arena and MT-Bench), we find the sensitivity of the BT-based top-k rankings to worst-case data-dropping to be similar between human- and LLM-judged evaluations (see Table 1 and Figure 2).

When asked what percentage of evaluations were required to change the top-k model rankings, we find that the human-evaluated data were more sensitive (required dropping fewer evaluations) for $k \in \{5, 20\}$ while the LLM-evaluated platform was more sensitive for $k \in \{1, 3, 10, 20\}$ (see Figure 2). This finding agrees with previous work showing that strong LLMs, such as GPT-4, can closely approximate human preferences in model evaluation tasks. In particular, Zheng et al. (2023) demonstrate that GPT-4 achieves over 80% agreement with expert human annotators on MT-Bench.

4.4. Player Involvement in Dropped Matches

For every instance where the top-k leaderboard changes due to dropped evaluations, we find that the affected matches always involve at least one of the models whose rank is altered (see Figure 3). This holds true for both human-judged and LLM-judged Chatbot Arenas. While Min et al. (2025) find that adding in a small fraction of rigged votes can influence a target model's ranking even when the target model is not directly involved in the matchup, we are unable to find

Arena	Evaluator (Judge)	Number of Points Dropped	Fraction of Points Dropped	Win Percent
Chatbot Arena	Human	9 out of 39716	0.000226	55.17%
Chatbot Arena	LLM	4 out of 34297	0.000116	53.73%
MT-Bench	Human	83 out of 2575	0.0322	62.75%
MT-Bench	LLM	27 out of 2180	0.0124	51.72%

Table 1. Results of checking top-1 robustness of BT-scores on each of the two arenas (Chatbot Arena and MT-Bench). The “Number of Points Dropped” column reports on the number of matches that are sufficient to flip the first and second-place models. The “Fraction of Points Dropped” column shows this number as a fraction of the number of total matches in the full arena. The “Win Percent” column shows the proportion of head-to-head matchups won by the first-place model in matchups against the second-place model. Small BT score differences and win margins indicate that the two models have similar performance, so even minor perturbations in data may easily flip their ranking.

instances where removing a small fraction of evaluations where neither of the affected models were involved flips the rankings.

Also, notice in Figure 3 that there are no partial bars or mixed compositions. We investigate why this homogeneous pattern appears consistently across bars. Inspecting dropped matchups manually (see Appendix B), we find that the reason the involvement is always entirely either one or both flipped players is because the dropped matchups always consists of games between a central model and a specific competitor (or group of competitors) whose outcomes all favor or disfavor that specific model, a structure that leads to homogeneous bars. This finding reveals something about how non-robustness appears in our analyses: small, consistent sets of matchups are sufficient to push a model just above or below another on the leaderboard.

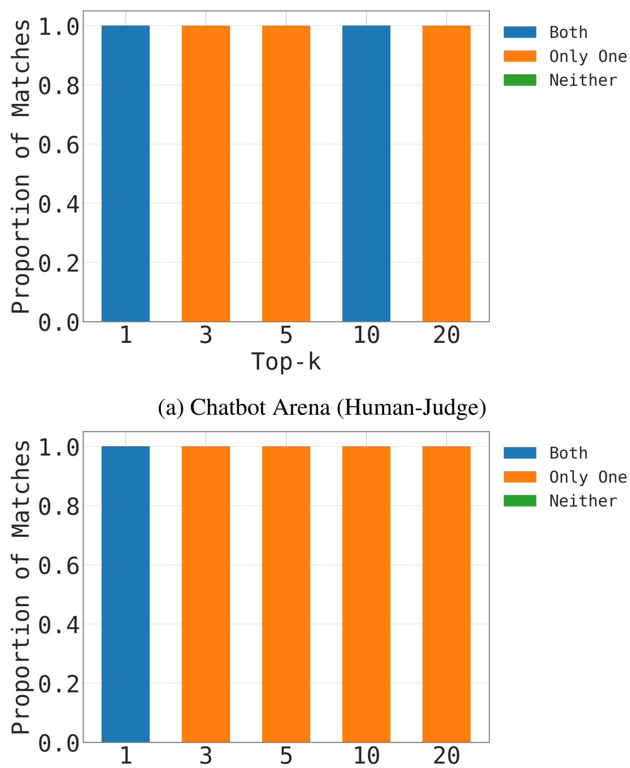
4.5. Inspecting Dropped Matches

We demonstrate that our method can be used to investigate the dropped prompts and responses, and we find the dropped responses to be similar in content.

