Ranking Unraveled: Recipes for LLM Rankings in Head-to-Head AI Combat

Roland Daynauth[*], **Christopher Clarke**[*], **Krisztian Flautner, Lingjia Tang, Jason Mars**
University of Michigan
{daynauth, csclarke, manowar, lingjia, profmars}@umich.edu

Abstract

Evaluating large language models (LLMs) is a complex task. Pairwise ranking, where humans compare LLM outputs based on predefined criteria, has become a leading approach. By aggregating these comparisons through algorithms such as Elo, rankings across multiple LLMs can be derived. However, applying ranking algorithms in LLM evaluation presents several challenges. Traditional systems like Elo, designed initially for structured competitions such as chess, often produce inconsistent and unstable rankings due to the dynamic and context-dependent nature of LLM performance. Despite the increasing reliance on these methods, a systematic study of ranking algorithms for LLM evaluation remains lacking. This paper examines the effectiveness of various ranking systems for head-to-head LLM comparisons. We define key principles for robust ranking, conduct extensive evaluations of different ranking algorithms, and analyze their stability, accuracy, and sensitivity to real-world conditions. Our findings offer insights into the limitations of existing approaches and provide guidelines for selecting the most appropriate ranking method based on evaluation objectives and resource constraints.

1 Introduction

Rapid progress in large language models (LLMs) has highlighted the critical need for LLM evaluation methodologies, especially for complex generation tasks (Guo et al., 2023). Traditionally, NLP evaluation has relied on benchmarks such as GLUE (Wang et al., 2019), SuperGLUE (Wang et al., 2020), and more recent LLM-focused benchmarks like LM-Eval (Gao et al., 2023). However, a significant limitation of these classical benchmarks is their reliance on established ground truth, making them unsuitable for evaluating more nuanced, diverse, and complex cases where such ground truth is difficult to define or entirely unattainable (Chiang et al., 2024). This problem is particularly evident in complex tasks such as open-ended text generation, conversational dialogue, and creative writing, where human judgment is paramount. Previous work has shown that these benchmark evaluation results do not correlate well with human LLM evaluations (Zheng et al., 2024; Irugalbandara et al., 2023).

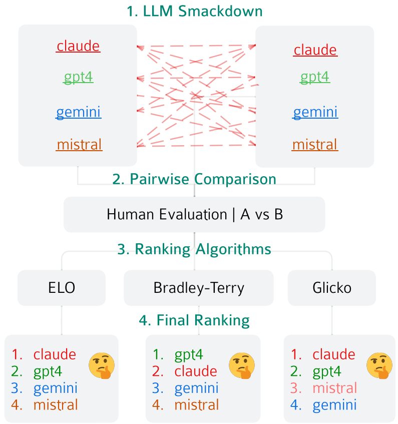


Figure 1: Different ranking algorithms can produce different rankings with the same human evaluation data, making it difficult to determine which algorithm is appropriate for various use cases.

This shortcoming has led the NLP community to adopt pairwise human evaluation methods (Dubois et al., 2024; Zheng et al., 2024; Chiang et al., 2024). A widely recognized example is *Chatbot Arena* (Chiang et al., 2024), which directly compares models through "head-to-head combat," where users vote between responses from two competing models. This method aligns with the Elo rating system (Elo, 1978), initially developed to classify chess players according to their historical competition results and relative strengths. Elo and other ranking methods have gained popularity as structured meth-

[*] Equal contribution.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26078–26091
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

ods incorporating human feedback and preferences to effectively rank models (Dettmers et al., 2024; Wu and Aji, 2023; Chiang et al., 2023).

However, as our experiments later demonstrated, *Elo and other established ranking algorithms can often produce inconsistent and sometimes unreliable results* (Figure 1), generating conflicting ranks. Thus, it is critical to understand how to select the ranking algorithm appropriately. There has been a lack of comprehensive studies across multiple ranking algorithms applied to LLM evaluations.

- First, a general framework for such a study has been lacking. It remains unclear which key metrics should be used to evaluate ranking algorithms and the factors that can affect the accuracy of the ranking algorithm.

- There has been no comprehensive study comparing multiple ranking algorithms or offering guidelines for selecting the most suitable in the context of LLM evaluations. Although prior work such as Boubdir et al. (2023) has explored some challenges associated with using Elo for LLM ranking, its focus is limited to Elo. In addition, our experiments show that recommendations from previous work are not always effective in practice.

