A Unified Perturbation Framework for Analyzing Leaderboard Stability and Manipulation

Hosna Oyarhoseini[1] **Jimmy Lin**[1,3] **Amir-Hossein Karimi**[1,2,3]
[1]David R. Cheriton School of Computer Science, University of Waterloo
[2]Department of Electrical and Computer Engineering, University of Waterloo
[3]Vector Institute
{hoyarhos, jimmylin, amirh.karimi}@uwaterloo.ca

Abstract

Evaluation leaderboards such as LMArena play a central role in benchmarking large language models by aggregating pairwise human preferences into model rankings, yet the robustness of these rankings remains poorly understood. We present a unified perturbation framework for analyzing Bradley–Terry leaderboards under structured data modifications using influence-based approximations. Our framework studies three match-level perturbations—`Drop`, `Add`, and `Flip`—together with player removal, and evaluates their effects on *top-k membership*, global ranking consistency via *Kendall's τ*, and *confidence-interval-based uncertainty*. Across Chatbot Arena and six additional pairwise-comparison datasets, we show that modern leaderboards are non-robust across all three objectives: sub-1% targeted perturbations can change the top-ranked model, degrade *Kendall's τ*, and alter confidence intervals. We summarize these effects with normalized dataset-level robustness scores that compare fragility across leaderboard designs. We further show that influence scores enable efficient targeted manipulation, promoting or demoting specific models with fewer actions than prior manipulation baselines, while also identifying additional matchups that reduce uncertainty for target models. Finally, player-removal analysis shows that removing influential models can induce broad reordering, highlighting model deprecation as a source of the leaderboard illusion. These findings reveal fundamental limitations of current leaderboard designs and motivate more robust evaluation protocols.

1 Introduction

The rapid proliferation of Large Language Models (LLMs) has necessitated the development of scalable, human-centric evaluation frameworks [Frick et al., 2025, Miroyan et al., 2026, Zheng et al., 2023, Guo et al., 2023]. Because the quality of open-ended generation is subjective and hard to capture with a single absolute metric, modern platforms have increasingly adopted pairwise preference comparisons, where judges choose between two model outputs. In systems such as LMArena/Chatbot Arena, these preferences are primarily collected from human voters [Chiang et al., 2024], though LLM-based judges are also increasingly used in related evaluation settings [Zheng et al., 2023]. This paradigm builds on a classical approach for ranking items from direct comparison outcomes [Bradley and Terry, 1952, Negahban et al., 2017, Wauthier et al., 2013, Shah and Wainwright, 2018], but has recently become a central pillar of LLM benchmarking [Zheng et al., 2023, Chiang et al., 2024]. Most notably, Chatbot Arena [Chiang et al., 2024] uses Bradley–Terry scores derived from crowdsourced votes to produce model rankings. Beyond leaderboard evaluation, Bradley–Terry models also underpin reward-model training for RLHF [Ouyang et al., 2022, Bai et al., 2022, Lee et al., 2024, Touvron et al., 2023, Xu et al., 2024, Sun et al., 2025] and have been used to route queries to suitable LLMs or inference-time scaling strategies [Damani et al., 2025]. As these leaderboards

increasingly shape model adoption and industry recognition [Metz, 2025, Kruppa, 2024, Singh et al., 2025], they are often treated as stable estimates of true model skill.

Recent work challenges this assumption [Huang et al., 2026, Min et al., 2025, Huang et al., 2025, Singh et al., 2025, Wu et al., 2022]. Huang et al. [2026] show that removing only a small number of pairwise comparisons can alter top rankings, while other studies identify vulnerabilities from injected votes [Min et al., 2025], gamed LLM judges [Zheng et al., 2025, Raina et al., 2024], apathetic or arbitrary annotators [Zhao et al., 2025], and data leakage or selective reporting [Singh et al., 2025]. However, existing analyses typically focus on a single *perturbation type* or ranking objective. This leaves open a broader question: *how do different small, structured changes to the comparison dataset propagate through different leaderboard outcomes?* We address this question by building on the idea of Huang et al. [2026] with a unified influence-based perturbation framework for Bradley–Terry leaderboards. Our framework treats dataset modifications as structured interventions and propagates their effects through the ranking estimator to downstream leaderboard conclusions. This allows us to study both robustness failures and targeted interventions within the same formalism, including match-level perturbations and player-level removal motivated by model deprecation or exclusion.

Our main contributions are:

- We introduce a unified influence-based framework for auditing Bradley–Terry leaderboards under three match-level perturbations—`Drop`, `Add`, and `Flip`—instantiated for three ranking objectives: *top-k membership*, *Kendall's τ* for global consistency, and *confidence-interval-based uncertainty* for the reliability of estimated skills.
- Across seven pairwise-comparison datasets, we show that leaderboard non-robustness is systematic across datasets, *perturbation type*, and ranking criteria: fewer than 1% of comparisons substantially affect *top-k membership*, *Kendall's τ*, and CI-based stability. We also aggregate influence-guided failures into normalized dataset-level robustness scores, giving a compact audit profile across Top-k, CI-aware, and global-ranking views.
- The same framework supports targeted interventions, promoting or demoting specific models with fewer actions than prior vote-manipulation [Min et al., 2025] and identifying matchups that reduce uncertainty more effectively than Chatbot Arena-style active sampling [Chiang et al., 2024].
- A player-removal analysis shows that removing influential players can induce broad reordering, highlighting model deprecation as a source of the leaderboard illusion [Singh et al., 2025].

Overall, this work provides a unified perspective on leaderboard non-robustness by bridging robustness analysis and adversarial manipulation, revealing fundamental limitations in current leaderboard designs and highlighting the need for more reliable and trustworthy benchmarking methodologies.

2 Preliminaries

2.1 Pairwise comparison and Bradley–Terry framework

Modern LLM leaderboards rank models from pairwise preference data rather than absolute scores. In this setting, each *player* denotes an entity being ranked; for LLM leaderboards, a player corresponds to an LLM or model. A *match* or *vote* denotes one pairwise comparison between two players, typically obtained when a human or judge compares two model responses to the same prompt and selects the preferred output. These comparisons are commonly modeled using the Bradley–Terry (BT) framework [Bradley and Terry, 1952], which estimates latent skill scores from pairwise outcomes.

We adopt the notation of Chiang et al. [2024], defining a set of M models $\mathcal{M} = \{m_1, \ldots, m_M\}$ and a dataset $D = \{z_n\}_{n=1}^{N}$. Each observation $z_n = (x_n, y_n)$ compares two players (i_n, j_n), where $x_n = e_{i_n} - e_{j_n} \in \mathbb{R}^M$ encodes the matchup and $y_n \in \{0, 1\}$ indicates whether i_n beats j_n. For a comprehensive breakdown of the data format and our specific protocol for handling ties, please refer to Appendix B.1.

The BT model assigns each model m_i a scalar latent skill coefficient $\theta_i \in \mathbb{R}$. For comparison n, let (i, j) denote the ordered pair of models being compared, and let $y_n = 1$ indicate that model i is preferred to model j. The Bradley–Terry probability is

$$p_n := P(y_n = 1 \mid i, j; \theta) = \sigma(x_n^\top \theta) = \sigma(\theta_i - \theta_j) = \frac{1}{1 + e^{-(\theta_i - \theta_j)}}. \tag{1}$$

To ensure identifiability against constant shifts, we fix $\theta_1 = 0$. The parameter vector θ is estimated by minimizing the empirical binary cross-entropy loss:

$$\hat{\theta} = \arg\min_{\theta} \sum_{n=1}^{N} \ell(z_n; \theta) = \arg\min_{\theta} \sum_{n=1}^{N} \big[-y_n \log p_n - (1 - y_n) \log(1 - p_n) \big]. \tag{2}$$

2.2 Influence functions and propagation to objectives

Previous work has used influence scores to identify high-impact deletions under a fixed budget [Broderick et al., 2023]. Building on this idea, we study how perturbing individual BT comparisons affects both the estimated skills and downstream leaderboard conclusions. Let $w \in \mathbb{R}^N$ denote match weights and define the weighted BT estimator

$$\hat{\theta}(w) = \arg\min_{\theta:\theta_1=0} \sum_{n=1}^{N} w_n \ell(z_n; \theta), \tag{3}$$

where $w = \mathbf{1}$ gives the full-data fit and $w_n = 0$ removes match z_n. Let $\hat{\theta} = \hat{\theta}(\mathbf{1})$, $g_n = \nabla_\theta \ell(z_n; \hat{\theta})$, and $H = \nabla^2_\theta \sum_{n=1}^{N} \ell(z_n; \hat{\theta})$. Influence functions approximate the local effect of changing a match weight without refitting [Koh and Liang, 2017]:

$$\frac{\partial \hat{\theta}(w)}{\partial w_n}\Big|_{w=\mathbf{1}} = -H^{-1} g_n. \tag{4}$$

We propagate this parameter sensitivity to a scalar leaderboard objective $f(\hat{\theta}(w), w)$, such as a *top-k membership*, *Kendall's τ* surrogate, or *confidence-interval-based uncertainty* criterion. Allowing explicit dependence on w captures objectives, such as uncertainty, that change directly with the weighted comparison graph. The objective-level influence of match z_n is

$$\begin{aligned} I_n^{(f)} &:= \nabla_\theta f(\hat{\theta}(w), w)^\top \frac{\partial \hat{\theta}(w)}{\partial w_n} + \frac{\partial f(\hat{\theta}(w), w)}{\partial w_n}, \\ f(\hat{\theta}(w + \Delta w), w + \Delta w) &\approx f(\hat{\theta}(w), w) + \sum_{n=1}^{N} \Delta w_n I_n^{(f)}. \end{aligned} \tag{5}$$

In the leaderboard setting, Huang et al. [2026] instantiate this idea with *top-k membership* objectives $f(\theta) = \theta_i - \theta_j$ across top-k boundaries, showing that dropping a small number of preference votes can change top-ranked membership. We generalize this view by decoupling the perturbation mechanism from the audited objective: the same propagation rule applies to multiple leaderboard objectives and, later, to `Drop`, `Add`, and `Flip` actions. Because our actions are finite rather than infinitesimal, following Huang et al. [2026], we use a one-step Newton (1sN) refinement to partially account for curvature changes in the perturbed objective; details are in Appendix B.2.

3 Influence framework

The goal of our framework is to systematically quantify how perturbations to the leaderboard dataset D propagate to downstream ranking objectives, with an overview of the framework shown in Figure 1.

3.1 Action space

We consider three fundamental actions that characterize the ways a leaderboard can be manipulated. Each action corresponds to a specific modification of the weighted M-estimator defined in Eq. 3.

Action 1: Match Dropping (`Drop`). Following prior work that uses influence methods to show that LLM leaderboards are non-robust to small amounts of data removal [Huang et al., 2026], we model dropping a match $z_n \in D$ as setting its weight from $w_n = 1$ to $w_n = 0$. The effect of this single perturbation on a downstream objective is approximated using influence as:

$$\Delta f_{\text{drop},n} \approx -\mathcal{I}_n^{(f)}. \tag{6}$$

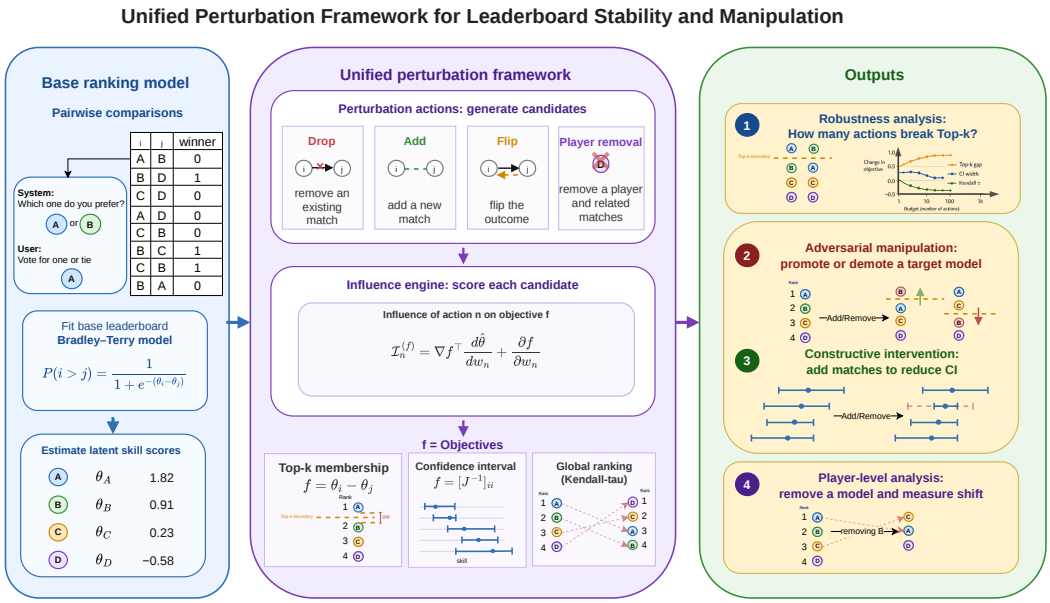


Figure 1: Overview of the framework: influence scores are computed for each action type (`Drop`, `Add`, `Flip`) and propagated through the Bradley–Terry estimator to ranking objectives (*top-k membership*, *confidence-interval-based uncertainty*, *Kendall's τ*).

Action 2: Match Addition (`Add`). Adding a match involves selecting a candidate z_{new} and increasing its weight from $w_{\text{new}} = 0$ toward 1. Specifically, we define our candidate set as the union of the current dataset (where $w_n = 1$) and all possible pairs (where $w_n = 0$). We then identify which w_{new} to shift from zero to one based on its estimated influence on the objective:

$$\Delta f_{\text{add,new}} \approx (+1) \cdot \mathcal{I}^{(f)}_{\text{new}}, \quad \text{where} \quad \mathcal{I}^{(f)}_{\text{new}} = -\nabla_\theta f^\top H^{-1} \nabla_\theta \ell(z_{\text{new}}; \hat{\theta}) + \left(\frac{\partial f}{\partial w_{\text{new}}} \right). \tag{7}$$

This action models different levels of control over the added data, from benign data collection to stronger manipulation. We consider three candidate spaces of increasing control:

- **all_pairs:** Lowest control. The method selects an unordered pair (i, j), but the outcome is fixed by the estimated skill ordering, modeling data collection without outcome control.
- **all_outcomes_weighted:** Intermediate control. Both outcomes ($i \succ j$ and $j \succ i$) are considered separately, and each outcome's effect is weighted by its BT probability. These probability-weighted outcome scores are used for scoring and selection only.
- **all_outcomes:** Strongest control. Both outcomes are treated as separate candidates, allowing the method to select the outcome with largest effect, modeling stronger manipulation.