Setup. To investigate the dropped prompt-response pairs, we compare the response similarity of just *five* human evaluations that were enough to flip the rankings of mistral-medium (the 5th ranked model) with qwen1.5-72b-chat (the 6th ranked model) on Chatbot Arena.

These five influential evaluations are those where qwen1.5-72b-chat performed exceedingly well, winning against gpt-4-1106-preview (the second-place model). Dropping these five matchups was enough to demote qwen1.5-72b-chat’s ranking from out of the top 5, to below that of the 6th place mistral-medium. These five prompts spanned a diverse range of tasks (e.g., technical instruction, legal reasoning, opinion clarification) and lengths.

Given the influence of this small set of responses in determining the top-5 rankings on the arena, we display the prompts and responses involved in Table 2. Upon visual inspection of these and other dropped matchups, we find that the paired responses are similar in both content and style. This similarity—despite leading to ranking reversals—



(a) Chatbot Arena (Human-Judge)

(b) Chatbot Arena (LLM-Judge)

Figure 3. Player involvement in the most influential matches whose removal caused a model to drop out of the top-k positions. Each bar represents the composition of matches dropped in order to demote a model from the top-k set. The proportions indicate whether none (green), one (orange), or both (blue) of the affected models were involved in each dropped matchup. For Chatbot Arena (Human-Judge) (top), the number of matches dropped for each of the top-k is $(9, 24, 5, 3, 2)$, respectively. For Chatbot Arena (LLM-Judge) (bottom), the number of matches dropped for each of the top-k is $(4, 15, 10, 2, 2)$. Across all k, we find that 100% of flipped rankings involved matches containing at least one of the players whose ranking changed.

Table 2. Side-by-side comparison of LLM responses to five influential prompts that were dropped. Lengthy prompts and answers are truncated here (indicated by "...") to ensure readability.

Prompt	**Response A**	**Response B**
"Read this content. Do not change the content written. Add 5 more tips in the same tone of voice and style. One be use AI. Hey freelance writers: Let's talk about writer's block..."	Use AI Writing Tools. Incorporating AI writing tools can occasionally help break the creative gridlock...	Use AI to Your Advantage. In this age of technology, leverage the power of AI to overcome writer's block....
"Which is the newest model that also contain data of 2023?"	I'm sorry, but your question isn't clear. When you ask about the newest model that also contains data of 2023, I need more context to provide you with an accurate answer....	As an AI, I don't have real-time access to updates or the ability to browse the internet for the most recent information....
"Please answer this questions in 8 sentences and do not use wording indicative of AI. You are provided with a legal description that seems to make sense when you initially read but once you arrive at the site it no longer does..."	When a legal description appears clear initially but becomes confusing at the site, it indicates an issue of latent ambiguity...	The type of ambiguity encountered is likely a latent ambiguity, where the description seems clear on paper but is problematic when applied to the physical location....
"Is it possible to give a transformer custom embeddings?"	Yes, it is possible to give a transformer model custom embeddings...	Yes, it is possible to give a transformer model custom embeddings...
"How to use Poetry to install packages in python?"	Poetry is a tool for dependency management and packaging in Python...	Poetry is a dependency manager for Python that makes it easy to manage project dependencies and create self-contained virtual environments...

underscores the fragility of the ranking system and raises concerns about its sensitivity to subtle variations in language model responses.

5. Discussion

Crowdsourced LLM evaluation platforms like Chatbot Arena offer a way to rank LLMs by aggregating preferences over responses to open-ended prompts. There is good reason that this setup has been widely-adopted: it is easy to scale, doesn't require expert annotators, and enables the aggregation of many prompts and judgments across a wide range of users (Zheng et al., 2023; Don-Yehiya et al., 2024).

In theory, this aggregation helps average out individual annotator variability and yields a signal that is generalizable. However, in practice, we find that model rankings can depend on just a small handful of human (or LLM) evaluations. Thus, we encourage users of leaderboards and benchmark contests to run our method to investigate the fragility of crowdsourced LLM evaluation platforms before publishing results.