Choosing the best model can significantly impact the user experience of the application that incorporates it. This paper explores the critical issues surrounding ranking LLMs in head-to-head human evaluations. Our contributions are as follows.

- We first systemically outline a set of fundamental properties that ranking systems should adhere to when evaluating LLMs.

- Second, we evaluate the performance of four widely used ranking algorithms (*Elo, Bradley-Terry, Glicko, and Markov Chain*) in two different evaluation scenarios (*Arena Style vs. Controlled Style*), analyzing their ability to maintain transitivity, ensure prediction accuracy, and remain stable across hyperparameter variations.

- Third, we conduct an in-depth analysis of these results, highlighting flaws in the current application of these popular algorithms. Despiteits wide use for LLM ranking, Elo remains unstable even after extensive tuning is applied.

- Lastly, we propose systematic guidelines for selecting the most appropriate approach based on the specific characteristics of the evaluation task and the available resources.

To our knowledge, this paper represents the first work that systematically addresses the complexities and intricacies of ranking LLMs across various methodologies. We release all code, data, and models to facilitate reproducibility and further research in this area.

2 Background

In this section, we formally define the task of pairwise ranking and briefly describe the suite of algorithms often used for aggregating ranking from pair-wise comparison results. The first three algorithms—Elo (Elo, 1978), Bradley-Terry (Bradley and Terry, 1952), and Glicko (Glickman, 1999)—are well-established and widely used in contexts such as sports competitions, while the Markov chain, based on a random walker model, is simpler in design and has demonstrated strong performance in previous studies (Callaghan et al., 2003, 2004).

2.1 Ranking

Let M be the set of models under evaluation where $M = \{1, 2, \cdots, m\}$. For any pair $i, j \in M$, the result S_{ij} of a head-to-head match is defined as $S_{ij} = 1$ when i beats j and $S_{ij} = 0$ when i loses and $S_{ij} = 0.5$ for ties.

If θ_i, θ_j represents the strength of i and j respectively, then the probability that $S_{ij} = 1$, written as p_{ij}, for θ_i, θ_j is assumed to follow a probability distribution F such that (David, 1988):

$$p_{ij} = F(\theta_i - \theta_j) \tag{1}$$

2.2 Elo Rating system

The Elo rating system was first adopted by the United States Chess Federation (USCF) in 1960 to measure players' strengths (Elo, 1978). For two players i and j with relative strengths θ_i and θ_j, the probability that i beats j, is calculated as follows:

$$p_{ij} = \frac{1}{1 + 10^{(\theta_j - \theta_i)/400}} \tag{2}$$

For any match between model i and j, θ_i (also called the Elo score for i) is updated via the formula:

$$\theta_i' = \theta_i + k \times (S_{ij} - p_{ij}) \tag{3}$$

where k is a hyperparameter know as the k-factor. The value of k determines the rate at which the Elo score of a player changes based on the outcome of the game.

2.3 Bradley-Terry Model

The Bradley-Terry model (Bradley and Terry, 1952) was introduced in 1952 and assumes that F in equation 1 follows a logistic distribution where

$$p_{ij} = \frac{e^{\theta_i}}{e^{\theta_i} + e^{\theta_j}} \tag{4}$$

The strengths θ_i for each model can be estimated by solving the likelihood function $\mathcal{L}(\theta)$ such that:

$$\mathcal{L}(\theta) = \prod_{i<j} p_{ij}^{y_{ij}} p_{ji}^{y_{ji}} \tag{5}$$

where y_{ij} is the number of times that model i beats j and p_{ij} is the probability of i beating j.

2.4 Glicko Rating system

The Glicko rating system (Glickman, 1999) was introduced in 1995 as an improvement to Elo. Glicko extends Elo by introducing a secondary parameter (σ) known as the *rating deviation*. This parameter measures the reliability of a player's rating score, acting as a confidence interval for the player's rank, decreasing based on the number of games played, regardless of wins or losses. This factor is important given that two players can potentially have the same score in a tournament despite a different number of previous matches. This situation is prevalent in LLM evaluations, as new models are frequently introduced and the number of games played between them vary significantly. In Appendix A.1 we explore the details of the algorithm.