Action 3: Outcome Flipping (`Flip`). Reversing the outcome of match z_n to z'_n (where $y'_n = 1 - y_n$) is equivalent to simultaneously dropping the original and adding the reversed match:

$$\Delta f_{\text{flip},n} \approx \mathcal{I}^{(f)}_{z'_n} - \mathcal{I}^{(f)}_{z_n}. \tag{8}$$

We use `Flip` primarily as a counterfactual robustness and explanation diagnostic rather than as an external attack, since reversing recorded outcomes requires write access to historical data (Appendix B.10 lists the capability each action assumes), measuring how a leaderboard conclusion would change if an observed comparison had resolved in the opposite direction [Pearl, 2009, Wachter et al., 2018]. This explains which outcomes most support the current ranking and which reversals would most effectively manipulate it.

3.2 Objectives

We characterize leaderboard robustness and sensitivity through three distinct classes of objectives. These scalar functionals, $f(\theta, w)$, allow us to quantify how data perturbations propagate to specific ranking outcomes, from local skill gaps to global ranking consistency and statistical uncertainty.

Top-k Membership Objective / Gap Objective (*top-k membership*). The top-k membership objective (gap objective), following prior work on leaderboard robustness [Huang et al., 2026], measures the difference in estimated skill between two models, which drives rank ordering and top-k membership. For a target pair (i, j), it is defined as follows (derivative in Appendix B.5):

$$f_{\text{gap}}(\theta) = \theta_i - \theta_j. \tag{9}$$

Statistical Uncertainty Objectives (*confidence-interval-based uncertainty*). A confidence interval (CI) quantifies uncertainty around an estimated leaderboard skill, indicating how precisely a model's latent strength is identified from the comparison data. We use a local information approximation motivated by the BT uncertainty analysis of Gao et al. [2023], where the coordinate-wise uncertainty of the MLE is governed by the inverse local Fisher information of each player.

For each unordered pair (i, j), let $p_{ij} = \sigma(\hat{\theta}_i - \hat{\theta}_j)$ and $v_{ij} = p_{ij}(1 - p_{ij})$. Let $w_{ij} = \sum_{n:\{i_n,j_n\}=\{i,j\}} w_n$ denote the total comparison weight assigned to the unordered pair $\{i, j\}$. We define the local information of player i as $\hat{\rho}_i^2(w) = \sum_{j\neq i} w_{ij} v_{ij}$. Intuitively, $\hat{\rho}_i^2(w)$ is large when player i has many informative comparisons, leading to smaller uncertainty. Following this local-information view, we use its inverse as a scalable proxy for player-wise variance and define

$$f_{\text{CI-player}}(i; w) = \frac{1}{\hat{\rho}_i^2(w)}, \qquad f_{\text{CI-trace}}(w) = \sum_{i=1}^{M} \frac{1}{\hat{\rho}_i^2(w)}. \tag{10}$$

The first objective measures uncertainty for a target player, which we use for targeted CI reduction, while the second provides a trace-style proxy for global leaderboard uncertainty. These objectives depend both on $\hat{\theta}$ and explicitly on the comparison weights w; their corresponding direct weight derivatives are given in Appendix B.5. We discuss the relation between this Gao-style local approximation and the more general sandwich covariance estimator in Appendix B.7.

Global Ranking Consistency (*Kendall's τ*). To evaluate how perturbations affect the entire leaderboard structure, we use a smooth surrogate of *Kendall's τ* correlation against a reference ranking π (which is the ranking from BT fitted on the whole initial dataset). Since the discrete *Kendall's τ* is non-differentiable, we employ a $\tanh$-based relaxation:

$$f_{\tau,T}(\theta; \pi) = \frac{2}{M(M-1)} \sum_{a<b} s_{ab} \tanh\left(\frac{\theta_a - \theta_b}{T}\right), \tag{11}$$

where $s_{ab} \in \{-1, +1\}$ encodes the relative order of items a and b in the reference π, and T is a temperature parameter. As $T \to 0$, this objective converges to the discrete *Kendall's τ*. This differentiable form allows us to compute influence scores $\mathcal{I}_n^{(\tau,T)}$ to identify which specific comparisons most heavily affect overall ranking consistency. The derivative is given in Appendix B.5.

3.3 Player-level influence

Following Giordano et al. [2019], we extend influence functions from individual matches to structured group perturbations. For a subset $\mathcal{G} \subset \{1, \ldots, N\}$ of matches, such as all matches incident to a given player, the first-order change in the BT estimator is

$$\Delta\theta_{\mathcal{G}} \approx \sum_{n\in\mathcal{G}} \Delta w_n \frac{\partial\hat{\theta}(w)}{\partial w_n} = -H(w)^{-1} \sum_{n\in\mathcal{G}} \Delta w_n \nabla_\theta \ell(z_n; \hat{\theta}(w)). \tag{12}$$

We then propagate $\Delta\theta_{\mathcal{G}}$ through a scalar objective $f(\theta, w)$ as in Eq. 5, yielding a player-level influence score from the aggregate contribution of that player's incident matches.

For player-removal analysis, we additionally use a grouped Newton refinement: after the first-order grouped deletion step in Eq. 12, we take one Newton correction on the remaining comparison data. This is analogous to the 1sN correction for single-match perturbations, but applied jointly to all matches incident to the removed player; details are in Appendix B.4.

4 Experiments

Dataset. Our primary evaluation uses Arena 55k [LMArena AI, 2024] (64 LLMs, 55k human votes), a subset of Chatbot Arena [Chiang et al., 2024]. We additionally evaluate on six pairwise-comparison

Table 1: Minimum influence-guided actions needed to change the Top-1 model under a 5% budget. Percentages are relative to total matches; robust means no change within budget.

Dataset	Dataset size	Drop	Flip	Add-pairs	Add-outcomes	Add-weighted
Arena 55k	57,477	5 (0.01%)	3 (0.01%)	9 (0.02%)	9 (0.02%)	9 (0.02%)
Arena LLM-J	49,938	9 (0.02%)	6 (0.01%)	22 (0.04%)	21 (0.04%)	21 (0.04%)
MT-Bench	3,355	92 (2.74%)	46 (1.37%)	robust	robust	robust
NBA Top-50	109,892	24 (0.02%)	15 (0.01%)	77 (0.07%)	55 (0.05%)	57 (0.05%)
ATP Top-10	278	6 (2.16%)	3 (1.08%)	robust	9 (3.24%)	14 (5.00%)
Vision Arena	29,849	50 (0.17%)	25 (0.08%)	robust	robust	robust
WebDev Arena	10,501	179 (1.70%)	12 (0.11%)	robust	robust	robust

datasets: Chatbot Arena LLM Judges and MT-Bench Human Judgments [Zheng et al., 2023], Vision Arena [Chou et al., 2025], WebDev Arena [Vichare et al., 2025], NBA Elo Top-50 [FiveThirtyEight, 2025], and ATP Top-10 [Sackmann, 2024]. Dataset details are in Appendix A.1.

Experimental protocol. Unless otherwise stated, we greedily select perturbations using influence scores under a fixed budget, without refitting during selection. We refit after the selected perturbations to report the exact post-perturbation leaderboard and success conditions, though these checks could be monitored by influence estimates. Full algorithms are provided in Appendix B.6. We validate the influence approximations against exact BT refitting across all seven datasets, three actions, and three objectives (Appendix B.3): 1sN preserves the exact ordering of candidates (median Spearman $\rho \geq 0.92$, top-10 overlap ≥ 0.90), and its top predicted action is the exact best in 49 of 63 settings and within the exact top five in 58. Under sequential selection, fixed and adaptive scoring reach the same final objective in 19 of 21 settings, with near-zero median regret versus exact greedy selection.

4.1 Match analysis

We study how individual matches affect leaderboard outcomes along two axes: (i) a *robustness* question (how many perturbations are needed before a criterion changes?) and (ii) a *manipulation* question (can we steer ranking or confidence towards a desired outcome?).

4.1.1 Robustness analysis

***Top-k membership* robustness.** Following Huang et al. [2026], we measure the minimum number of match perturbations required to change the composition of the top-k set. We instantiate our framework with the gap objective f_{gap} and score all candidate actions by their predicted effect on crossing a top-k boundary. Starting from the original leaderboard, we greedily apply the highest-influence action until the top-k set changes or a budget of 5% of the dataset is reached.

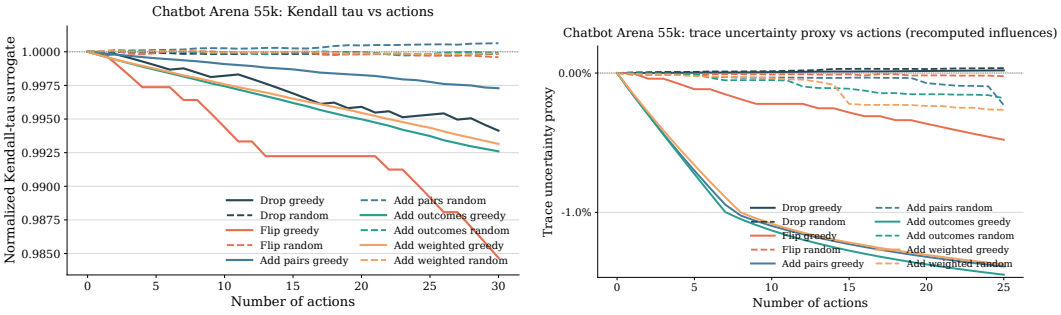
Table 1 reports the minimum influence-guided actions needed to change the Top-1 model. On major LLM leaderboards, fewer than 1% targeted actions suffice (e.g., 3 `Flip` or 5 `Drop` on Arena 55k). Across datasets, `Flip` is the most action-efficient perturbation, followed by `Drop`, while `Add` variants typically require larger budgets or remain robust within the budget. We ablate the effect of varying k on Arena 55k in Appendix E.2.

CI-aware *Top-k membership* robustness. We extend the Top-k robustness analysis to incorporate estimation uncertainty, since small rank changes may be practically meaningless when confidence intervals overlap. For a chosen boundary rank k, we augment the usual membership objective with confidence bounds: we fix the boundary pair consisting of the model currently ranked k and the model currently ranked $k+1$, and compare the upper confidence bound of the rank-k model to the lower confidence bound of the rank-$(k+1)$ model. A perturbation is declared successful only if, after refitting, the rank-$(k+1)$ model is not merely above the rank-k model in point estimate, but is *strictly* above under uncertainty as well, meaning that its lower CI bound exceeds the rank-k model's upper CI bound. This strict CI-aware criterion is stronger than an ordinary non-CI-aware (point-estimate) Top-k membership; the algorithm and an example are given in Appendices B.6 and D.3.

Table 2 compares the minimum number of influence-guided actions needed under point-estimate and CI-aware criteria, using a per-dataset boundary rank k. Accounting for uncertainty increases the required budget, but does not make the leaderboard robust. CI-aware criteria require on average

Table 2: Minimum number of actions needed to change the top-k boundary under point-estimate / CI-aware criteria (percentages relative to total matches) with 95% confidence level and a 5% perturbation budget. Across datasets, the CI-aware criterion generally requires substantially more actions than the point-estimate criterion. robust: target not changed within the 5% budget.

Dataset	Dataset size	k	Drop (point-estimate / CI-aware)	Flip (point-estimate / CI-aware)	Add-pairs	Add-outcomes	Add-weighted
Arena 55k	57,477	22	2 (0.00%) / 19 (0.03%)	1 (0.00%) / 12 (0.02%)	2 (0.00%) / 39 (0.07%)	2 (0.00%) / 26 (0.05%)	3 (0.01%) / 28 (0.05%)
Arena LLM-J	49,938	31	1 (0.00%) / 31 (0.06%)	1 (0.00%) / 17 (0.03%)	1 (0.00%) / 50 (0.10%)	1 (0.00%) / 32 (0.06%)	1 (0.00%) / 45 (0.09%)
MT-Bench	3,355	2	13 (0.39%) / 62 (1.85%)	7 (0.21%) / 33 (0.98%)	robust / robust	robust / robust	robust / robust
NBA Top-50	109,892	8	2 (0.00%) / 46 (0.04%)	1 (0.00%) / 25 (0.02%)	5 (0.00%) / robust	3 (0.00%) / robust	6 (0.01%) / robust
ATP Top-10	278	8	1 (0.36%) / robust	1 (0.36%) / 7 (2.52%)	1 (0.36%) / robust	1 (0.36%) / robust	1 (0.36%) / robust
Vision Arena	29,849	14	4 (0.01%) / 43 (0.14%)	3 (0.01%) / 33 (0.11%)	6 (0.02%) / robust	5 (0.02%) / robust	10 (0.03%) / robust
WebDev Arena	10,501	12	3 (0.03%) / 342 (3.26%)	2 (0.02%) / 23 (0.22%)	7 (0.07%) / robust	6 (0.06%) / robust	10 (0.10%) / robust



(a) Global ranking robustness (*Kendall's* τ).

(b) Trace uncertainty change (%).

Figure 2: Global ranking and uncertainty robustness on Chatbot Arena 55k under budgets of 30 actions for global ranking and 25 for uncertainty (solid = influence-guided, dashed = random). **Left:** Influence-guided `Flip` causes the largest degradation ($\tau \approx 0.985$), while `Drop` is more conservative ($\tau \approx 0.994$); random baselines remain near 1.0. **Right:** Influence-guided `Add` reduces trace uncertainty by up to $\approx -1\%$, whereas random actions slightly increase it.

(using the most efficient action) $13\times$ more actions than point-estimate changes, yet several datasets remain non-robust within the allowed budget. `Drop` and `Flip` remain the most effective actions, while `Add` variants often require larger budgets or remain robust within the budget.

Global ranking robustness (*Kendall's* τ). Beyond local boundary changes, we score candidates using the f_τ surrogate and apply actions greedily, comparing against random selection (Figure 2, left). Influence-guided `Flip` causes the steepest τ degradation, while `Drop` is noticeably more conservative and `Add` variants cluster at intermediate degradation. In all cases, random selection barely moves τ, confirming that influence-guided selection is essential: random perturbations are insufficient to disrupt global ranking even at budget 30. Results for all datasets are in Appendix D.2.

***Confidence-interval-based uncertainty* robustness.** We next evaluate how perturbations affect global leaderboard uncertainty. Using $f_{\text{CI-trace}}$ to score candidates and applying actions greedily (Figure 2, right), the action-type ordering reverses: `Add` variants yield the steepest uncertainty reduction, reaching roughly -1% of the initial trace by budget 25, about an order of magnitude larger than `Drop` or `Flip`. In contrast, random additions slightly increase uncertainty, as they tend to introduce less-informative comparisons. Results for all datasets are in Appendix D.2.

From influence scores to leaderboard audits. Beyond identifying non-robustness, we define normalized dataset-level robustness metrics that summarize influence-guided failures as audit signals. These metrics measure either the budget needed to change a leaderboard conclusion, as in Top-1, or the normalized magnitude of change, as in Kendall's τ and confidence interval stability. Appendix D.1 gives the full metric definitions. Table 3 reports these scores for all seven datasets. Lower values indicate less robustness. Based on R_{all}, MT-Bench is the most robust dataset, followed closely by NBA Top-50 and the arena-style LLM datasets, whereas ATP Top-10 and WebDev Arena are least robust overall due to lower global-ranking stability.