We find that rankings based on MT-Bench matchups are more robust than those from Chatbot Arena to the removal of a small number of evaluations, likely due to higher-quality data; unlike Chatbot Arena, which relies on noisy, user-generated prompts and crowdsourced votes from anonymous users, MT-Bench uses carefully constructed, multi-turn questions evaluated by expert annotators, such as graduate students (Zheng et al., 2023). Additionally, MT-Bench prompts are specifically designed for reasoning, math, and instruction-following, which may lead to more decisive win/loss outcomes (we observed many fewer ties in MT-Bench). Together, these factors result in MT-Bench rankings being less sensitive to worst-case small-fraction data-dropping, indicating that using carefully-constructed queries and expert evaluators may result in more robust benchmarks for evaluating chat model performance.

For researchers in the field of human-AI alignment, more rigorous and nuanced evaluation strategies are needed. To this end, we highlight several promising directions for the future of open human feedback. These include eliciting not only binary preference but also evaluators' confidence levels

(Méndez et al., 2022),[9] creating tools to identify prompts requiring specialized knowledge in order to route them to appropriate evaluators (Don-Yehiya et al., 2024), using mediators to perform fine-grained assessments of crowdsourced responses (Don-Yehiya et al., 2024), and categorizing prompts by instruction type (e.g., factual recall, creative generation) to promote more fine-grained model comparisons within categories (Chia et al., 2023).

Impact Statement

We introduce a fast method to audit the robustness of LLM ranking systems—such as Chatbot Arena and MT-Bench—to the removal of a very small fraction of evaluation data. Our findings show that leaderboard rankings can be overturned by dropping as little as 0.02% of evaluations, raising concerns about the generalizability of human preference-based evaluations. By identifying specific evaluations that flip rankings, our method provides a diagnostic tool for investigating potentially outlying prompts and responses in LLM evaluation platforms. This work inspires actionable insights and provides a concrete tool for improving the trusworthiness and credibility of leaderboard-based evaluation of generative AI systems.

References

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

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

Broderick, T., Giordano, R., and Meager, R. An automatic finite-sample robustness metric: When can dropping a little data make a big difference? *arXiv preprint arXiv:2011.14999v1*, 2020.

Chia, Y. K., Hong, P., Bing, L., and Poria, S. Instructeval: Towards holistic evaluation of instruction-tuned large language models. *arXiv preprint arXiv:2306.04757*, 2023.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhu, B., Zhang, H., Jordan, M., Gonzalez, J. E., et al. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*, 2024a.

[9]The weighted logistic regression model used by chatbot arena can easily be extended to take in confidence ratings on top of binary preferences. One could imagine implementing this through encoding the confidence rating as a weight in the Win-Counts matrix described in the "Chatbot Arena Leaderboard Calculation (Bradley–Terry model)" colab notebook referenced in Section 4.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhu, B., Zhang, H., Jordan, M., Gonzalez, J. E., et al. Chatbot Arena Leaderboard Calculation (Bradley–Terry model). https://colab.research.google.com/drive/1KdwokPjirkTmpO_P1WByFNFiqxWQquwH, 2024b. Google Colab notebook.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhu, B., Zhang, H., Jordan, M., Gonzalez, J. E., et al. Chatbot Arena Leaderboard. https://lmarena.ai/leaderboard/text, 2024c. Accessed: 2025-06-23.

Damani, M., Shenfeld, I., Peng, A., Bobu, A., and Andreas, J. Learning how hard to think: Input-adaptive allocation of lm computation. *arXiv preprint arXiv:2410.04707*, 2024.

Don-Yehiya, S., Burtenshaw, B., Astudillo, R. F., Osborne, C., Jaiswal, M., Kuo, T.-S., Zhao, W., Shenfeld, I., Peng, A., Yurochkin, M., et al. The future of open human feedback. *arXiv preprint arXiv:2408.16961*, 2024.

Freund, D. and Hopkins, S. B. Towards practical robustness auditing for linear regression. *arXiv preprint arXiv:2307.16315*, 2023.