2.5 Markov Chain

The Markov Chain Model is a nonparametric ranking algorithm adapted from Callaghan et al. (2003, 2004) and adapted into a Markov Chain Model by (Kvam and Sokol, 2006) (see Appendix A.2 for details). The Markov chain employs a series of random walkers traversing a graph, with each node representing a player/model in the tournament connected by an edge denoting a match between two nodes. Each walker chooses an initial node i, staying in the node or moving to an adjacent node j according to a single condition p. In a random match between i and j, the walker moves to the winning node with a probability of p for $p > 0.5$, a hyperparameter chosen in advance. Every time a walker moves or stays on a node, it represents a vote for that node, and the sum of all votes from every walker represents the rank of that player/model. In practice, the optimal value for p can vary depending on the tournament. In our experiments, we use $p = 0.8$ as the default value similar to Callaghan et al. (2003), which achieved stable results with p ranging from $p = 0.75$ to $p = 0.95$.

3 Desirable Properties of Ranking System

The goal of a ranking system is to provide a structured approach to compare and order items based on their performance or quality. In the context of LLMs, an effective ranking system allows researchers, practitioners, and end-users to identify the most capable models for their specific needs. . By simplifying decision-making and optimizing resource allocation, such a system highlights models that are likely to deliver superior performance in the real-world applications. However, implementing a robust ranking system requires addressing several key properties to ensure its validity and reliability. In this section, we identify and discuss three essential properties that a ranking system for LLMs should adhere to: 1) *Transitivity*, 2) *Prediction Accuracy*, and 3) *Sensitivity to Hyperparameters and Battle Conditions*.

3.1 Transitivity

Transitivity is a fundamental property for a ranking system to maintain consistency.

Definition 3.1. For any three models $i, j, k \in M$; their match-ups are *transitive* if $p_{ij} > 0.5, p_{jk} > 0.5$ and $p_{ik} > 0.5$. If i, j, k are transitive, we say that a ranking system *preserves transitivity* if it ranks these models in order i, j, k. Transitivity ensures that the ranking is coherent and interpretable.

Human evaluation data contains transitive and intransitive relationships, whereas ranking algorithms always produce perfectly transitive rankings. Transitive and intransitive relationships are mutually exclusive; however, intransitive relationships are inevitably lost, since ranking enforces transitivity. One of the key strengths of a ranking algorithm lies in its ability to preserve as much of the original transitive structure from human data as possible. Failure to do so can lead to paradoxical rankings, where the ordering of models becomes inconsistent, ultimately diminishing trust in the rating system.

3.2 Prediction Accuracy

Like transitivity, prediction accuracy is a critical metric that assesses how well the ranking system forecasts the results in head-to-head matches. It measures the probability that the ranking system correctly predicts the winner between any two LLMs. High prediction accuracy indicates that the ranking system is reliable and can be trusted to make informed decisions about the superiority of the model for a downstream task. For combat-style evaluation in LLM, where assessment is based on subjective human judgment, prediction accuracy acts as an indicator of the ranking system's ability to align with human preferences. This alignment is crucial for tasks like open-ended text generation or conversational agents, where the model's performance is best judged by human preference. Thus, the ranking system should reflect statistical superiority and resonate with qualitative human evaluations to truly enhance the utility and adoption of LLM.

3.3 Sensitivity to Hyperparameters

Another critical aspect of evaluating ranking algorithms is understanding their sensitivity to hyperparameters, which can significantly affect the consistency and stability of rankings (Boubdir et al., 2023). Ranking algorithms such as Elo have been shown to produce drastically different rankings based on the order in which matches are evaluated and the number of permutations (Boubdir et al., 2023). Since these hyperparameters can influence the behavior and outcomes of the ranking system, it is essential to choose the right settings to ensure reliable and consistent rankings. Sensitivity to hyperparameters means that the ranking system should respond to changes in these settings in a predictable and manageable way. An overly sensitive system might produce drastically different rankings with minor parameter changes, leading to instability and unreliability. In contrast, a system with too little sensitivity might fail to capture essential differences between models. In this work, we consider the impact of two different classes of hyperparameters/conditions: 1) *Algorithmic Hyperparameters*: parameters native to the ranking algorithm itself and 2) *Matchup Distribution*: Arena style vs. controlled style.

Algorithmic Hyperparameters such as k-factor in Elo, p-value in Markov Chain, and σ in Glicko can significantly affect the rankings produced by these algorithms. Similarly, the distribution of matchups between models can also influence the rankings. In practice, certain models may participate in many more matchups than others, creating an uneven distribution of data. This skews the rankings because models with fewer matches do not have sufficient data to be ranked reliably, while those with more matchups dominate the rankings. The ranking system chosen should be robust to these variations and produce consistent and reliable rankings across different hyperparameters and battle conditions.