Top-k entry and removal. We use Top-k entry and removal as targeted manipulation tasks: promotion moves an outside model into the Top-k set, while demotion pushes an inside model out. We evaluate both under a sequentially revealed stream of candidate comparisons. For each dataset,

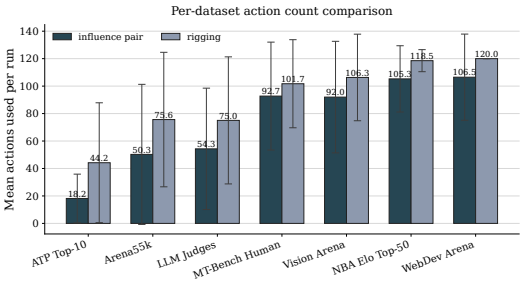


Figure 3: Mean intervention actions per run for pairwise influence versus the omni-rigging baseline. Error bars show variability across repeated trials. Each run is capped at 120 exposed pairs.

Table 3: Dataset-level robustness scores across the Top-1, CI views, and global ranking. Lower values indicate less robustness under a 5% data budget. **Bold** marks the most robust dataset.

Dataset	$R_{\text{Top-1}}$	R_{CI}	R_{τ}	R_{all}
Arena 55k	0.001	0.976	**0.993**	0.656
Arena LLM-J	0.000	0.975	0.992	0.656
MT-Bench	0.048	0.955	0.985	**0.662**
NBA Top-50	0.003	**0.992**	0.982	0.659
ATP Top-10	**0.072**	0.810	0.474	0.452
Vision Arena	0.003	0.976	0.991	0.657
WebDev Arena	0.004	0.579	0.782	0.455

we partition the leaderboard into top, middle, and lower skill regions and sample one boundary rank from each region; promotion targets rank $(k+1)$ and demotion targets rank k. We compare our influence-guided policy with an omni-rigging baseline [Min et al., 2025] under the same exposed stream, where each method may discard a pair, accept either directional outcome, or encode a tie; full details are in Appendix B.6. Our method uses a dynamic boundary-gap objective, recomputing the boundary opponent after each accepted perturbation and selecting the action with largest predicted influence on the current gap.

Figure 3 shows that the influence-guided policy typically achieves targeted promotion and demotion with fewer actions than the omni-rigging baseline across most datasets. Across datasets, influence-guided selection uses about 74 actions on average, compared with about 92 for omni-rigging, a roughly 19% reduction. This indicates that directly optimizing the current top-k boundary gap is more efficient than locally greedy rank manipulation.

CI reduction for a target model. From a leaderboard provider's perspective, we study whether the framework can constructively reduce the confidence interval of a target model m^*. We use the target uncertainty objective $f_{\text{CI-player}}(m^*)$ and restrict the action space to `Add`, selecting candidate additions with the most negative predicted influence. We compare influence-guided `Add` with the `all_pairs` candidate space against *Random* and *Arena Active* [Chiang et al., 2024]; Random samples pairs uniformly, while Arena Active prioritizes target-involving uncertain comparisons. After selecting a pair, all methods assign the outcome by the current BT skill ordering, refit after each addition, and measure target-model CI reduction.

For each dataset, we choose three target players from the top, middle, and lower skill regions and run each method for 12 `Add` actions. Table 5 reports $\%\Delta\text{CI} = 100 \times (\text{final} - \text{initial})/\text{initial}$, averaged over targets; more negative values indicate larger uncertainty reduction. Averaged over all results, influence-guided sampling produces a larger CI-width reduction than *Arena Active* and *Random*, showing that directly optimizing the target CI objective outperforms Arena-style heuristics.

From audits to defenses: certificate-gated reporting. The same audit can gate what a leaderboard publishes. After each data batch, we refit BT; if the internal Top-1 differs from the last published Top-1, we compute $b^\star_{\text{flip}}$, the minimum number of influence-ranked `Flip` actions that reverse the proposed update, and publish only if $b^\star_{\text{flip}} \geq \kappa$ for an operator-chosen threshold κ ($\kappa = 5$ in our experiments). We replay Arena 55k as a stream of 1,000-vote batches over 10 fixed random permutations; since the data carry no timestamps, this measures sampling-induced churn rather than temporal drift. A published change is *false* if the Top-1 reverts to the previously published model within three snapshots. Table 4 compares the gate with publishing every change (Always Publish) and with publishing a new Top-1 only after it persists for three consecutive snapshots (Persistence-3). The gate removes all false changes at a latency similar to Persistence-3, at the cost of withholding more updates. When an update is withheld, influence-guided `Add` collection over 50 held-out comparisons certifies 43% of withheld updates (15.6 comparisons on average), versus 17% (26.0) for Arena Active and 0% for Random. Protocol, metric definitions, and additional results are in Appendix D.4.

Table 4: Certificate-gated Top-1 reporting on Arena 55k (1,000-vote batches, 10 fixed random permutations, pooled; gate uses $\kappa = 5$). Latency is in batches.

Policy	Published	False change	Latency	Withheld	Missed
Always Publish	54	48.1%	0.0	0.000	0.000
Persistence-3	17	29.4%	2.0	0.553	0.294
Robustness gate (ours)	8	**0.0%**	2.5	0.750	0.467

Table 5: Percent change in target-model CI width after 12 targeted actions, $\%\Delta\mathrm{CI} = 100 \times (\mathrm{final} - \mathrm{initial})/\mathrm{initial}$. More negative is better.

Dataset	Influence-based	Arena Active	Random
MT-Bench	**-0.27 ± 0.05**	-0.25 ± 0.07	-0.08 ± 0.03
WebDev Arena	-0.28 ± 0.05	**-0.39 ± 0.23**	-0.02 ± 0.00
Vision Arena	**-0.17 ± 0.12**	-0.16 ± 0.11	-0.01 ± 0.01
Arena LLM-J	**-0.47 ± 0.25**	-0.30 ± 0.11	-0.00 ± 0.00
Arena 55k	**-0.72 ± 0.22**	-0.54 ± 0.28	-0.01 ± 0.01
NBA Top-50	**-0.33 ± 0.09**	-0.29 ± 0.05	-0.01 ± 0.00
All datasets	**-0.37 ± 0.13**	-0.32 ± 0.14	-0.02 ± 0.01

Table 6: Most-influential-player ablation results. We remove the player with largest influence and all associated matches, then measure changes.

Dataset	$\Delta\tau$	Moved	Max Shift	Top-10	Removed %
ATP Top-10	-0.167	2	2	2	24.1%
MT-Bench	0.000	0	0	0	30.3%
WebDev	-0.091	5	2	4	23.2%
Vision	-0.050	2	2	0	14.1%
Arena LLM-J	-0.023	25	5	4	6.2%
Arena 55k	-0.030	28	10	5	6.2%
NBA Top-50	-0.053	27	13	6	0.5%

4.2 Player removal analysis

Beyond individual match perturbations, we consider player removal: dropping all matches of a given model, as occurs when models are retired, deprecated, or excluded from evaluation. Prior work on the leaderboard illusion [Singh et al., 2025] shows that excluding models can qualitatively change which remaining models appear superior; we quantify this effect systematically. We use our influence framework to rank all players by predicted $|\tau|$-influence and validate by removing the most influential player in each dataset and measuring the resulting *Kendall's* τ shift. Table 6 shows that removing an influential player can induce broad reordering, moving up to 28 players, shifting ranks by up to 13 positions, and changing as many as 6 top-10 memberships. We analyze which structural features predict removal impact in Appendix E.3; connectivity is the strongest and most consistent predictor.

5 Related work

Ranking systems and evaluation leaderboards. The Bradley–Terry model [Bradley and Terry, 1952, Hamilton et al., 2025, Fang et al., 2026] and related models like Elo rating systems [Boubdir et al., 2024, Liu et al., 2025] form the basis of modern probabilistic ranking. Maximum-likelihood estimation and uncertainty quantification for BT models are well-studied [Hunter, 2004, Bong and Rinaldo, 2022, Gao et al., 2023, Fan et al., 2024], with spectral and sorting-based methods providing efficient alternatives [Negahban et al., 2017, Wauthier et al., 2013]. In LLM evaluation, pairwise preference aggregation has become central through Chatbot Arena [Chiang et al., 2024], with recent work improving statistical reliability [Ameli et al., 2025, Gao et al., 2025] and studying evaluation bias [Daynauth et al., 2025, Levtsov and Ustalov, 2025].

Leaderboard robustness and manipulation. Benchmark rankings can be sensitive to dataset shifts, benchmark overfitting, and small comparison perturbations [Boubdir et al., 2024, Huang et al., 2026, Singh et al., 2025]. The leaderboard illusion [Singh et al., 2025] shows that benchmark rankings can be distorted by selective disclosure, private testing, uneven data allocation, and model inclusion or exclusion decisions, while vote manipulation and adversarial leaderboard attacks expose additional manipulation risks [Min et al., 2025, Huang et al., 2025, Suri et al., 2025]. Related theory studies robustness of estimators to sample removal [Azar et al., 2025, Broderick et al., 2023], and recent uncertainty-aware ranking methods propose rank sets or confidence diagrams [Chatzi et al., 2024, Wang et al., 2025]. Our work provides a unified framework covering match addition, removal, and flipping, extending prior work that typically studies one *perturbation type* or one objective.

Explainability and data attribution. Data attribution is a central tool for explaining model behavior by assigning importance scores to individual training examples or groups of examples. Existing approaches include Data Shapley [Ghorbani and Zou, 2019], TracIn [Pruthi et al., 2020], and TRAK [Park et al., 2023], which quantify how training data contributes to model predictions or learned

representations; see Hammoudeh and Lowd [2024] for a survey. Influence functions are another classical approach to data attribution, estimating the effect of training points on model parameters without retraining [Koh and Liang, 2017, Pregibon, 1981], with roots in robust statistics [Huber, 1967, Freedman, 2006]. Extensions include the infinitesimal jackknife [Giordano et al., 2019], group influence methods [Koh et al., 2019], second-order approximations [Basu et al., 2020], and analyses of when influence estimates are reliable [Bae et al., 2022]. We build on this line of work by propagating influence scores to leaderboard-specific objectives.

6 Conclusion, limitations, and future work

We introduce a unified influence-based perturbation framework for evaluating Bradley–Terry leaderboard stability and manipulation. Across seven datasets, we show that leaderboard conclusions are fragile: Top-1 membership changes with fewer than 1% targeted actions on major LLM leaderboards, CI-aware Top-k changes require larger budgets but remain vulnerable, and removing highly connected players can induce broad rank shifts. Our normalized dataset-level robustness scores summarize these effects across Top-k, CI-aware, and global-ranking views. Overall, expert-curated MT-Bench is generally more robust, while crowdsourced arena-style leaderboards are more sensitive; among actions, `Flip` is usually most effective for changing rankings, whereas `Add` is most useful for reducing uncertainty through targeted data collection. Beyond auditing, the same framework enables targeted manipulation: influence-guided actions promote or demote selected models with fewer interventions than the omni-rigging baseline, while supporting constructive uncertainty reduction through comparisons that narrow confidence intervals more effectively than Arena-style sampling.

Our analysis has several limitations. Influence estimates are local approximations and may be less accurate under large perturbation budgets or highly nonlinear ranking changes. Our uncertainty objectives use scalable BT-based proxies, which do not capture all sources of annotation noise, judge bias, prompt-level dependence, or model-specific evaluation artifacts; the Gao-style local proxy closely tracks sandwich-based CI certification on most datasets but can be optimistic on sparse comparison graphs such as ATP (Appendix B.8). Future work should study higher-order attribution methods, outlier-robust BT estimators, and sample-complexity guarantees for leaderboard stability. Another important direction is to better understand what makes a match, player, or dataset highly influential or robust. Finally, the framework also serves as a defensive tool: low robustness certificates gate leaderboard updates and guide targeted data collection (Section 4.1.1), with highly influential comparisons treated as signals of fragile conclusions rather than as evidence of invalid data.

Acknowledgements and Disclosures

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the O'Donovan Chair in Trustworthy AI.

References

Siavash Ameli, Siyuan Zhuang, Ion Stoica, and Michael W. Mahoney. A statistical framework for ranking LLM-based chatbots. In *The Thirteenth International Conference on Learning Representations*, 2025.

Eyar Azar, Michael J. Feldman, and Boaz Nadler. Robustness of OLS to sample removals: Theoretical analysis and implications. arXiv preprint arXiv:2512.23069, 2025.

Juhan Bae, Nathan Ng, Alston Lo, Marzyeh Ghassemi, and Roger Grosse. If influence functions are the answer, then what is the question? In *Advances in Neural Information Processing Systems*, volume 35, pages 17953–17967, 2022.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv preprint arXiv:2204.05862, 2022.

Samyadeep Basu, Xuchen You, and Soheil Feizi. On second-order group influence functions for black-box predictions. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 715–724. PMLR, 2020.

Heejong Bong and Alessandro Rinaldo. Generalized results for the existence and consistency of the MLE in the Bradley–Terry–Luce model. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2160–2177. PMLR, 2022.

Meriem Boubdir, Edward Kim, Beyza Ermis, Sara Hooker, and Marzieh Fadaee. Elo uncovered: Robustness and best practices in language model evaluation. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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. doi: 10.2307/2334029.

Tamara Broderick, Ryan Giordano, and Rachael Meager. An automatic finite-sample robustness metric: When can dropping a little data make a big difference? arXiv preprint arXiv:2011.14999, 2023.

Ivi Chatzi, Eleni Straitouri, Suhas Thejaswi, and Manuel Gomez Rodriguez. Prediction-powered ranking of large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

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*, pages 8359–8388. PMLR, 2024.

Christopher Chou, Lisa Dunlap, Wei-Lin Chiang, Koki Mashita, Krishna Mandal, Trevor Darrell, Ion Stoica, and Joseph E. Gonzalez. VisionArena: 230k real world user-VLM conversations with preference labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. Learning how hard to think: Input-adaptive allocation of LM computation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Roland Daynauth, Christopher Clarke, Krisztian Flautner, Lingjia Tang, and Jason Mars. Ranking unraveled: Recipes for LLM rankings in head-to-head AI combat. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26078–26091. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1265.

Jianqing Fan, Jikai Hou, and Mengxin Yu. Uncertainty quantification of MLE for entity ranking with covariates. *Journal of Machine Learning Research*, 25(358):1–83, 2024.

Shuxing Fang, Ruijian Han, Yuanhang Luo, and Yiming Xu. Recent advances in the Bradley–Terry model: Theory, algorithms, and applications. arXiv preprint arXiv:2601.14727, 2026.

FiveThirtyEight. NBA-ELO dataset. `https://github.com/fivethirtyeight/data/tree/master/nba-elo`, 2025. Accessed 2026-04-30.

David A. Freedman. On the so-called "Huber sandwich estimator" and "robust standard errors". *The American Statistician*, 60(4):299–302, 2006. doi: 10.1198/000313006X152207.

Evan Frick, Connor Chen, Joseph Tennyson, Tianle Li, Wei-Lin Chiang, Anastasios N. Angelopoulos, and Ion Stoica. Prompt-to-leaderboard. arXiv preprint arXiv:2502.14855, 2025.