Ghosh, S., Stephenson, W., Nguyen, T. D., Deshpande, S., and Broderick, T. Approximate cross-validation for structured models. *Advances in neural information processing systems*, 33:8741–8752, 2020.

Giordano, R., Jordan, M. I., and Broderick, T. A higher-order Swiss Army infinitesimal jackknife. *arXiv preprint arXiv:1907.12116v1*, 2019.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011.

Huang, J. Y., Burt, D. R., Nguyen, T. D., Shen, Y., and Broderick, T. Approximations to worst-case data dropping: unmasking failure modes. *arXiv preprint arXiv:2408.09008*, 2024.

Hui, B., Yang, J., Cui, Z., Yang, J., Liu, D., Zhang, L., Liu, T., Zhang, J., Yu, B., Lu, K., et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Kuschnig, N., Zens, G., and Crespo Cuaresma, J. Hidden in plain sight: Influential sets in linear models. Technical report, CESifo Working Paper, 2021.

Lee, H., Phatale, S., Mansoor, H., Lu, K. R., Mesnard, T., Ferret, J., Bishop, C., Hall, E., Carbune, V., and Rastogi, A. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. In *Proceedings of Machine Learning Research*, volume 235, 2023.

Méndez, A. E., Cartwright, M., Bello, J. P., and Nov, O. Eliciting confidence for improving crowdsourced audio annotations. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW1):1–25, 2022.

Min, R., Pang, T., Du, C., Liu, Q., Cheng, M., and Lin, M. Improving your model ranking on chatbot arena by vote rigging. *arXiv preprint arXiv:2501.17858*, 2025.

Moitra, A. and Rohatgi, D. Provably auditing ordinary least squares in low dimensions. *arXiv preprint arXiv:2205.14284*, 2022.

Nguyen, T. D., Giordano, R., Meager, R., and Broderick, T. Using gradients to check sensitivity of MCMC-based analyses to removing data. In *ICML 2024 Workshop on Differentiable Almost Everything: Differentiable Relaxations, Algorithms, Operators, and Simulators*, 2024.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Park, S. M., Georgiev, K., Ilyas, A., Leclerc, G., and Madry, A. Trak: Attributing model behavior at scale. In *International Conference on Machine Learning*, pp. 27074–27113. PMLR, 2023.

Pregibon, D. Logistic Regression Diagnostics. *The Annals of Statistics*, 9(4), 1981.

Rubinstein, I. and Hopkins, S. Robustness auditing for linear regression: To singularity and beyond. *The 13th International Conference on Learning Representations (ICLR 2025)*, 2025.

Sekhari, A., Acharya, J., Kamath, G., and Suresh, A. T. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

Shiffman, M., Giordano, R., and Broderick, T. Could dropping a few cells change the takeaways from differential expression? *arXiv preprint arXiv:2312.06159*, 2023.

Singh, S., Nan, Y., Wang, A., D'Souza, D., Kapoor, S., Üstün, A., Koyejo, S., Deng, Y., Longpre, S., Smith, N., et al. The leaderboard illusion. *arXiv preprint arXiv:2504.20879*, 2025.

Sun, H., Shen, Y., and Ton, J.-F. Rethinking reward modeling in preference-based large language model alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.

Suriyakumar, V. and Wilson, A. C. Algorithms that approximate data removal: New results and limitations. *Advances in Neural Information Processing Systems*, 35: 18892–18903, 2022.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

White, C., Dooley, S., Roberts, M., Pal, A., Feuer, B., Jain, S., Shwartz-Ziv, R., Jain, N., Saifullah, K., Naidu, S., et al. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*, 4, 2024.

Wilson, A., Kasy, M., and Mackey, L. Approximate cross-validation: Guarantees for model assessment and selection. In *International conference on artificial intelligence and statistics*, pp. 4530–4540. PMLR, 2020.

Wu, W., Junker, B. W., and Niezink, N. Asymptotic comparison of identifying constraints for bradley-terry models. *arXiv preprint arXiv:2205.04341*, 2022.

Xu, S., Yue, B., Zha, H., and Liu, G. Uncertainty-aware preference alignment in reinforcement learning from human feedback. In *ICML 2024 Workshop on Models of Human Feedback for AI Alignment*, 2024.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623, 2023.