Algorithm	Arena (%)	SLAM (%)
Elo	68.24	52.5
Markov	51.38	51.67
Glicko	56.54	53.33
Bradley-Terry	**77.29**	**56.67**

Table 1: Ranking Algorithm Performance on preserving transitivity on Chatbot Arena & SLAM Dataset.

4 Evaluating Ranking Algorithms

Building on the properties discussed in section 3, this section outlines a series of experiments designed to assess the quality of traditional ranking systems and their use in LLM rankings. We investigate how each algorithm performs under different conditions and evaluate their effectiveness in producing stable and consistent rankings.

4.1 Datasets

To evaluate these ranking systems, we use two diverse human evaluation datasets representative of two evaluation styles:

- **Chatbot Arena** (Chiang et al., 2024) is a dynamic platform where users evaluate two randomly selected model outputs head-to-head. As newer models are released, they are added to the arena to battle the older models. This data set consists of 57 models and a total of 244,978 match-ups. The distribution of matches varies wildly, with some models having up to 30416 matches or as few as 954. As such, we refer to this dataset as an *Arena-Style* dataset.

- **SLAM** (Irugalbandara et al., 2023) consists of pairwise match-ups between models evaluated using the SLAM tool on domain-specific task questions. Unlike Chatbot Arena, the

SLAM platform is tightly controlled with a fixed number of models and a similar number of match-ups between models, thus making it a controlled style dataset. The SLAM dataset consists of 11 models with a total of 2858 match-ups, each model having at most 529 match-ups and at least 501. As such, we refer to this dataset as a *Controlled Style* dataset.

These two diverse datasets enable us to assess ranking algorithms under varying conditions, offering a comprehensive view of their behavior. To evaluate dataset consistency, we apply Kendall's test (Kendall and Smith, 1940) (see A.6 for details), which confirms that any inconsistencies in the datasets are statistically insignificant.

4.2 Preserving Transitivity

Using the Chatbot Arena and SLAM datasets, we evaluated the ability of each ranking algorithm to preserve transitivity (see 3.1). We evaluated the transitive performance of each ranking algorithm by first calculating the number of triples within the data set. Three models i, j, k are part of a triple if i wins the majority of comparisons with j, and j wins the majority of comparisons with k, and transitivity is considered preserved if this order is maintained in the final ranking. Table 1 shows the performance of each ranking algorithm under the arena and controlled style evaluation scenarios. We observe that the Bradley-Terry model is the most performant, preserving transitivity in 77.29% and 56.67% of the triples, respectively, compared to its closest competitor, Elo, at 68.24% and 52.5%.

Bradley-Terry uses Maximum Likelihood Estimation to calculate each player's strength based on the outcome of all their matches concurrently. This is in contrast to Elo, which updates each player's rating after each match sequentially and is therefore sensitive to the other in which matches are evaluated. This gives Bradley-Terry the advantage in preserving transitivity in larger datasets such as Chatbot Arena.

Bradley-Terry consistently outperforms other algorithms in preserving transitivity, achieving the highest rankings across both datasets.

4.3 Prediction Accuracy

Next, we evaluate the ability of each ranking algorithm to predict the outcome of unseen LLM matches. We first partition each dataset into a 75/25 train/test split, running each ranking algorithm on the training set to determine the final ranking and strength of each model π. We then use each ranking algorithm to predict the outcome of the matches in the test set and calculate the F1 score as our performance metric.

Algorithm	Arena (F1)	SLAM (F1)
Elo	**0.90**	0.87
Markov	0.77	**0.88**
Glicko	0.88	**0.88**
Bradley-Terry	0.82	0.87

Table 2: Ranking algorithm performance in prediction accuracy (F1 score) on Chatbot Arena & SLAM Dataset.

Performance on the Arena Dataset Due to its uneven distribution of model match-ups, the Arena data set presents an interesting challenge for ranking algorithms to predict future outcomes. When matches between models are unevenly distributed, some algorithms struggle with prediction accuracy. Our results show the Elo model to be the most accurate, achieving an F1 score of 0.90 compared to the F1 score of 0.88, 0.82, and 0.77 for the Glicko, Bradley-Terry and Markov Chain models, respectively (Table 2).