Chao Gao, Yandi Shen, and Anderson Y. Zhang. Uncertainty quantification in the Bradley–Terry–Luce model. *Information and Inference: A Journal of the IMA*, 12(2):1073–1140, 2023.

Mingqi Gao, Yixin Liu, Xinyu Hu, Xiaojun Wan, Jonathan Bragg, and Arman Cohan. Re-evaluating automatic LLM system ranking for alignment with human preference. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4605–4629. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.260.

Amirata Ghorbani and James Zou. Data Shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2242–2251. PMLR, 2019.

Ryan Giordano, William Stephenson, Runjing Liu, Michael I. Jordan, and Tamara Broderick. A Swiss Army infinitesimal jackknife. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1139–1147. PMLR, 2019.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Supryadi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, and Deyi Xiong. Evaluating large language models: A comprehensive survey. arXiv preprint arXiv:2310.19736, 2023.

Ian Hamilton, Nick Tawn, and David Firth. The many routes to the ubiquitous Bradley–Terry model. arXiv preprint arXiv:2312.13619, 2025.

Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: A survey. *Machine Learning*, 113(5):2351–2403, 2024. doi: 10.1007/s10994-023-06495-7.

Jenny Y. Huang, Yunyi Shen, Dennis Wei, and Tamara Broderick. Dropping just a handful of preferences can change top large language model rankings. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yangsibo Huang, Milad Nasr, Anastasios Nikolas Angelopoulos, Nicholas Carlini, Wei-Lin Chiang, Christopher A. Choquette-Choo, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Ken Ziyu Liu, Ion Stoica, Florian Tramèr, and Chiyuan Zhang. Exploring and mitigating adversarial manipulation of voting-based leaderboards. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 25654–25671. PMLR, 2025.

Peter J. Huber. The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 221–234. University of California Press, 1967.

David R. Hunter. MM algorithms for generalized Bradley–Terry models. *The Annals of Statistics*, 32 (1):384–406, 2004.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 2017.

Pang Wei Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Miles Kruppa. The UC Berkeley project that is the AI industry's obsession. The Wall Street Journal, `https://www.wsj.com/tech/ai/the-uc-berkeley-project-that-is-the-ai-industrys-obsession-bc68b3e3`, December 2024. Accessed 2026-04-30.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 26874–26901. PMLR, 2024.

Georgii Levtsov and Dmitry Ustalov. Confidence and stability of global and pairwise scores in NLP evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 40–52. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-srw.3.

Zirui Liu, Jiatong Li, Yan Zhuang, Qi Liu, Shuanghong Shen, Jie Ouyang, Mingyue Cheng, and Shijin Wang. am-ELO: A stable framework for arena-based LLM evaluation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 38857–38868. PMLR, 2025.

LMArena AI. Arena human preference 55k dataset. https://huggingface.co/datasets/lmarena-ai/arena-human-preference-55k, 2024. Hugging Face dataset. Accessed 2026-04-30.

Rachel Metz. Before DeepSeek blew up, Chatbot Arena announced its arrival. Bloomberg Businessweek, https://www.bloomberg.com/news/articles/2025-02-18/before-deepseek-blew-up-one-website-announced-its-arrival, February 2025. Accessed 2026-04-30.

Rui Min, Tianyu Pang, Chao Du, Qian Liu, Minhao Cheng, and Min Lin. Improving your model ranking on Chatbot Arena by vote rigging. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 44252–44271. PMLR, 2025.

Mihran Miroyan, Tsung-Han Wu, Logan King, Tianle Li, Jiayi Pan, Xinyan Hu, Wei-Lin Chiang, Anastasios Nikolas Angelopoulos, Trevor Darrell, Narges Norouzi, and Joseph E. Gonzalez. Search Arena: Analyzing search-augmented LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 65(1):266–287, 2017. doi: 10.1287/opre.2016.1534.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 27074–27113. PMLR, 2023.

Judea Pearl. Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146, 2009. doi: 10.1214/09-SS057.

Daryl Pregibon. Logistic regression diagnostics. *The Annals of Statistics*, 9(4):705–724, 1981.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pages 19920–19930, 2020.

Vyas Raina, Adian Liusie, and Mark Gales. Is LLM-as-a-judge robust? Investigating universal adversarial attacks on zero-shot LLM assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7499–7517. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.427.

Jeff Sackmann. ATP tennis rankings, results, and stats. https://github.com/JeffSackmann/tennis_atp, 2024. GitHub repository. Accessed 2026-04-30.

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.

Jack Sherman and Winifred J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950. doi: 10.1214/aoms/1177729893.

Shivalika Singh, Yiyang Nan, Alex Wang, Daniel D'Souza, Sayash Kapoor, Ahmet Üstün, Sanmi Koyejo, Yuntian Deng, Shayne Longpre, Noah A. Smith, Beyza Ermis, Marzieh Fadaee, and Sara Hooker. The leaderboard illusion. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025.

Hao Sun, Yunyi Shen, and Jean-François Ton. Rethinking reward modeling in preference-based large language model alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.

Anshuman Suri, Harsh Chaudhari, Yuefeng Peng, Ali Naseh, Amir Houmansadr, and Alina Oprea. Exploiting leaderboards for large-scale distribution of malicious models. arXiv preprint arXiv:2507.08983, 2025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023.

Aryan Vichare, Anastasios N. Angelopoulos, Wei-Lin Chiang, Kelly Tang, and Luca Manolache. WebDev Arena: A live LLM leaderboard for web app development. LMArena blog post, `https://news.lmarena.ai/webdev-arena`, 2025. Accessed 2026-10-31.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2): 841–887, 2018.

Zebin Wang, Yi Han, Ethan X. Fang, Lan Wang, and Junwei Lu. Confidence diagram of nonparametric ranking for uncertainty assessment in large language models evaluation. arXiv preprint arXiv:2412.05506, 2025.

Fabian Wauthier, Michael Jordan, and Nebojsa Jojic. Efficient ranking from pairwise comparisons. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 109–117. PMLR, 2013.

Weichen Wu, Nynke Niezink, and Brian Junker. A diagnostic framework for the Bradley–Terry model. *Journal of the Royal Statistical Society, Series A: Statistics in Society*, 185(Supplement 2): S461–S484, 2022. doi: 10.1111/rssa.12959.

Sheng Xu, Bo Yue, Hongyuan Zha, and Guiliang Liu. Uncertainty-aware preference alignment in reinforcement learning from human feedback. In *ICML 2024 Workshop on Models of Human Feedback for AI Alignment*, 2024.

Wenting Zhao, Alexander M. Rush, and Tanya Goyal. Challenges in trustworthy human evaluation of chatbots. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3359–3365. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.186.

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, pages 46595–46623, 2023.

Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. Cheating automatic LLM benchmarks: Null models achieve high win rates. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Datasets

A.1 Dataset descriptions

We evaluate our framework on seven pairwise comparison datasets spanning LLM evaluation and sports ranking.

Chatbot Arena 55k. A crowdsourced platform where users simultaneously interact with two anonymous chatbots and vote for the preferred response [Chiang et al., 2024]. We use the `arena-human-preference-55k` split, containing 57,477 human preference judgments across 64 models. Its large scale and diverse user base make it the primary benchmark for LLM evaluation, but also expose it to noise and adversarial risk.

Chatbot Arena LLM Judges. A companion dataset from the same platform in which pairwise preferences are collected using an LLM-as-a-judge rather than human votes [Chiang et al., 2024]. The `chatbot-arena-llm-judges` split contains 49,938 comparisons across 64 models, allowing us to contrast automated and human evaluation robustness.

MT-Bench Human Judgments. A curated multi-turn benchmark designed to evaluate instruction-following and reasoning [Zheng et al., 2023]. Preferences were collected from 58 expert-level annotators (predominantly graduate students), yielding 3,355 high-quality pairwise judgments. Its smaller size and expert annotations make it substantially more robust than crowdsourced platforms.

Vision Arena. A crowdsourced arena for vision-language models in which users compare two anonymous models on visual question-answering tasks. We use the 29,849 single- and multi-turn conversations in `lmarena-ai/VisionArena-Battle`.

WebDev Arena. A crowdsourced arena focused on web-development tasks such as building interactive applications and webpages. We use the `lmarena-ai/webdev-arena-preference-10k` dataset (10,501 prompts), which complements open-ended chat with domain-specific evaluation.

ATP Top-10. Match records from the ATP tour (2020–2024). We restrict to the top-10 ranked players by 2024 season standing who each played at least 20 matches, yielding 278 games in total. This small, sparse graph tests our framework in a regime where each match carries high weight.

NBA Elo Top-50. Historical NBA game records from all seasons. We focus on the top-50 teams by total games played, yielding 109,892 matchups. Its large, dense comparison graph represents the opposite extreme from ATP and assesses robustness in high-data settings.

Licenses. Arena 55k is released under Apache-2.0; MT-Bench Human Judgments under CC BY 4.0; Vision Arena (`VisionArena-Battle`) and WebDev Arena (`webdev-arena-preference-10k`) under the LMArena dataset license agreements accepted on their Hugging Face pages; the FiveThirtyEight NBA Elo data under CC BY 4.0; and the ATP data of Sackmann [2024] under CC BY-NC-SA 4.0, which permits our non-commercial research use. We use all datasets for research only and do not redistribute them.

A.2 Dataset processing

Each pairwise comparison in the raw datasets is represented as a directed observation (i, j, y) where $y \in \{0, 1\}$ indicates whether item i beat item j. To handle ties uniformly, we adopt a symmetric formulation: each tied comparison is decomposed into two directed observations—$(i, j, 1)$ and $(j, i, 1)$—treating both items as winning once against the other.

For consistency, each non-tied comparison is also represented as two directed observations: (i, j, y) and $(j, i, 1 - y)$. This ensures a uniform representation across all comparisons and simplifies the construction of the feature matrix $X \in \mathbb{R}^{N \times M}$ used in the BT likelihood, where each row $x_n = e_i - e_j$ encodes the matched pair. The resulting matrix X and outcome vector y are the direct inputs to the weighted M-estimator in Eq. 3.

B Framework details

B.1 BT input–output format and tie handling

The Bradley–Terry model takes a directed comparison dataset as input and outputs a latent skill vector for the ranked players. Let M denote the number of players and let $e_i \in \mathbb{R}^M$ be the standard basis vector for player i. Each directed comparison between players i and j is encoded as

$$x_{ij} = e_i - e_j,$$

where the binary outcome $y_{ij} \in \{0, 1\}$ indicates whether the first player in the directed pair wins. After preprocessing, the full BT input is represented as

$$X = \begin{bmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_N^\top \end{bmatrix} \in \{-1, 0, 1\}^{N \times M}, \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \in \{0, 1\}^N,$$

where each row $x_n = e_{i_n} - e_{j_n}$ contains one $+1$ entry for the first player, one -1 entry for the second player, and zeros elsewhere.

Given this input, the BT model estimates a skill vector

$$\hat{\theta} = \begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_M \end{bmatrix} \in \mathbb{R}^M,$$

where larger values indicate stronger players. The main BT outputs are the fitted skill vector $\hat{\theta}$, the induced pairwise win probabilities, and the resulting leaderboard obtained by sorting players according to $\hat{\theta}$.

To keep the representation symmetric, every raw comparison is converted into a two-row directed block. A decisive comparison in which player i beats player j is represented as

$$i \succ j \quad \mapsto \quad B_b = \{(x_{ij}, 1), (x_{ji}, 0)\}.$$

A tied comparison is represented as

$$i \sim j \quad \mapsto \quad B_b = \{(x_{ij}, 1), (x_{ji}, 1)\}.$$

Thus, decisive outcomes provide one win and one loss across the two directions, whereas ties assign equal win evidence to both players. This avoids choosing an arbitrary winner for tied comparisons while keeping all raw comparisons in the same binary directed BT input format.

For influence scoring and perturbation selection, we treat each two-row block B_b as the atomic unit. If the two directed rows corresponding to raw comparison b are indexed by n_{ij} and n_{ji}, its block-level influence is computed by summing the row-level influences,

$$I^{(f)}_{B_b} = I^{(f)}_{n_{ij}} + I^{(f)}_{n_{ji}}.$$

All perturbation actions are then applied at the block level. A `Drop` action removes both directed rows in B_b; an `Add` action inserts both directed rows for the candidate comparison; and a `Flip` action modifies the two directed outcomes consistently. This ensures that influence scores and perturbation budgets are defined at the level of the original raw comparisons, rather than at the level of individual directed rows.

B.2 One-step Newton refinement

The influence approximation in Eq. 5 linearizes the effect of an infinitesimal change in a match weight. However, the actions used in our experiments are finite. Following the classical one-step case-deletion approximation for logistic regression diagnostics [Pregibon, 1981] and its use in influence-based leaderboard analysis [Huang et al., 2026], we use a one-step Newton (1sN) correction to better approximate these finite actions without fully refitting the BT model for every candidate.

For a comparison $z_n = (x_n, y_n)$ evaluated at the full-data fit $\hat{\theta}$, let

$$p_n = \sigma(x_n^\top \hat{\theta}), \qquad r_n = y_n - p_n, \qquad v_n = p_n(1 - p_n).$$

Since

$$g_n = \nabla_\theta \ell(z_n; \hat{\theta}) = (p_n - y_n)x_n = -r_n x_n,$$

the first-order deletion approximation from Eq. 4 gives

$$\Delta\hat{\theta}^{\texttt{Drop}}_{n,\mathrm{IF}} = -r_n H^{-1} x_n.$$

For adding a candidate comparison $z_c = (x_c, y_c)$, the sign is reversed:

$$\Delta\hat{\theta}^{\texttt{Add}}_{c,\mathrm{IF}} = r_c H^{-1} x_c.$$

The influence approximation uses the original Hessian H and therefore ignores the fact that a finite `Drop` or `Add` action also changes the local curvature of the BT objective. For a single comparison, this curvature change is rank-one because the Hessian contribution of z_n is

$$H_n = v_n x_n x_n^\top.$$

Thus, dropping one comparison changes the local Hessian from H to $H - H_n$, while adding one comparison changes it from H to $H + H_n$. Applying the Sherman–Morrison [Sherman and Morrison, 1950] identity to this rank-one Hessian update yields the leverage term

$$h_n = v_n x_n^\top H^{-1} x_n.$$

This quantity measures how strongly comparison n affects the local curvature at the BT fit.

The 1sN refinement rescales the influence update by this leverage term. For a single dropped comparison,

$$\Delta\hat{\theta}^{\texttt{Drop}}_{n,\mathrm{1sN}} = \frac{\Delta\hat{\theta}^{\texttt{Drop}}_{n,\mathrm{IF}}}{1 - h_n},$$

whereas for a single added candidate comparison,

$$\Delta\hat{\theta}^{\texttt{Add}}_{c,\mathrm{1sN}} = \frac{\Delta\hat{\theta}^{\texttt{Add}}_{c,\mathrm{IF}}}{1 + h_c}.$$

Equivalently, 1sN performs one Newton step toward the optimum of the perturbed objective, starting from the original fitted parameters $\hat{\theta}$. Compared with influence, which keeps the curvature fixed at H, 1sN partially accounts for the curvature change induced by the finite action without a full BT refit.