A. The AMIP Method

In principle, one might solve

$$\max_{w \in W_\alpha} \left([\widehat{\theta}_i(1_N) - \widehat{\theta}_j(1_N)] - [\widehat{\theta}_i(w) - \widehat{\theta}_j(w)] \right) \tag{12}$$

by computing $[\hat{\theta}_i(w) - \hat{\theta}_j(w)]$ for each of the $\binom{N}{\lfloor \alpha N \rfloor}$ values of $w \in W_\alpha$. However, this brute force approach can be computationally prohibitive even for moderately-sized N.

Approximate Maximum Influence Perturbation (AMIP). Broderick et al. (2020) propose relaxing w to allow continuous values and replacing the w-specific quantity of interest with a first-order Taylor series expansion with respect to w around 1_N. This first-order Taylor series expansion is known as the *influence function (IF)* approximation (Hampel et al., 2011), a classic technique from robust statistics that approximates the affect of upweighting (or dropping) a data point on model parameters using a first-order Taylor series approximation in data-weight space. Influence functions have become popular tools for approximating resampling methods (Giordano et al., 2019) and assigning value to data that a model was trained on (Koh & Liang, 2017; Park et al., 2023). This approximation applies to more general data analyses and quantities of interest. In our case, this approximation amounts to replacing Equation (12) with

$$\max_{w \in W_\alpha} \sum_{n=1}^{N} (1 - w_n) \left(\frac{\partial \hat{\theta}_i(w)}{\partial w_n}\Big|_{w=1_N} - \frac{\partial \hat{\theta}_j(w)}{\partial w_n}\Big|_{w=1_N} \right). \tag{13}$$

Let e_j denote the jth standard basis vector and $\mathbf{X} \in \mathbb{R}^{N \times P}$ denote the design matrix. Let $\widehat{p}_n = \sigma(\hat{\theta}^\top x_n)$ and $\boldsymbol{V} = \operatorname{diag}(\{\widehat{p}_n(1 - \widehat{p}_n)\}_n)$. For logistic regression with an effect-size quantity of interest, θ_j, the formula for the influence score for the nth data point (Pregibon, 1981) is given by

$$\frac{\partial \hat{\theta}_j(w)}{\partial w_n}\Big|_{w=1_N} = e_j^\top (\mathbf{X}^\top \boldsymbol{V} \mathbf{X})^{-1} x_n \widehat{p}_n (1 - \widehat{p}_n) \left(\frac{y_n + 1}{2} - \widehat{p}_n \right), \tag{14}$$

In addition to influence functions, our framework enables a second data-dropping approximation known as the *One-step Newton (1sN)* approximation, which approximates the effect of dropping a data point on model parameters using a second-order Taylor expansion in parameter space. This Newton-style update has become popular for approximating the deletion of data in recent works on approximate cross validation (Ghosh et al., 2020; Wilson et al., 2020) and machine unlearning (Sekhari et al., 2021; Suriyakumar & Wilson, 2022). The 1sN is slightly more expensive to compute than the IF approximation (as it corrects the IF with a multiplicative correction term) but is more accurate when the to-be-dropped data point has high a leverage score (because the correction term involves the leverage score of a data point). Previous works have proposed approximating the removal of a group of data points by the sum of leave-one-out 1sN scores, in an algorithm known as the **Additive one-step Newton approximation** (Huang et al., 2024; Park et al., 2023).

To run the AMIP and Additive one-step Newton algorithm, we:

1. Fit a BT model on the entire arena.
2. Compute the *influence scores* (Equation (11)) (one-step Newton scores for the Additive one-step Newton algorithm) for all matches in the arena.
3. Identify the $\lfloor \alpha N \rfloor$ matches with the largest influence scores.
4. Approximate impact of dropping the top-scoring $\lfloor \alpha N \rfloor$ matches by the sum of the scores of the top $\lfloor \alpha N \rfloor$ matches.
5. If the approximation predicts a change in ranking, then refit the model leaving out the identified subgroup.[10]

These data-dropping algorithms replace a computationally intractable combinatorial search with an algorithm that costs only

$$O(Analysis + N \log(\alpha N) + NP^2 + P^3),$$

where $Analysis$ represents the cost of fitting the initial Bradley–Terry model on the original arena to compute scores. Data-dropping approximations make identifying candidate subsets of the arena that may induce top-k non-robustness very fast because they eliminate the need to retrain the BT model repeatedly on every candidate subset. Once a candidate subset is identified, however, our method always performs a *refitting* of the BT model with the identified subset removed to verify whether the non-robustness is true. This final verification step ensures that our method does not return false positives.