The impact of the match-up distribution on the prediction accuracy is shown in the performance of the Markov chain which is affected by the sparsity of certain model match-ups, leading to lower prediction accuracy, as shown by the poor performance on models such as *deepseek-llm-67b-chat* and *dolphin-2.2.1-mistral-7b* in figure 4 of the appendix.

Another example of the impact of matchup distribution is shown in the performance of the Bradley-Terry model in the appendix figure 4. We observe that the Bradley-Terry system suffers when dealing with "powerful models," where there is a strong imbalance between a model's wins and losses. For example, in the arena dataset, *gpt-4-turbo* had a win/loss ratio of 12288/3979. This skew in model wins/loss results in overestimation of the model's strength as shown by the low F1 score of 0.82 achieved by *gpt-4-turbo* using Bradley-Terry compared to an F1 score of 0.95, 0.96 and 0.92 from Elo, Glicko, and Markov Chain, respectively. This is known as the 'rare events' problem in logistic regression (King and Zeng, 2001). One method of countering this is to use a weighted logistic regression(Chiang et al., 2024). However, we observe that this has a negligible impact on the final result.

Method	Elo	Markov	Glicko	Bradley-Terry	Win-Rate
Elo Rating	1.00	0.89	0.93	0.95	0.91
Markov	-	1.00	0.99	0.97	0.98
Glicko	-	-	1.00	0.99	0.99
Bradley-Terry	-	-	-	1.00	0.98
Win-Rate	-	-	-	-	1.00

Table 3: Spearman's correlation between ranking algorithms on the SLAM dataset

Method	Elo	Markov	Glicko	Bradley-Terry	Win-Rate
Elo Rating	1.00	0.74	0.86	0.94	0.93
Markov	-	1.00	0.76	0.81	0.78
Glicko	-	-	1.00	0.81	0.89
Bradley-Terry	-	-	-	1.00	0.91
Win-Rate	-	-	-	-	1.00

Table 4: Spearman's correlation between ranking algorithms on the Arena dataset

Furthermore, the bootstrap confidence interval used by Chatbot Arena shows a narrow range of [-2.2, 2.3], suggesting a high confidence in the model ranking. However, this interval does not guarantee how well the model will perform against unseen opponents.

Prediction accuracy on the SLAM dataset The distribution of matches between models in the SLAM dataset is tightly controlled so that every reviewer evaluates an equal number of models, resulting in a near uniform distribution of matches. We observe near-identical performance for all models. This result is due to the balanced distribution of match-ups in the dataset, where if a model has the most wins in the competition, its performance is directly reflected in its win rate. We further validate this occurrence by ranking all the models in SLAM by win rate and calculating the correlation between each pair of ranking algorithms, as shown in Tables 3 and 4. We observe a high correlation between all methods and the win rate.

All algorithms perform similarly on the SLAM dataset, but on Arena, Elo leads in accuracy, followed by Glicko. Bradley-Terry, however, is negatively affected by the uneven distribution of the Arena dataset.

4.4 Hyperparameter Sensitivity

To measure the sensitivity of the hyperparameters, we calculate the F1 score of each ranking system as described in Section 4.3, varying each hyperparameter over a range of 100 values. Figure 2 showcases the range of results produced by Elo, Markov and Glicko across the two evaluation scenarios.

Elo k-factor The k factor in Elo dictates the extent to which the score of a player is influenced by the outcome of an individual match. For example, with large k values, highly ranked players can incur a significant penalty when losing to a player with a lower ranking, while defeating such a player only results in a slight increase in score. The evaluation of the performance of the model (depicted in Figure 2) at varying levels of hyperparameter k shows marked sensitivity, which emphasizes the need for meticulous tuning. Elo shows the largest variation across both datasets, and is less stable, particularly for the smaller SLAM dataset. Through our tuning of this parameter, we determined that the ideal k factor for Elo is less than the standard 32 commonly used in previous research (Dettmers et al., 2024; Wu and Aji, 2023; Chiang et al., 2023). This reduced convergence rate can be attributed to the complexities of human preferences as opposed to the typical assessment of real-world games, which rely on the differing abilities of opposing teams.

Markov Chain: p Score The p score in the Markov Chain determines the extent to which the overall ranking is influenced by each model's win rate, with values closer to 1 indicating greater influence compared to those closer to 0.5. We observe that Markov shows stable performance on the controlled SLAM datasets, performing as well as Glicko (Figure 2), however, the algorithm struggles on the large ARENA dataset due to its over reliance on win-rates for calculating ranks. When optimizing the value of this parameter, we find that the optimal p value for the Markov chain on the SLAM dataset is larger than the Arena dataset, indicating that the model's strength is greatly influenced by its overall win rate.