Given either the IF or 1sN parameter change, the predicted objective change is computed using the same projection rule as Eq. 5:

$$\Delta f \approx \nabla_\theta f(\hat{\theta}, w)^\top \Delta\hat{\theta} + \left(\frac{\partial f}{\partial w}\right)_{\text{explicit}}.$$

The explicit term is zero for objectives that depend on the data only through $\hat{\theta}$, and nonzero for uncertainty objectives that directly depend on the weighted comparison graph.

For `Flip` actions, the local Hessian is unchanged because it depends only on the comparison features x_n and not on the outcome y_n. As a result, the 1sN correction does not apply, and the flip update reduces to its first-order influence approximation.

B.3 Validation against exact refitting

We test whether IF and 1sN reproduce the effects of finite actions computed by exact BT refitting.

Single-action fidelity. For each of the seven datasets, three actions (`Drop`, `Add`, `Flip`), and three objectives (Top-1 gap, global CI width, smooth *Kendall's* τ), we sample up to 100 candidate actions, compute the predicted objective change with IF and with 1sN, and compare it with the exact change after a full BT refit. We report Spearman's ρ between predicted and exact changes, the overlap between the predicted and exact top-10 candidates, and the relative L_1 error $\text{rel-}L_1 = \sum_i |\hat{\Delta}_i - \Delta_i^{\text{exact}}| \big/ \sum_i |\Delta_i^{\text{exact}}|$ over the sampled candidates. Table 7 reports medians across datasets, with the minimum Spearman correlation in parentheses. Across the 63 settings, the top predicted action is the exact best in 49 and lies in the exact top five in 58.

Table 7: Single-action fidelity of IF / 1sN against exact refitting. Entries are medians across datasets (minimum Spearman ρ in parentheses).

Objective	Action	Spearman ρ	Top-10 overlap	Rel. L_1 error
Top-1 gap	`Drop`	0.95 (0.71) / 0.95 (0.71)	1.00 / 1.00	0.054 / 0.054
	`Add`	0.92 (0.65) / 0.92 (0.65)	0.90 / 0.90	0.215 / 0.215
	`Flip`	0.97 (0.78) / 0.97 (0.78)	1.00 / 1.00	0.037 / 0.037
Global CI width	`Drop`	1.00 (0.79) / 1.00 (0.79)	1.00 / 1.00	0.053 / 0.055
	`Add`	0.99 (0.86) / 0.99 (0.91)	1.00 / 1.00	0.045 / 0.046
	`Flip`	0.99 (0.84) / 0.99 (0.85)	0.90 / 0.90	0.126 / 0.119
Smooth *Kendall's* τ	`Drop`	0.99 (0.75) / 0.99 (0.75)	0.90 / 0.90	0.163 / 0.163
	`Add`	0.99 (0.78) / 0.99 (0.78)	1.00 / 1.00	0.077 / 0.076
	`Flip`	0.99 (0.94) / 0.99 (0.94)	0.90 / 0.90	0.155 / 0.155

Multi-action fidelity. To test whether errors accumulate under sequential selection, we select up to $B = 10$ `Drop` actions with three policies: *Fixed* (F) computes approximate scores once and keeps them fixed; *Adaptive* (A) refits and recomputes approximate scores after every action; and *Exact greedy* (E) fully refits every remaining candidate at each step and selects the exact best. Overlap is the fraction of selected actions also chosen by E, and regret is the difference in final objective from E. Table 8 shows that F and A reach the same final objective in 19 of 21 settings with near-zero median regret; recomputing after each action raises overlap only from 0.85 to 0.87, at much higher cost.

Table 8: Multi-action fidelity under sequential `Drop` selection: Fixed (F) and Adaptive (A) influence scoring versus Exact greedy (E).

Objective	A–E overlap	F–E overlap	Same final A=F	Median regret A / F
Top-1 gap	0.93	0.93	7/7	0 / 0
CI width	0.80	0.80	6/7	2.5×10^{-6} / 1.2×10^{-6}
Kendall's τ	0.87	0.81	6/7	3.1×10^{-7} / 2.3×10^{-6}
Overall	0.87	0.85	19/21	–

Fidelity is not uniform: the largest deviations occur for `Add` (worst-case Spearman 0.65 on the Top-1 gap) and in weakly supported settings, where 1sN can improve on IF (e.g., `Add` on CI width, 0.86 $\rightarrow$ 0.91). We therefore use influence scores only to rank and select actions, and verify every reported outcome by exact refitting.

B.4 Grouped Newton refinement for player removal

The 1sN correction in Appendix B.2 is applied to individual `Drop` or `Add` actions. Player removal is a larger structured perturbation: for a player m, it removes the full set of incident comparisons

$$\mathcal{G}_m = \{n : z_n \text{ contains player } m\}.$$

Because many comparisons are removed simultaneously, we use a grouped Newton refinement rather than applying the single-row leverage correction independently to each match.

Let

$$p_n = \sigma(x_n^\top \hat{\theta}), \qquad r_n = y_n - p_n, \qquad s_n = r_n x_n$$

denote the fitted probability, residual, and score contribution of comparison n at the full-data solution. For a group deletion $\mathcal{G}$, the first step sums the removed score contributions and applies the grouped first-order deletion update:

$$s_{\mathcal{G}} = \sum_{n\in\mathcal{G}} s_n, \qquad \theta_{\mathcal{G}}^{(1)} = \hat{\theta} - H^{-1} s_{\mathcal{G}}.$$

This is the group analogue of summing individual deletion influences in Eq. 12.

The second step takes one Newton correction using only the comparisons that remain after the group is removed. Let

$$\mathcal{K} = \{1, \dots, N\} \setminus \mathcal{G}$$

be the kept set. At $\theta_{\mathcal{G}}^{(1)}$, define the kept-data score

$$S_{\mathcal{K}}(\theta_{\mathcal{G}}^{(1)}) = \sum_{n\in\mathcal{K}} x_n \Big(y_n - \sigma(x_n^\top \theta_{\mathcal{G}}^{(1)}) \Big),$$

and the kept-data Hessian

$$H_{\mathcal{K}}(\theta_{\mathcal{G}}^{(1)}) = \sum_{n\in\mathcal{K}} v_n^{(1)} x_n x_n^\top + \lambda I, \qquad v_n^{(1)} = p_n^{(1)}(1 - p_n^{(1)}),$$

where

$$p_n^{(1)} = \sigma(x_n^\top \theta_{\mathcal{G}}^{(1)}).$$

The grouped Newton estimate is then

$$\theta_{\mathcal{G}}^{(2)} = \theta_{\mathcal{G}}^{(1)} + H_{\mathcal{K}}(\theta_{\mathcal{G}}^{(1)})^{-1} S_{\mathcal{K}}(\theta_{\mathcal{G}}^{(1)}).$$

Finally, the predicted player-removal effect is evaluated by applying the downstream objective to this approximate parameter vector:

$$\Delta f_{\mathcal{G}} \approx f(\theta_{\mathcal{G}}^{(2)}) - f(\hat{\theta}).$$

For the player-removal experiments, f is the smooth *Kendall's* τ objective computed over the remaining players, so the removed player is excluded from the reference ranking.

This refinement is related to 1sN because both start from an influence-based finite-deletion step and then use local curvature information to improve the approximation. The difference is that 1sN uses a scalar leverage correction for a single comparison, whereas the grouped Newton refinement performs one joint Newton correction on the score equation of the remaining comparison data.

B.5 Objective influence derivations

The objective-level influence of a match z_n is given by Eq. 5 and is defined as the derivative of the objective with respect to increasing the match weight w_n. Thus, for a finite `Drop` action, where $\Delta w_n = -1$, the predicted drop effect is

$$\Delta f_{\mathrm{drop},n} \approx -\mathcal{I}_n^{(f)}.$$

Equivalently, specializing to first-order case-deletion, the parameter shift induced by dropping match n is

$$\Delta\theta_n^{\mathrm{drop}} \approx -\, r_n H^{-1} x_n, \qquad r_n = y_n - p_n, \qquad p_n = \sigma(x_n^\top \hat{\theta}),$$

and the corresponding drop effect is

$$\Delta f_{\mathrm{drop},n}^{(f)} \approx \nabla_\theta f(\hat{\theta}, w)^\top \Delta\theta_n^{\mathrm{drop}} - \left(\frac{\partial f}{\partial w_n}\right)_{\mathrm{explicit}}.$$

Therefore,

$$\mathcal{I}_n^{(f)} \approx -\Delta f_{\mathrm{drop},n}^{(f)}.$$

For objectives that depend on the data only through $\hat{\theta}$, the explicit term is zero.

Top-k membership objective / gap objective. For

$$f_{\mathrm{gap}}(\theta) = \theta_i - \theta_j,$$

the full gradient is

$$\nabla_\theta f_{\mathrm{gap}} = e_i - e_j,$$

and there is no explicit weight term, so the drop effect is

$$\Delta f_{\mathrm{drop},n}^{(f_{\mathrm{gap}})} = \nabla_\theta f_{\mathrm{gap}}(\hat{\theta})^\top \Delta\theta_n^{\mathrm{drop}}.$$

Equivalently,

$$\mathcal{I}_n^{(f_{\mathrm{gap}})} = -\nabla_\theta f_{\mathrm{gap}}(\hat{\theta})^\top \Delta\theta_n^{\mathrm{drop}}.$$

Player-uncertainty objective. The single-player uncertainty proxy is

$$f_{\text{CI-player}}(m; w) = \frac{1}{\hat{\rho}_m^2(w)}, \qquad \hat{\rho}_m^2(w) = \sum_{j \neq m} w_{mj}\, v_{mj},$$

with

$$v_{mj} = \sigma(\hat{\theta}_m - \hat{\theta}_j)\big(1 - \sigma(\hat{\theta}_m - \hat{\theta}_j)\big).$$

Define

$$v'_{mj} = v_{mj}\big(1 - 2\sigma(\hat{\theta}_m - \hat{\theta}_j)\big).$$

Then

$$\frac{\partial \hat{\rho}_m^2}{\partial \theta_k} = \sum_{j \neq m} w_{mj}\, v'_{mj}\, (\mathbf{1}_{k=m} - \mathbf{1}_{k=j}),$$

so

$$\nabla_\theta f_{\text{CI-player}}(m; w) = -\frac{1}{(\hat{\rho}_m^2)^2} \nabla_\theta \hat{\rho}_m^2.$$

Unlike the gap and *Kendall's* τ objectives, this objective has a nonzero explicit weight derivative. If match n compares players a_n and b_n, then

$$\left(\frac{\partial f_{\text{CI-player}}(m;w)}{\partial w_n}\right)_{\text{explicit}} = -\frac{v_n\,\mathbf{1}\{m \in \{a_n, b_n\}\}}{(\hat{\rho}_m^2)^2},$$

where

$$v_n = \sigma(\hat{\theta}_{a_n} - \hat{\theta}_{b_n})\big(1 - \sigma(\hat{\theta}_{a_n} - \hat{\theta}_{b_n})\big).$$

Therefore, the drop effect is

$$\Delta f_{\text{drop},n}^{(f_{\text{CI-player}})} = \nabla_\theta f_{\text{CI-player}}(m;w)^\top \Delta\theta_n^{\text{drop}} + \frac{v_n\,\mathbf{1}\{m \in \{a_n, b_n\}\}}{(\hat{\rho}_m^2)^2}.$$

Equivalently,

$$\mathcal{I}_n^{(f_{\text{CI-player}})} = -\nabla_\theta f_{\text{CI-player}}(m;w)^\top \Delta\theta_n^{\text{drop}} - \frac{v_n\,\mathbf{1}\{m \in \{a_n, b_n\}\}}{(\hat{\rho}_m^2)^2}.$$

Trace-uncertainty objective. The global uncertainty proxy is

$$f_{\text{CI-trace}}(w) = \sum_{i=1}^{M} \frac{1}{\hat{\rho}_i^2(w)}.$$

Hence

$$\nabla_\theta f_{\text{CI-trace}} = \sum_{i=1}^{M} -\frac{1}{(\hat{\rho}_i^2)^2} \nabla_\theta \hat{\rho}_i^2.$$

For match n between players a_n and b_n, the explicit weight derivative is

$$\left(\frac{\partial f_{\text{CI-trace}}}{\partial w_n}\right)_{\text{explicit}} = -v_n \left(\frac{1}{(\hat{\rho}_{a_n}^2)^2} + \frac{1}{(\hat{\rho}_{b_n}^2)^2}\right).$$

Therefore, the drop effect is

$$\Delta f_{\text{drop},n}^{(f_{\text{CI-trace}})} = \nabla_\theta f_{\text{CI-trace}}(\hat{\theta}, w)^\top \Delta\theta_n^{\text{drop}} + v_n \left(\frac{1}{(\hat{\rho}_{a_n}^2)^2} + \frac{1}{(\hat{\rho}_{b_n}^2)^2}\right).$$

Equivalently,

$$\mathcal{I}_n^{(f_{\text{CI-trace}})} = -\nabla_\theta f_{\text{CI-trace}}(\hat{\theta}, w)^\top \Delta\theta_n^{\text{drop}} - v_n \left(\frac{1}{(\hat{\rho}_{a_n}^2)^2} + \frac{1}{(\hat{\rho}_{b_n}^2)^2}\right).$$

***Kendall's* τ surrogate.**

$$f_{\tau,T}(\theta;\pi) = \frac{2}{M(M-1)} \sum_{a<b} s_{ab} \tanh\left(\frac{\theta_a - \theta_b}{T}\right),$$

where $s_{ab} \in \{-1, +1\}$ is induced by the reference ranking. Its gradient is

$$[\nabla_\theta f_{\tau,T}]_k = \frac{2}{M(M-1)} \sum_{a<b} s_{ab} \frac{1 - \tanh^2\left(\frac{\theta_a - \theta_b}{T}\right)}{T} (\mathbf{1}_{k=a} - \mathbf{1}_{k=b}).$$

There is no explicit weight term, so the drop effect is

$$\Delta f_{\text{drop},n}^{(f_{\tau,T})} = \nabla_\theta f_{\tau,T}(\hat{\theta})^\top \Delta\theta_n^{\text{drop}}.$$

Equivalently,

$$\mathcal{I}_n^{(f_{\tau,T})} = -\nabla_\theta f_{\tau,T}(\hat{\theta})^\top \Delta\theta_n^{\text{drop}}.$$

B.6 Algorithms

Algorithm 1: Top-k Robustness Action Search

Input: Fitted BT model $\hat{M}$ on dataset D; top-k cross-boundary candidate pairs $\mathcal{P}$; fixed action variant $a \in \{\texttt{drop}, \texttt{flip}, \texttt{add_pairs}, \texttt{add_outcomes}, \texttt{add_weighted}\}$; maximum action count A
Output: Smallest action count that makes at least one candidate top-k boundary gap cross zero, together with the corresponding pair and selected actions

1. Fit the Bradley–Terry model on D and obtain $\hat{\theta}$.
2. Construct the ordered set $\mathcal{P}$ of top-k cross-boundary candidate pairs.
3. Initialize an empty cache of per-pair influence reports.
4. **For** $\alpha = 1, \ldots, A$:
   (a) **For each** pair $(i, j) \in \mathcal{P}$:
      i. Define the gap objective
      $$f_{ij}(\hat{\theta}) = \hat{\theta}_i - \hat{\theta}_j.$$
      ii. If not already cached, compute the one-step influence report for (i, j) under action variant a over its corresponding candidate pool.
      iii. Let $g_{ij} = f_{ij}(\hat{\theta})$.
      iv. Select the top α candidate actions from the cached report: if $g_{ij} > 0$, choose the most negative influences; otherwise choose the most positive influences. Selection is by logical action/match, with grouped forward/reverse copies treated as one action when applicable.
      v. Apply the selected α actions to the original fitted dataset D, refit the BT model and compute the updated gap $g_{ij}^{(\alpha)}$ or estimate it using influence scores.
      vi. **If**
      $$g_{ij} > 0 \text{ and } g_{ij}^{(\alpha)} \leq 0, \qquad \text{or} \qquad g_{ij} < 0 \text{ and } g_{ij}^{(\alpha)} \geq 0,$$
      return α, the pair (i, j), and the selected actions.
5. **Return** failure if no pair succeeds within A actions.