[10] Our algorithm gives users the option to refit the BT model for all matchups, regardless of whether a predicted ranking change occurs.

B. Player Involvement, Homogeneous Bars

Across all top-k robustness experiments, 100% of dropped matches involved either one or both of the models whose rankings were flipped, with 100% belonging to one of these two cases within a given k (see Figure 3). There are no partial bars or mixed compositions. Readers may ask: Why does this homogeneous pattern consistently appear? Could this be a property of the arena data?

We investigate this by manually inspecting the dropped matchups returned by our robustness assessing algorithm for each value of k. Specifically, in each case, we identified the dropped matchups and inspected which players appeared in these matchups. We summarize the findings here:

- $k = 1$: 9 games were dropped to flip GPT-4-0125-preview (originally 1st) and GPT-4-1106-preview (2nd). All were matches where GPT-4-0125-preview beat GPT-4-1106-preview.
- $k = 3$: 24 games were dropped to flip GPT-4-0314 (3rd) and Qwen1.5-72B-Chat (5th). All dropped matches featured Qwen1.5-72B-Chat losing out to various other models.
- $k = 5$: 5 games were dropped to flip Qwen1.5-72B-Chat (5th) and Mistral-Medium (6th). All matches were between Qwen1.5-72B-Chat and GPT-4-1106-preview (the 2nd place model), with Qwen1.5-72B-Chat winning.
- $k = 10$: 3 games were dropped to flip Yi-34B-Chat (10th) and Gemini-Pro (11th). All were matches where Yi-34B-Chat beat Gemini-Pro.
- $k = 20$: 2 games were dropped to flip Nous-Hermes-2-Mixtral-8x7B-DPO (20th) and Vicuna-33B (21st). Both were matches where Nous-Hermes-2 beat GPT-4-1106-preview (the 2nd place model).

The reason the involvement is always entirely either one or both affected players is because all of the dropped matchups consist of games played between a central model and a specific competitor (or group of competitors) whose outcomes all favor or disfavor the specific model. This structure then leads the dropped matchups to consist entirely of evaluations that involved one or both ranking-flipped models. This finding reveals something interesting about the nature of the non-robustness in our analysis: small, consistent sets of matchups are sufficient to push a model just above or below another on the leaderboard.

C. Additional Supporting Figures

The figures in this section provide additional insights related to our analysis. Figure 4 shows the distribution of model appearances in Chatbot Arena, respectively, revealing differences in evaluation density and coverage across platforms. Figure 5 presents the BT scores of top-performing models on MT-Bench, highlighting the competitiveness of high-ranked models. Figure 6 illustrates the relationship between the robustness of model rankings and the BT score gap between adjacent models, confirming that small score differences tend to coincide with greater sensitivity to worst-case data-dropping.