Glicko: Rating deviation For Glicko, the ranking deviation (σ) is a measure of consistency or variability in a player's performance over time and determines how flexible a player's rating can be. If σ is initially set too high, the system will interpret the early game results as more variable or uncertain. This leads to larger rating adjustments, making the player's rating fluctuate more. In contrast, setting σ too low initially restricts the system from adjusting the rating freely. As a result, players, especially newcomers, may take much longer to reach a rating that accurately reflects their true skill level. This is particularly impactful in fast-changing or skill-evolving games, where a low initial variability setting could lead to underestimation of skill. We observe that Glicko's prediction performance is consistent across both datasets, showing that its

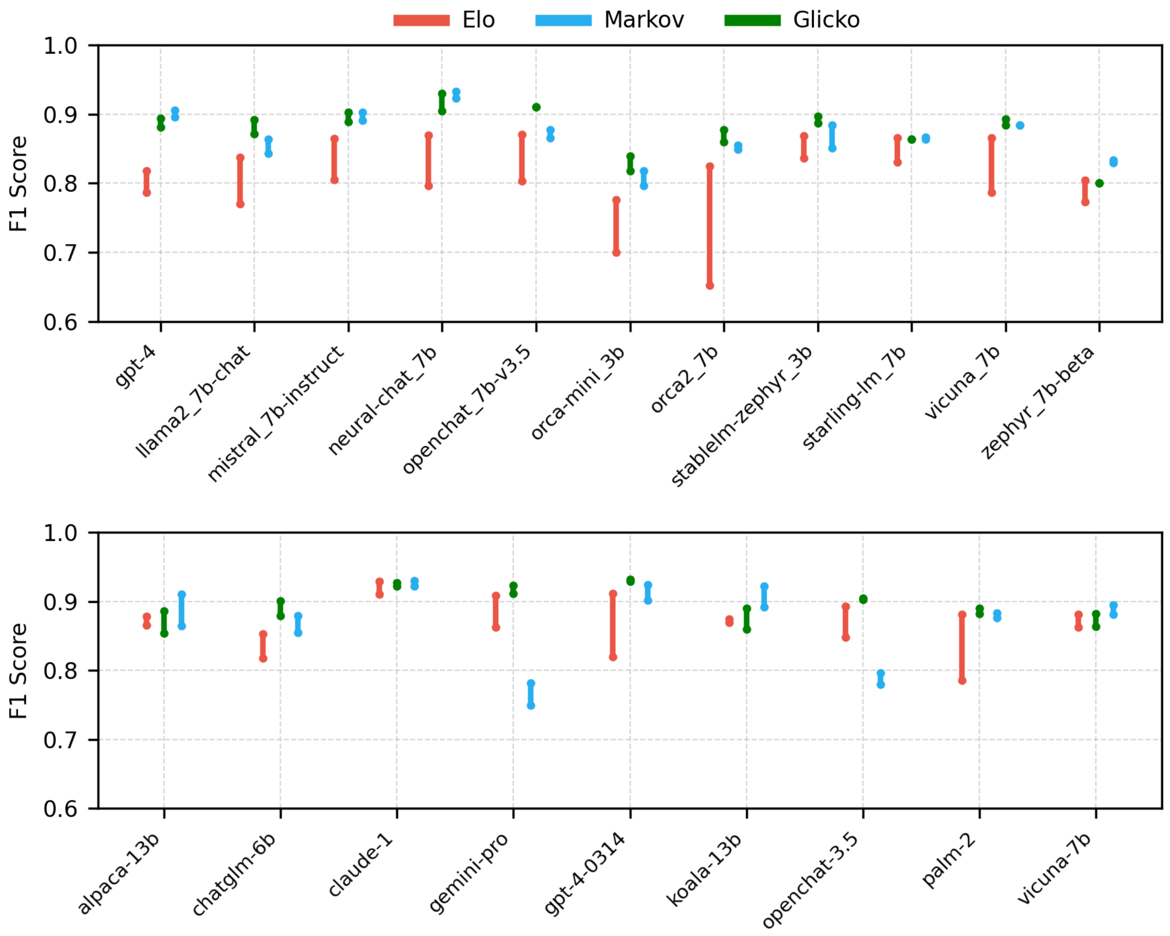


Figure 2: Distribution of F1 scores for the SLAM (top) and ARENA (bottom) datasets, showing the performance of Elo, Markov, and Glicko algorithms across a subset of models with 100 different hyperparameter settings. The results highlight the volatility of Elo, demonstrating its high sensitivity to hyperparameter changes compared to the more stable performance of Markov and Glicko.

performance is less impacted by changes in its hyperparameters compared to Elo and Markov Chain. We theorize that this is due to the large number of games played for each model in both scenarios, allowing the system to adjust the rating deviation dynamically.