Algorithm 2: Strict CI-Aware Top-k Manipulation

Input: Fitted BT model $\hat{M}$ on dataset D; target boundary rank k; fixed insider–outsider pair (i, j) with i = rank-k model and j = rank-$(k+1)$ model; CI method; CI level α_{CI}; fixed action variant $a \in \{\texttt{drop}, \texttt{flip}, \texttt{add_pairs}, \texttt{add_outcomes}, \texttt{add_weighted}\}$; maximum action budget A
Output: Minimum number of actions certifying that outsider j lies above insider i under strict CI separation

1. Fit the Bradley–Terry model on D and compute the point estimate $\hat{\theta}$ and confidence intervals
$$[L_m, U_m] = \left[\hat{\theta}_m - z_{\alpha_{\text{CI}}}\text{SE}_m,\ \hat{\theta}_m + z_{\alpha_{\text{CI}}}\text{SE}_m\right], \qquad z_{\alpha_{\text{CI}}} = \Phi^{-1}(1 - \alpha_{\text{CI}}/2).$$
2. Fix the target pair (i, j) before the action search begins, where i is the current insider and j is the current outsider.
3. Define the strict CI objective
$$g_{ij}(\hat{\theta}) = \left(\hat{\theta}_i + z_{\alpha_{\text{CI}}}\text{SE}_i\right) - \left(\hat{\theta}_j - z_{\alpha_{\text{CI}}}\text{SE}_j\right).$$
4. Proceed only if the strict target is initially unmet, i.e.
$$g_{ij}(\hat{\theta}) \geq 0.$$
5. Compute the full one-step influence report for action variant a with respect to the strict objective g_{ij} over its corresponding candidate pool.
6. **For** $\ell = 1, \ldots, A$:
   (a) Select the top ℓ logical actions that most decrease g_{ij}. Candidates are ranked in ascending order of influence. When forward/reverse copies share a match identifier, they are grouped and treated as one logical action.
   First-order screening step. To reduce computation, we first use the influence approximation to screen budgets instead of refitting the BT model after every candidate action set.
      i. Form the first-order screened objective
      $$\widetilde{g}_{ij}^{(\ell)} = g_{ij}(\hat{\theta}) + \sum_{n \in S_{ij}^{(\ell)}} \mathcal{I}_n^{(g_{ij})}.$$

ii. If $\widetilde{g}_{ij}^{(\ell)} \geq 0$, continue to the next budget without refitting.

iii. If $\widetilde{g}_{ij}^{(\ell)} < 0$, apply the selected actions to the original fitted dataset and refit the BT model.

(b) Recompute the exact strict objective on the refit model:

$$g_{ij}^{\text{refit}} = \big(\hat{\theta}_j^{\text{refit}} - z_{\alpha_{\text{CI}}} \text{SE}_j^{\text{refit}}\big).$$

(c) Let

$$L_j^{\text{refit}} = \hat{\theta}_j^{\text{refit}} - z_{\alpha_{\text{CI}}} \text{SE}_j^{\text{refit}}, \qquad U_i^{\text{refit}} = \hat{\theta}_i^{\text{refit}} + z_{\alpha_{\text{CI}}} \text{SE}_i^{\text{refit}}.$$

(d) Declare success if

$$g_{ij}^{\text{refit}} < 0, \qquad \text{equivalently} \qquad L_j^{\text{refit}} > U_i^{\text{refit}}.$$

(e) If successful, return ℓ, the pair (i, j), the selected actions, and the refit ranking.

7. Return failure if no strict CI-certified reversal is found within budget A.

Algorithm 2-helper: Strict CI-Aware k-Selection

Input: Fitted BT model $\hat{M}$ on dataset D; CI method; CI level α_{CI}
Output: One or more valid target triples (k, i, j) for strict CI-aware manipulation, where i is the insider at rank k and j is the outsider at rank $k+1$

1. Fit the Bradley–Terry model on D and compute the CI ranking.
2. For each boundary rank $k = 1, \ldots, M-1$, let

$$i = \text{rank-}k \text{ model}, \qquad j = \text{rank-}(k+1) \text{ model}.$$

3. Define the strict CI objective

$$g_{ij}(\hat{\theta}) = \big(\hat{\theta}_i + z_{\alpha_{\text{CI}}} \text{SE}_i\big) - \big(\hat{\theta}_j - z_{\alpha_{\text{CI}}} \text{SE}_j\big), \qquad z_{\alpha_{\text{CI}}} = \Phi^{-1}(1 - \alpha_{\text{CI}}/2).$$

4. Keep only valid boundaries with

$$g_{ij}(\hat{\theta}) \geq 0,$$

i.e. boundaries for which strict CI-certified outsider-above-insider separation is not already satisfied.

5. Return the valid boundary (k, i, j) with the smallest strict objective value $g_{ij}(\hat{\theta})$.
6. Return failure if no valid boundary satisfies $g_{ij}(\hat{\theta}) \geq 0$.

Algorithm 3: Online Rigging Baseline (Omni-On)

Input: Initial Elo/BT ranking on dataset D; target model m^*; direction $d \in \{\texttt{promote}, \texttt{demote}\}$; exposed pair stream $\{(a_t, b_t)\}_{t=1}^{B}$; budget B; Elo constants $K, \text{BASE}, \text{SCALE}$
Output: Ordered decisions and target-rank trajectory

1. Compute the initial ratings $\hat{\theta}_m$ for all models and record the initial rank of m^*.
2. **For** $t = 1, \ldots, B$:
   (a) Observe the currently exposed pair (a_t, b_t).
   (b) Define the candidate decision set

$$\mathcal{A}_t = \{\texttt{a_wins}, \texttt{b_wins}, \texttt{tie}, \texttt{remove}\}.$$

   (c) Let $r_a = \hat{\theta}_{a_t}$, $r_b = \hat{\theta}_{b_t}$, and $r_* = \hat{\theta}_{m^*}$, and compute

$$e_a = \frac{1}{1 + \text{BASE}^{(r_b - r_a)/\text{SCALE}}}, \qquad e_b = \frac{1}{1 + \text{BASE}^{(r_a - r_b)/\text{SCALE}}}.$$

   (d) For each $\alpha \in \mathcal{A}_t$, form one-step hypothetical ratings:
   - `a_wins`: $r_a' = r_a + Ke_b, \quad r_b' = r_b - Ke_b$;
   - `b_wins`: $r_a' = r_a - Ke_a, \quad r_b' = r_b + Ke_a$;
   - `tie`: $r_a' = r_a - \frac{K}{2}(e_a - e_b), \quad r_b' = r_b + \frac{K}{2}(e_a - e_b)$;
   - `remove`: $r_a' = r_a, \quad r_b' = r_b$.

   (e) Score each action by the surrogate reward

$$r_t^{(\alpha)} = \frac{1}{1 + \text{BASE}^{(r_a' - r_*)/\text{SCALE}}} + \frac{1}{1 + \text{BASE}^{(r_b' - r_*)/\text{SCALE}}}.$$

   For demotion tasks, use the inverted reward $-r_t^{(\alpha)}$.

(f) Select the greedy rigging decision

$$\alpha_t^* = \arg\max_{\alpha \in \mathcal{A}_t} r_t^{(\alpha)}.$$

(g) Apply α_t^* to the current dataset, refit the global Elo/BT ranking, and append the new rank of m^* to the history.

3. **Return** the decision history and the full target-rank trajectory.

Algorithm 4: Online Influence-Guided Targeted Top-k Manipulation

Input: Fitted BT model on dataset D; target model m^*; rank cutoff k; direction $d \in \{\texttt{promote}, \texttt{demote}\}$; exposed pair stream $\{(a_t, b_t)\}_{t=1}^{B}$; budget B
Output: Selected decision sequence and final ranking

1. Fit the initial BT model on D and record the initial ranking of m^*.
2. **For** $t = 1, \ldots, B$:
   (a) Stop if m^* already satisfies the target condition:

$$\mathrm{rank}(m^*) \le k \quad \text{for promotion,} \qquad \mathrm{rank}(m^*) > k \quad \text{for demotion.}$$

   (b) Observe the currently exposed pair (a_t, b_t).
   (c) Define the current boundary objective

$$f_t(\hat{\theta}) = \begin{cases} \hat{\theta}_{m^*} - \hat{\theta}_{m_k}, & d = \texttt{promote}, \\ \hat{\theta}_{m_{k+1}} - \hat{\theta}_{m^*}, & d = \texttt{demote}, \end{cases}$$

   where m_k and m_{k+1} are recomputed from the current ranking after every accepted intervention.
   (d) Form the allowable decision set

$$\mathcal{A}_t = \{\texttt{remove}, \texttt{a_wins}, \texttt{b_wins}, \texttt{tie}\}.$$

   (e) For each $\alpha \in \mathcal{A}_t$, compute its first-order influence score on the current boundary objective:
   - `remove`: score 0;
   - `a_wins`: add the corresponding row for the $a_t \succ b_t$;
   - `b_wins`: add the corresponding row for the $b_t \succ a_t$;
   - `tie`: add both directed rows and sum their influence contributions.

   (f) Select the greedy influence decision

$$\alpha_t^* = \arg\max_{\alpha \in \mathcal{A}_t} \mathcal{I}^{(f_t)}(\alpha).$$

   (g) Apply α_t^* to the current state, refit the BT model if any row is added or estimate it using influence scores, and update the ranking of m^*.
3. **Return** the selected decisions, success indicator, rank trajectory, and final ranking.

Algorithm 5: CI Reduction via Targeted Match Addition

Input: Fitted dataset D, target model m^*, budget B, candidate mode v, CI method
Output: Added matches and CI-width trajectory

1. Fit the BT model on D.
2. Define the target uncertainty objective

$$f_{\text{CI-player}}(m^*; w) = \hat{\rho}_{m^*}^{-2}(w),$$

   where $\hat{\rho}_{m^*}^2(w)$ is the BT pair-weight-aggregated variance proxy defined in Section 3.
3. Generate add candidates according to v:
   - `all_pairs`: one unordered pair, with the currently higher-rated model as winner.
4. Compute influence scores for all add candidates once using the initial fitted model.
5. Sort candidates in ascending influence order, since negative influence reduces $f_{\text{CI-player}}$.
6. **For** $t = 1, \ldots, B$:
   (a) Select the next unused candidate from the fixed sorted list.
   (b) Add the match to D, refit the BT model or estimate it using influence scores, and recompute the target CI width using the chosen CI method.
7. **Return** the added matches, uncertainty trajectory, and CI-width trajectory.

B.7 Relation between the sandwich covariance and Gao-style local information

In the main text, we use the Gao-style local information approximation as a scalable uncertainty proxy for BT leaderboard perturbations. Here, we clarify how this approximation relates to the more general sandwich covariance estimator and why we avoid propagating perturbations through the full covariance matrix.

For a comparison $z_n = (i_n, j_n, y_n)$, define $x_n = e_{i_n} - e_{j_n}$, $p_n = \sigma(x_n^\top \hat{\theta})$, and $v_n = p_n(1 - p_n)$. The weighted BT loss has score and Hessian

$$g_n(\hat{\theta}) = (p_n - y_n)x_n, \qquad H_n(\hat{\theta}) = v_n x_n x_n^\top. \tag{13}$$

The sandwich covariance estimator is

$$\begin{aligned} \Sigma_{\text{sand}}(w) &= J(w)^{-1} S(w) J(w)^{-1}, \\ J(w) = \sum_{n=1}^{N} w_n H_n(\hat{\theta}) + \lambda I, &\qquad S(w) = \sum_{n=1}^{N} w_n g_n(\hat{\theta}) g_n(\hat{\theta})^\top. \end{aligned} \tag{14}$$

Under a correctly specified BT model,

$$\mathbb{E}\big[g_n(\theta^\star) g_n(\theta^\star)^\top \mid i_n, j_n\big] = v_n x_n x_n^\top = H_n(\theta^\star),$$

because $\mathrm{Var}(y_n \mid i_n, j_n) = p_n(1 - p_n) = v_n$. Thus, replacing the empirical score covariance $S(w)$ by its information counterpart gives $S(w) \approx J(w)$, so the sandwich estimator reduces to the information-based covariance approximation

$$\Sigma_{\text{sand}}(w) = J(w)^{-1} S(w) J(w)^{-1} \approx J(w)^{-1}. \tag{15}$$

Gao et al. [2023] provide a coordinate-wise interpretation of uncertainty in the BT model: the leading uncertainty of each skill estimate is controlled by the inverse of a local Fisher information term. In our weighted finite-sample setting, this local information for player i is

$$\hat{\rho}_i^2(w) = \sum_{j \neq i} w_{ij} p_{ij}(1 - p_{ij}), \tag{16}$$

Equivalently, if

$$J_{\text{info}}(w) = \sum_{i<j} w_{ij} p_{ij}(1 - p_{ij})(e_i - e_j)(e_i - e_j)^\top + \lambda I,$$

then $\hat{\rho}_i^2(w)$ is the local diagonal information associated with player i.

The full inverse $J_{\text{info}}(w)^{-1}$ captures coupling across all players, but computing and differentiating its diagonal or trace under many candidate perturbations is substantially more expensive. We therefore use the Gao-style local approximation

$$\mathrm{Var}(\hat{\theta}_i) \approx \hat{\rho}_i^{-2}(w),$$

which leads directly to the player-wise uncertainty objective $f_{\text{CI-player}}(i; w) = \hat{\rho}_i^{-2}(w)$ and the trace-style global uncertainty proxy $f_{\text{CI-trace}}(w) = \sum_i \hat{\rho}_i^{-2}(w)$. This approximation preserves the main statistical intuition: uncertainty is reduced by adding informative comparisons incident to poorly measured players, and increased by removing such comparisons.