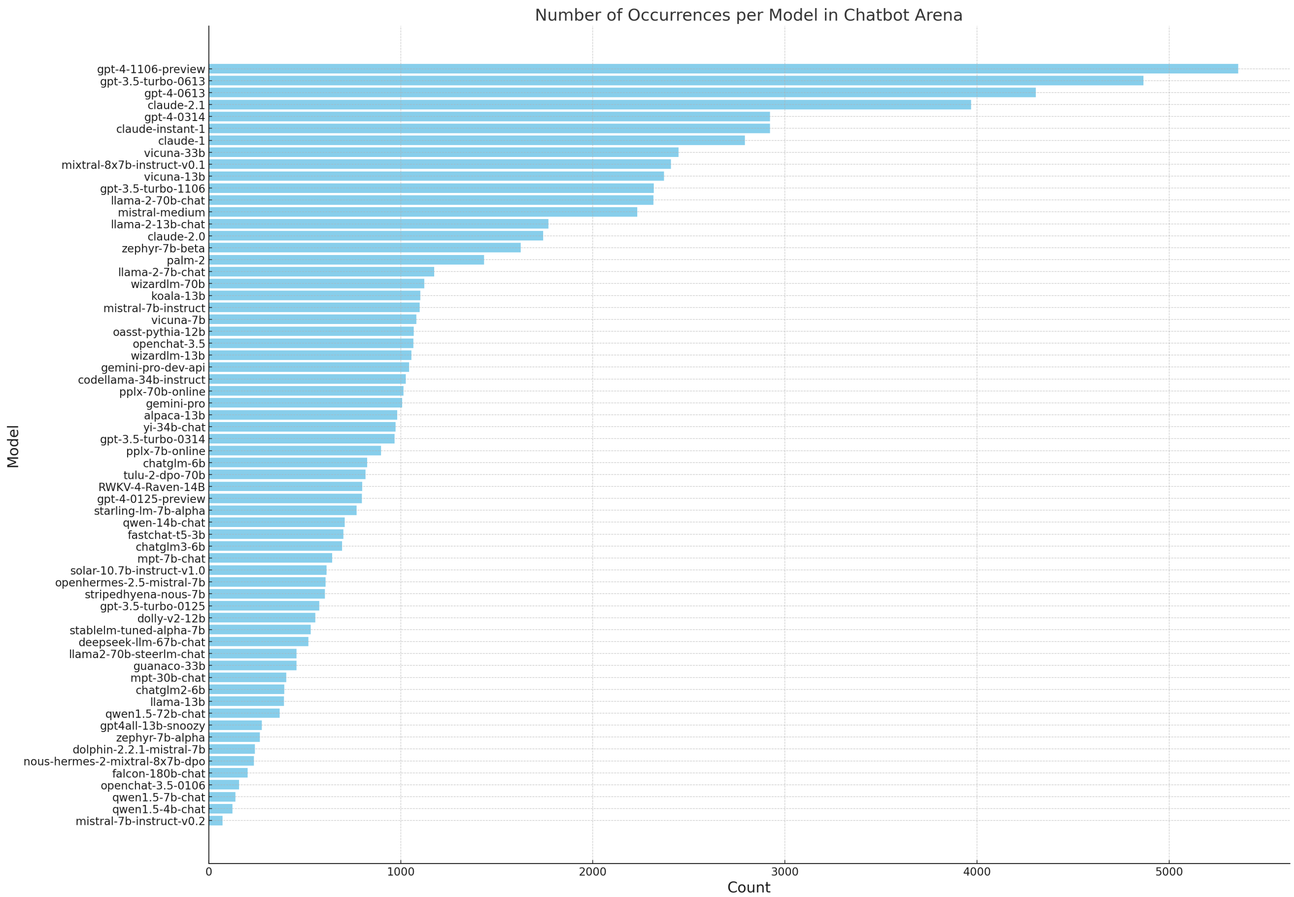


Figure 4. The number of times each model appears in a match in Chatbot Arena. The horizontal bar chart shows how frequently each model appeared in any match, with GPT-4 and GPT-3.5 variants being the most represented.