B.8 Empirical agreement with sandwich CIs

We check whether CI-aware conclusions under the Gao proxy also hold under the full sandwich covariance. For each dataset and action, we fix the Gao-selected action sequence, refit BT after every action, and evaluate the strict CI objective of Algorithm 2 under both estimators. Let b_{Gao} and b_{sand} be the first budgets at which each estimator certifies the objective. We report the Spearman correlation between the two objective trajectories and decision agreement, the fraction of budgets with the same success/failure outcome (Table 9). Across 18 dataset–action settings, the median trajectory correlation is 1.00 and the median decision agreement is 99.0%. The proxy can be optimistic on sparse graphs, most notably ATP Top-10, where Gao certifies `Drop` at 13 actions but sandwich does not within the budget.

Table 9: Gao proxy vs. sandwich covariance for strict CI-aware Top-k changes. Entries are $b_{\mathrm{Gao}}/b_{\mathrm{sand}}$; "–" means not certified within the 5% budget. The last two columns give the minimum trajectory Spearman correlation and minimum decision agreement across actions.

Dataset	`Drop`	`Flip`	`Add`	Min. ρ	Min. agreement
ATP Top-10	13/–	7/11	–/–	0.908	63.6%
MT-Bench Human	62/77	33/43	–/–	0.999	76.7%
Arena LLM-J	32/32	17/17	50/51	1.000	98.0%
WebDev Arena	21/19	23/23	–/–	0.987	90.5%
Arena 55k	19/19	12/12	40/39	1.000	97.5%
Vision Arena	43/46	33/36	–/–	1.000	91.7%

B.9 Arena Active baseline for target-model CI reduction

Arena Active [Chiang et al., 2024] is implemented as a target-restricted variance-count active-sampling baseline. At each step, it recomputes scores under the current Bradley–Terry fit, considers only candidate pairs involving the target model m^*, and selects the opponent j with the largest estimated one-step reduction

$$s_{\mathrm{AA}}(m^*, j) = \sqrt{\frac{\widehat{\mathrm{Var}}(\hat{\theta}_{m^*} - \hat{\theta}_j)}{N_{m^*j}}} - \sqrt{\frac{\widehat{\mathrm{Var}}(\hat{\theta}_{m^*} - \hat{\theta}_j)}{N_{m^*j} + 1}}, \tag{17}$$

where N_{m^*j} is the current number of comparisons between m^* and j. Thus, Arena Active is target-specific through candidate filtering, but uses a pair-level variance-count score rather than directly optimizing $f_{\text{CI-player}}(m^*)$. In contrast, our method scores each candidate `Add` action by its predicted effect on the target uncertainty objective.

Arena Active always uses an `all_pairs` candidate space, while the influence policy can use `all_pairs`, `all_outcomes`, or `all_outcomes_weighted`. Thus, `all_pairs` is the fairest direct comparison, whereas the outcome-aware spaces evaluate a richer influence action space.

B.10 Threat models and action interpretation

The actions differ in the access they assume. We map each to the actor it models:

- **External injector.** Our online promotion/demotion experiment (Algorithm 4): the attacker observes a platform-exposed pair and may vote for either model, submit a tie, or abstain, but cannot choose the pair or modify historical data. The attacker's actions are restricted to new votes, although the influence-guided policy assumes white-box access to the fitted BT model and comparison statistics.
- **Platform operator or data curator.** `Add` with `all_pairs` for targeted data collection, `Drop` for filtering or correcting observations, CI-reduction sampling, and player removal for model deprecation.
- **Insider or compromised platform.** Adversarial `Drop` or `Flip`, which require modifying stored records.
- **Auditor.** `Drop` and `Flip` used counterfactually to identify the observations that most strongly support a leaderboard conclusion.

Thus, the small `Drop` and `Flip` budgets reported in Section 4.1.1 measure structural fragility under stronger or counterfactual interventions, not the capability of an ordinary external attacker.

C Reproducibility details

For reproducibility, we report the main hyperparameters and compute settings used in our experiments. Unless otherwise stated, all Bradley–Terry models are fit with `hessian_ridge=0.0`. In the grouped Newton refinement, we use a small numerical stabilization ridge of 10^{-8}. For the smooth Kendall's τ surrogate, we use temperature $T = 0.1$ in the player-level grouped-drop analysis and $T = 0.5$ in the robustness and action-curve analyses. All influence computations use the `1sn` approximation throughout.

All main experiments are run with 4 CPUs and 32 GB memory per task. We use the same hyperparameter settings across datasets unless explicitly noted above.

D Other results

D.1 Dataset-level robustness scores

To summarize robustness across datasets, we define normalized audit scores for the three leaderboard conclusions studied in the main text. Let B denote the maximum perturbation budget used for the corresponding audit and let $b^\star$ be the minimum number of influence-guided actions required to change the audited criterion after refitting the Bradley–Terry model. If no change occurs within the budget, we set $b^\star = B$. We define

$$R_{\text{Top-1}} = \frac{b^\star_{\text{Top-1}}}{B_{\text{Top-1}}}, \qquad R_{\text{CI}} = \frac{U(B_{\text{CI}})}{U(0)}, \qquad R_\tau = \tau(B_\tau),$$

where $R_{\text{Top-1}}$ measures the normalized budget needed to change the point-estimate Top-1 boundary, R_{CI} reports the remaining trace-uncertainty proxy under the fixed CI-trace audit budget, and R_τ reports the Kendall's τ value after the fixed global-ranking audit budget. Thus, lower $R_{\text{Top-1}}$ indicates that fewer actions are needed to change the local leaderboard conclusion, while lower R_{CI} and lower R_τ indicate stronger degradation under the corresponding fixed-budget audits.

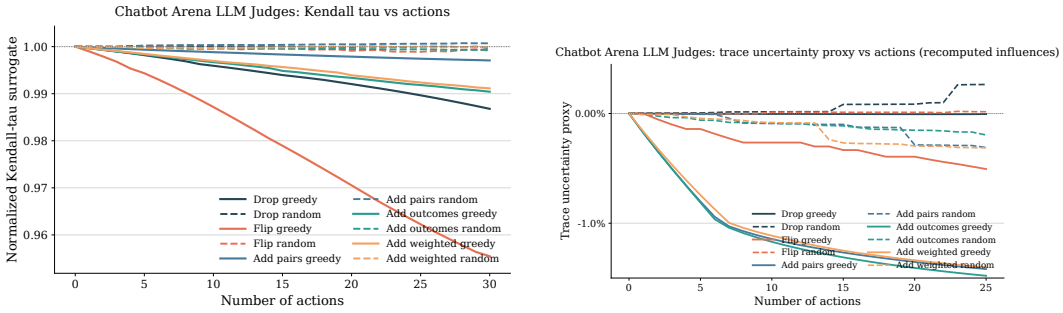
For a compact overall summary, we also report

$$R_{\text{all}} = \frac{1}{3}\left(R_{\text{Top-1}} + R_{\text{CI}} + R_\tau\right). \tag{18}$$

This aggregate is not intended to replace the three component scores. Instead, it provides a single coarse audit number while preserving the decomposition into local point-estimate fragility, CI-trace robustness, and global ranking stability.

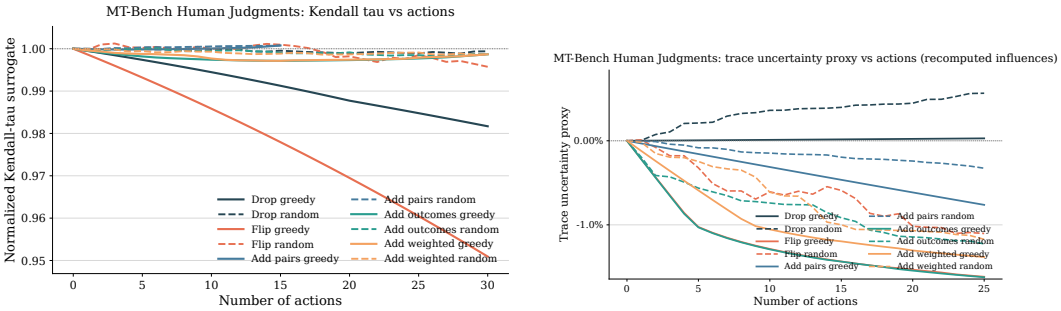
D.2 *Kendall's τ* and trace uncertainty curves

Figures 4–9 report the full perturbation-budget curves for other datasets beyond Arena 55k (shown in the main paper, Figure 2). The same qualitative patterns hold throughout: influence-guided `Flip` causes the steepest τ degradation, `Add` greedy variants dominate trace uncertainty reduction, and random selection is negligible in every case. Datasets with sparser comparison graphs (ATP) exhibit much larger absolute τ degradation—reaching near 0 in the ATP case—confirming that graph density is a key determinant of global robustness.



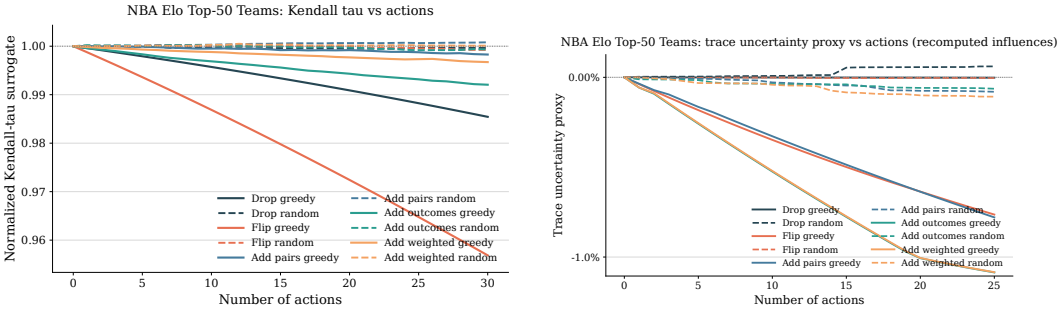
(a) *Kendall's τ*. (b) Trace uncertainty (%).

Figure 4: Perturbation-budget curves for **Chatbot Arena LLM Judges** (49,938 matches, 64 models). Influence-guided `Flip` degrades τ most steeply; `Add` greedy variants dominate uncertainty reduction.



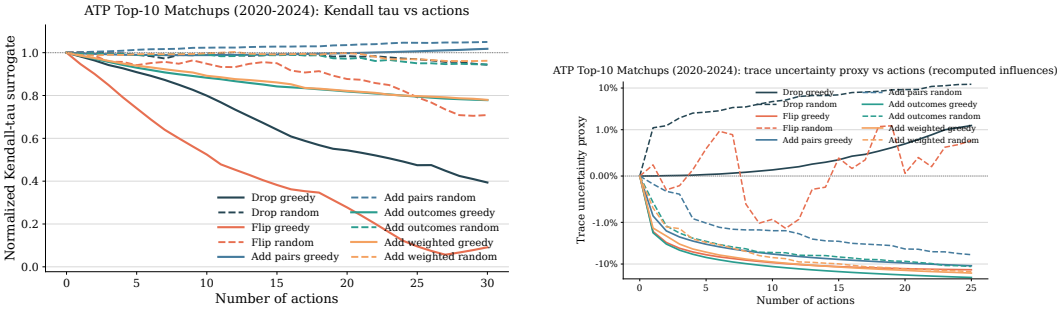
(a) *Kendall's* τ. (b) Trace uncertainty (%).

Figure 5: Perturbation-budget curves for **MT-Bench Human Judgments** (3,355 matches). τ degrades more slowly than on crowd-sourced datasets, consistent with Table 3 showing MT-Bench is the most robust dataset.



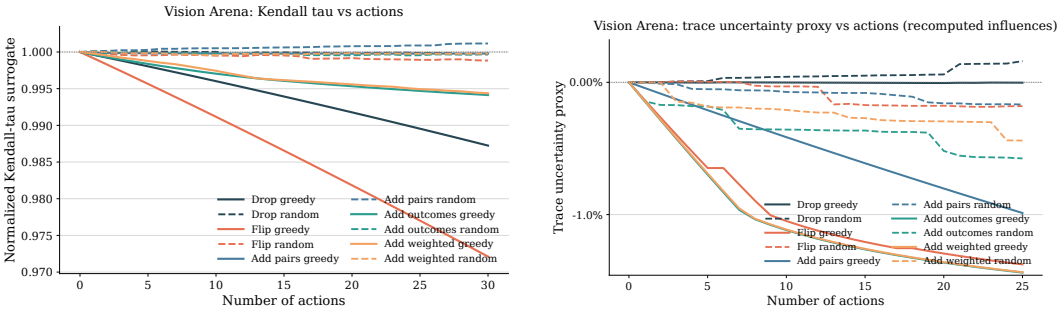
(a) *Kendall's* τ. (b) Trace uncertainty (%).

Figure 6: Perturbation-budget curves for **NBA Elo Top-50** (109,892 matches). The dense graph limits per-action impact on τ, but the reversal in which `Add` dominates uncertainty reduction persists.



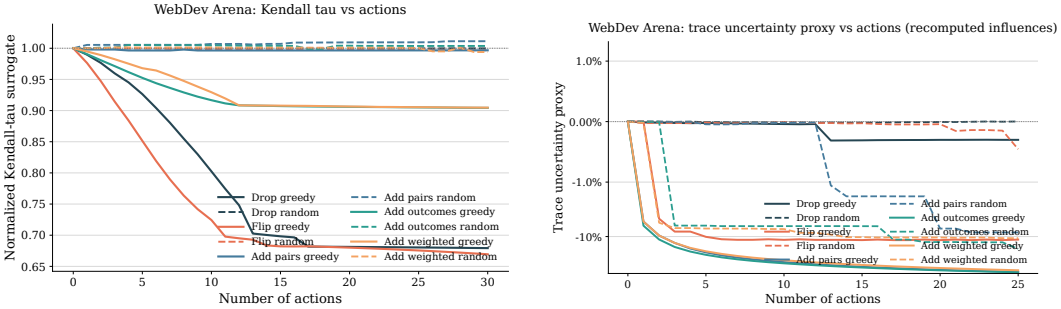
(a) *Kendall's* τ. (b) Trace uncertainty (%).

Figure 7: Perturbation-budget curves for **ATP Top-10** (278 matches). The extremely sparse graph makes this the least robust dataset: `Flip` drives τ to near 0 within 30 actions.



(a) *Kendall's* τ. (b) Trace uncertainty (%).

Figure 8: Perturbation-budget curves for **Vision Arena** (29,849 matches). Patterns mirror those of Arena 55k: `Flip` leads τ degradation and `Add` variants dominate uncertainty.



(a) *Kendall's* τ. (b) Trace uncertainty (%).

Figure 9: Perturbation-budget curves for **WebDev Arena** (10,501 matches). The sparse graph results in severe τ degradation (≈ 0.65 under `Flip`) while `Add` variants still dominate uncertainty.