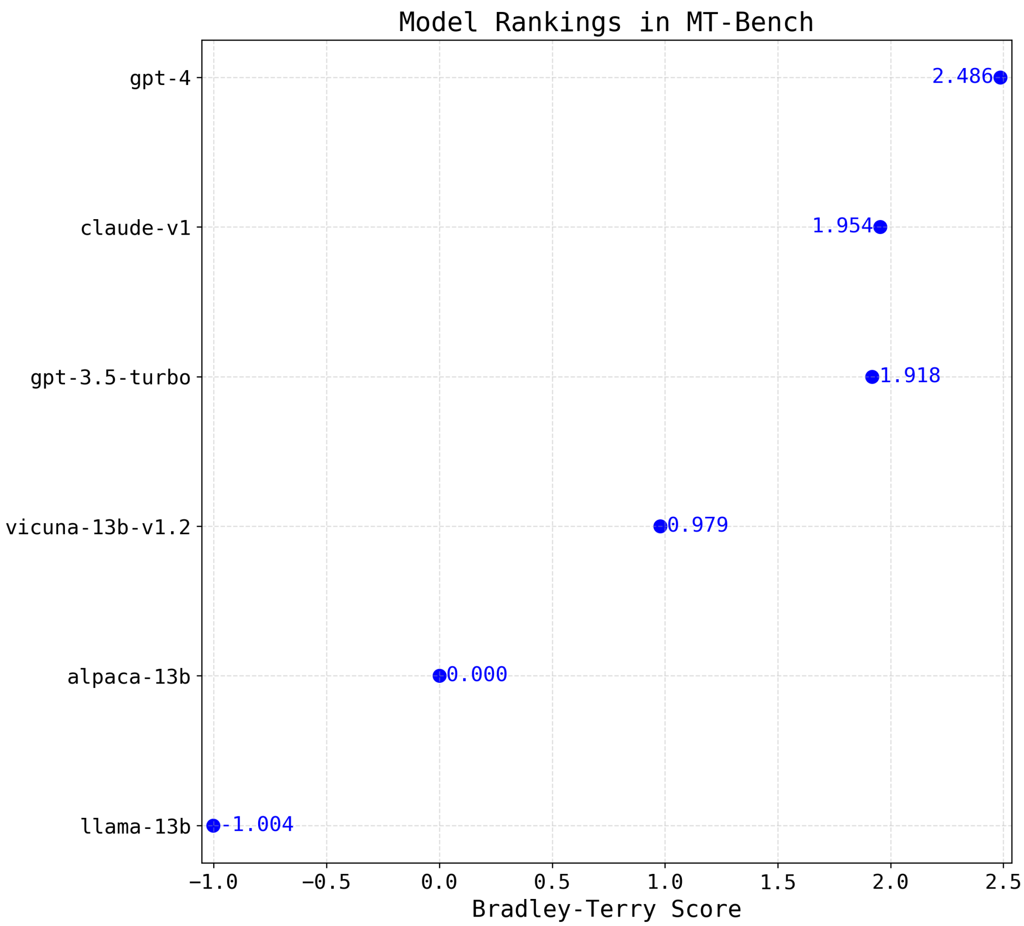


Figure 5. The model rankings in MT-Bench.

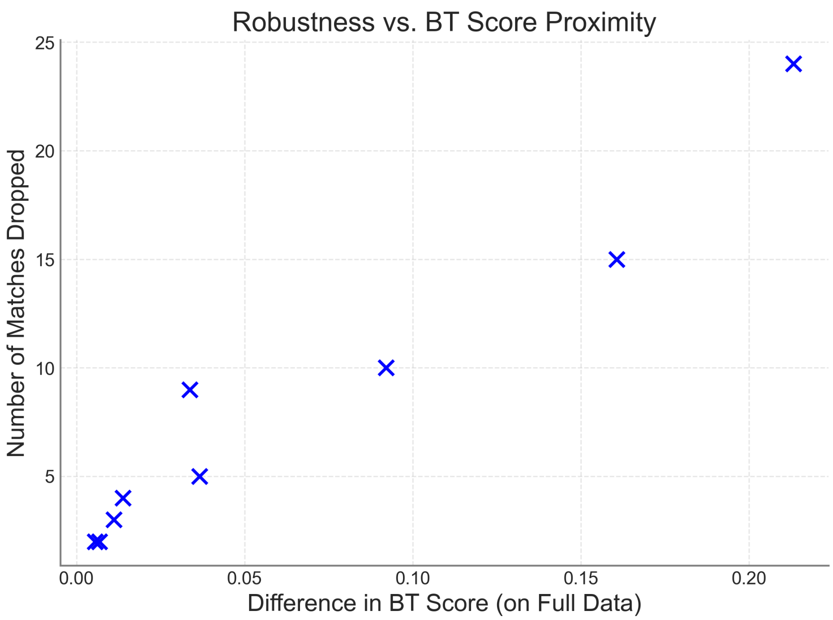


Figure 6. Robustness of results is correlated with the proximity of the BT scores. Each point represents a pair of models whose relative rankings flipped after dropping a small fraction of matchups. In every case, the flip causes one model to enter the top-k rankings (for some $k \in \{1, 3, 5, 10, 20\}$) while the other is demoted. These points are taken from both human and LLM-as-a-judge evaluation platforms.