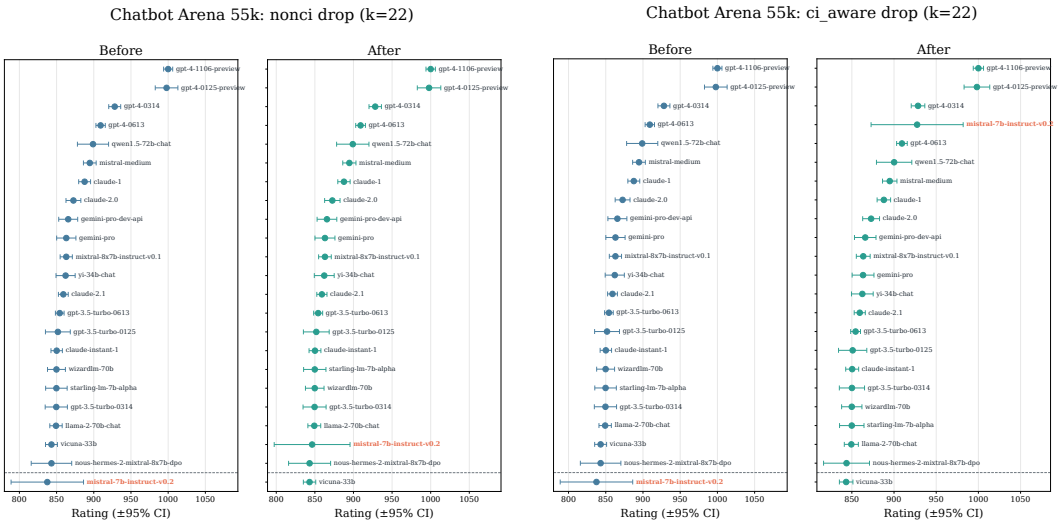
D.3 Top-k membership change: CI-aware vs. non-CI-aware

Figure 10 compares the effect of influence-guided drop perturbations on Top-k membership under the standard point-estimate objective and the stricter CI-aware objective. In the non-CI-aware setting, a membership change is counted as soon as the estimated skill ordering crosses the Top-k boundary, even if the affected models have overlapping confidence intervals; on Arena 55k with $k = 22$, this occurs after only 2 targeted `Drop` actions. By contrast, the CI-aware setting requires a stronger form of manipulation: the promoted model must not only cross the boundary in point estimate, but must do so with sufficient statistical separation from the displaced model, requiring 19 targeted `Drop` actions in the same setting. The comparison therefore separates two notions of robustness: instability of the displayed ranking versus instability that remains significant after accounting for uncertainty. As expected, CI-aware Top-k changes require stronger perturbations, but the fact that targeted drops can still alter membership under this stricter criterion shows that the leaderboard is not only locally sensitive but can also be vulnerable in a statistically meaningful sense.

D.4 Certificate-gated reporting and targeted collection

Publication rule. After each batch of new comparisons, we refit BT. If the internal Top-1 differs from the last published Top-1, we compute $b^{\star}_{\text{flip}}$, the minimum number of influence-ranked `Flip` actions required to reverse the proposed update (the Top-1 search of Section 4.1.1 with `Flip`). The update is published only if $b^{\star}_{\text{flip}} \geq \kappa$; otherwise it is withheld. Larger κ filters more fragile updates but increases withholding and latency.

Replay protocol. Arena 55k has no timestamps, so we replay it as a stream of 1,000-vote batches over 10 fixed random permutations; the replay therefore measures sampling-induced churn rather than temporal drift. We apply the rule whenever the internal winner changes, with primary threshold



(a) Non-CI-aware Top-k change.

(b) CI-aware Top-k change.

Figure 10: Top-k membership change under point-estimate and CI-aware objectives on Arena 55k. For $k = 22$, influence-guided `Drop` changes the point-estimate Top-k boundary after 2 actions, while the stricter CI-aware criterion requires 19 actions for the outsider to overtake the insider with non-overlapping confidence intervals. The two panels show that boundary membership is fragile, yet CI-aware success requires a more statistically robust change.

$\kappa = 5$. Since the audit cap is $B = 30$, this corresponds to $R_{\text{Top-1}} \geq 5/30 \approx 0.167$. We compare against *Always Publish*, which releases every internal Top-1 change immediately, and *Persistence-3*, which publishes a new Top-1 only after it appears in three consecutive snapshots.

Metrics. *Published updates* counts released Top-1 changes. A published change is *false* if the Top-1 returns to the previously published model within the next three snapshots; the *false-change rate* is the fraction of published changes that are false. *Mean latency* is the delay, in batches, between the first internal change and its publication. The *withholding rate* is the fraction of proposed updates not released, and the *miss rate* is the fraction of persistent changes that are never published. Results are in Table 4.

Targeted collection for withheld updates. For each withheld Top-1 change, each method may request up to 50 comparisons revealed from the held-out remainder of the permuted Arena 55k stream. After each comparison we refit BT and recompute $b^{\star}_{\text{flip}}$. Influence-guided collection ranks candidate `Add` comparisons by their influence on the contested Top-1 boundary; Arena Active and Random `Add` are as in Section 4.1.1. Table 10 reports the fraction of withheld updates certified within the budget and, among successes, the mean number of comparisons required. Influence-guided collection is more effective because it targets evidence around the specific unstable boundary rather than global under-sampling.

Table 10: Certifying withheld Top-1 updates with at most 50 additional comparisons.

Sampling method	Certified within 50	Mean comparisons to certification
Random `Add`	0%	never
Arena Active	17%	26.0
Influence-guided	**43%**	**15.6**

Threshold sensitivity and robustness of the replay. We sweep $\kappa \in \{1, 2, 3, 5, 10, 20, 30\}$; larger κ filters more fragile updates at the cost of more withholding and higher latency. At the primary operating point $\kappa = 5$, the pooled false-change rate over the 10 permutations is 0% across 8 published updates, with a mean latency of 2.5 batches (Table 4). Per permutation, the false-change rate is

0.000 ± 0.000 over the 7 permutations in which the gate published at least one update. A batch-size sensitivity check with 1,250-vote batches over 5 permutations gives the same qualitative result: the false-change rate drops from 0.438 under Always Publish to 0.000 at $\kappa = 5$, with a mean latency of 2.25 batches and a miss rate of 0.000.

E Ablation

E.1 Specification of matches

Influential matches reflect structure, not exposure. Figure 11 reports correlations between one-step match effects and simple match-level covariates across match-specification objectives. We consider four covariates: *match count*, the number of times the same unordered pair appears in the training data; *bridge variance*, a match-level structural diversity score computed from the variance of the opponents faced by the two endpoint players, intended to capture whether the match connects players with broad or heterogeneous comparison neighborhoods; *closeness*, the log-transformed absolute BT skill gap between the two matched players; and *surprise*, the discrepancy between the observed outcome and the probability predicted by the fitted BT model. Because the natural direction of degradation differs across objectives (decreasing *Kendall's* τ indicates worse global ranking stability, whereas increasing trace uncertainty indicates worse uncertainty), we interpret signed correlations relative to each objective rather than as a universal measure of influence magnitude. Under this objective-specific interpretation, *match count* is not a stable proxy for influence: frequently observed pairs are not consistently the ones whose perturbation most harms the leaderboard objective. By contrast, *bridge variance* and *closeness* often become strongly associated with large effects under flip objectives, suggesting that influence concentrates on structurally important and closely contested comparisons, where reversing an outcome can propagate broadly through the ranking. The *surprise* covariate is more heterogeneous, with its association changing across add, drop, and flip actions, especially for *Kendall's* τ, player uncertainty, and trace uncertainty. Overall, the figure suggests that influential matches are objective-dependent and are shaped more by structural uncertainty and near-boundary comparisons than by raw exposure alone.

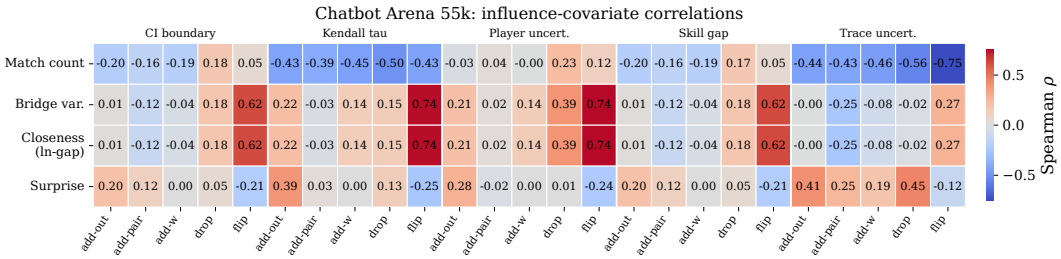


Figure 11: Signed Spearman correlations between one-step match influence scores and four match-level covariates in Chatbot Arena 55k, grouped by objective family and action type. Across most objectives, match count is weakly to strongly negative, while bridge variance and closeness are especially large for flip, highlighting the importance of uncertain, tightly contested comparisons in determining influential matches.

Influence concentrates on specified competitors. Under match-specific objectives, influence rankings concentrate sharply on the intended entities. Every objective that explicitly targets a player or player pair achieves a focus-hit rate of 1.0 at top-20 selected actions across all five action types: `Drop`, `Flip`, `Add`-pairs, `Add`-outcomes, and `Add`-weighted. We test this on Arena 55k: for the CI-boundary and skill-gap objectives defined on `gpt-4-1106-preview` versus `gpt-4-0125-preview`, as well as the player-uncertainty objective for `gpt-4-1106-preview`, all top-20 influential matches involve the designated focal player(s). Thus, when the objective specifies which competitors or matchups matter, influence does not spread diffusely across the dataset; it identifies edits tightly aligned with the target player or pair. In contrast, global objectives such as *Kendall's* τ and trace uncertainty have no designated focal player, so this focus-hit metric is not applicable to them.

E.2 Effect of k on *top-k membership*

To study how the choice of k affects the amount of data needed to change *top-k membership*, we vary k and measure the required perturbation budget. Figure 12 shows that the effort required to change the top-k ranking in Arena 55k is highly non-monotonic in k and depends strongly on the action type. The most striking feature is the sharp peak at $k = 3$, where adding pairs becomes dramatically more expensive than all other interventions, while adding weighted outcomes, adding outcomes, and dropping results remain substantially smaller, and flipping outcomes is consistently the least costly. Beyond $k = 5$, the required number of actions drops quickly for all methods and remains low and fairly stable through $k = 10, 20, 40$, indicating that deeper top-k boundaries are easier to perturb than the top few ranks. Overall, the figure suggests that the top-3 boundary is the most structurally resistant part of the ranking, especially when interventions are constrained to adding new comparisons rather than altering existing ones.

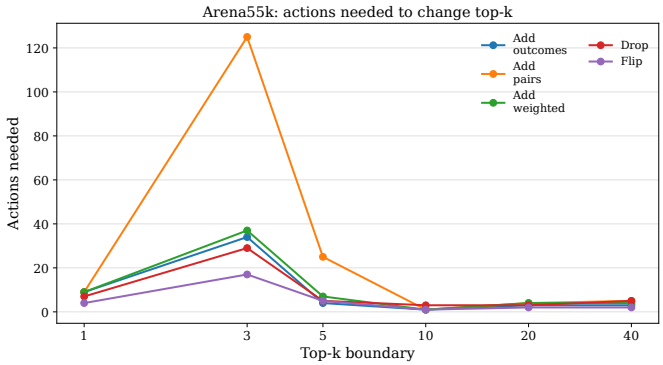


Figure 12: Top-k robustness is boundary-dependent. Minimum number of influence-guided actions required to change the Top-k set on Chatbot Arena 55k as k varies. The Top-3 boundary is the most structurally resistant, requiring roughly 120 actions, while several larger-k boundaries require only a few actions. Thus, robustness is not monotonic in k but depends on the local structure of the leaderboard near each cutoff.

E.3 Structural predictors of player-removal impact

To understand what drives player-removal impact, Figure 13 reports four association statistics between each structural player feature and $|\tau|$-influence, aggregated across datasets. We consider *degree*, the number of matches a player participates in; *bridge variance*, how unevenly a player connects otherwise weakly connected regions of the graph; *closeness*, how centrally located a player is in terms of shortest-path distances; and *surprise*, how unexpected a player's outcomes are relative to model predictions.

Degree is the strongest and most consistent predictor: it achieves the highest Pearson ($|r| = 0.41$) and Spearman ($|\rho| = 0.34$) correlations, and the largest quartile separation (Q4–Q1 mean $z = 0.65$, Cohen's $d = 0.86$). Bridge variance and closeness show weaker or less consistent associations, while surprise is only weakly associated.



Figure 13: Association between structural player features and $|\tau|$-influence, aggregated across datasets using Pearson and Spearman correlations, Q4–Q1 mean z-score, and Cohen's d. Degree is the strongest and most consistent predictor of player-removal impact.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main claims: a unified influence-based perturbation framework for Bradley–Terry leaderboards, evaluated under Drop, Add, Flip, and player removal. These claims are matched by the framework in Sections 2–3 and the empirical results in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in Section 6, including the locality of influence approximations, the use of scalable BT-based uncertainty proxies, and possible inaccuracies under large perturbation budgets or nonlinear ranking changes.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper states the Bradley–Terry modeling assumptions and influence-function setup in Sections 2–3, with additional derivations and approximation details in Appendix B. The results are primarily methodological derivations rather than standalone formal theorems.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the datasets, preprocessing, objectives, perturbation actions, baselines, budgets, and refitting protocol in Section 4 and Appendices A–B. Code to reproduce the main experimental results is available at `https://github.com/hosnahoseini/leaderboard-audit`.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, with instructions for data access, preprocessing, and reproducing the main experiments, is available at `https://github.com/hosnahoseini/leaderboard-audit`. All datasets used in the paper are public and are cited in Appendix A.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setting is described in Section 4 and Appendix A–B, including datasets, action spaces, perturbation budgets, objectives, baselines, and evaluation protocol. Hyperparameters and compute settings are reported in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Where stochasticity is present, the paper reports variability, for example the error bars in the targeted manipulation comparison. Many robustness experiments are deterministic given the fitted BT model and perturbation policy, so the reported quantities are exact post-refit outcomes rather than estimates over random seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C describes the compute environment: all main experiments were run with 4 CPUs and 32 GB of memory per task. The core experiments involve BT fitting and influence computations rather than training large models from scratch.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses public datasets and analyzes leaderboard robustness and manipulation risks in an auditing-oriented setting. We preserve anonymity in the submission and do not collect private or new human-subject data.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses positive impacts, such as auditing and improving leaderboard reliability, and negative impacts, such as the potential misuse of perturbation strategies for leaderboard manipulation, in Sections 1 and 6.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models, scraped datasets, or generative systems. Although the method has dual-use implications for leaderboard manipulation, these risks are discussed in the paper and the work is framed as an auditing and robustness tool.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites and credits the public datasets and prior assets used. Dataset details, sources, and licenses are provided in Appendix A.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code is released at `https://github.com/hosnahoseini/leaderboard-audit`, documented with a README and reproduction instructions. The paper does not introduce a new dataset or model checkpoint.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper analyzes existing public datasets that may contain previously collected human preferences, but it does not conduct new crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The work does not involve new human-subject experiments or new crowdsourced data collection, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method does not use LLMs as an original, important, or non-standard methodological component. LLMs were used for editing and drafting text, visualizing results, implementing standard methods, and understanding technical concepts; all content was verified by the authors.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.