Elo shows high sensitivity to hyperparameter whereas Glicko demonstrates greater stability and accuracy which suggests that Glicko is a more reliable alternative for LLM evaluation.

5 Recommended Best Practices

In sections 3 and 4, we outlined the desired properties of ranking systems and presented experiments assessing the performance of these systems in achieving these criteria. Based on these observations, this section puts these results into perspective and provides practical recommendations for selecting the most appropriate ranking algorithm. We outline which algorithms perform best under specific circumstances, offering guidance on their optimal use depending on the structure of the dataset, the matchup distribution, and the model evaluation goals. These recommendations, summarized in Table 5, aim to improve the reliability and applicability of classification systems in real-world LLM evaluations.

5.1 Challenges with Elo

An important result in this paper is that Elo, even with $>$ 1,000 permutations, struggles to achieve stable rankings A.5. This indicates that Elo's dependency on permutations for ranking stabilization is inadequate, challenging previous studies (Boubdir et al., 2023). Combined with the high sensitivity to hyperparameters and the average performance in preserving transitivity. **We do not**

Criteria	**Elo**	**Markov**	**Glicko**	**Bradley-Terry**
Transitivity	Moderate performance on smaller datasets, better than average performance on larger datasets.	Preserves transitivity but sensitive to sparse and uneven matchups.	Good transitivity preservation in both types of datasets.	Generally good at maintaining transitivity, in smaller controlled and larger uncontrolled datasets.
Hyperparameter Sensitivity	Highly sensitive to K-factor and permutation count; performance fluctuates with changes.	Sensitive to the p-value, with rankings varying significantly based on its setting.	Incorporates rating deviation, making it less sensitive than Elo, but hyperparameter tuning is still necessary.	Does not use any hyperparameters.
Prediction Accuracy	Moderate prediction accuracy but inconsistent with small sample sizes or highly uneven matches.	Moderate prediction accuracy but highly dependent on matchup distribution.	High prediction accuracy due to consideration of the uncertainty in the ratings (rating deviation).	Higher prediction accuracy in controlled datasets but suffers in uneven matchups.
Recommended Use Cases	Not recommended for small, unevenly distributed datasets; may perform better in larger datasets with consistent matchups.	Applicable where dataset size is small and even, but sensitive to distribution; effective if properly tuned in moderately structured environments.	Best suited for large, unevenly distributed datasets. Able to manage uncertainty and prevent low matchup models from being ranked too high.	Ideal for small, unevenly distributed datasets. Also effective for large, evenly distributed datasets where interpretability and scalability are valued.

Table 5: Consideration for Ranking Algorithm Selection based on Desired Ranking Properties

recommend ELO as an algorithm to rank LLM performance.

Best Practice 1. Elo's high sensitivity to changes in its K factor and instability (even with large permutations) make it unsuitable for efficient ranking of LLM performance in pairwise evaluation (Sections 4.4 and A.5).

5.2 Choose Bradley-Terry for Small Controlled Datasets

In section 4.3, we demonstrate that ranking algorithms exhibit varied performance based on battle conditions. When matches are evenly distributed, the ranking algorithms produce similar results. However, a heavily skewed distribution can negatively impact some methods such as Elo and Bradley-Terry. In this scenario, where matchups are balanced and the number of matches is small, objective findings show that Bradley-Terry is best due to its ability to preserve transitivity while also maintaining comparable prediction accuracy.

Best Practice 2. Bradley-Terry's simplicity and lack of hyperparameters make it ideal for controlled evaluation settings, although its reliance on sufficient matchup data limits its effectiveness in unevenly distributed scenarios.

5.3 Choose Glicko for large uneven datasets

For larger datasets, especially those with uneven matchup distributions, the Bradley-Terry model encounters difficulties due to the rare event problem. This problem occurs when new models with high win rates, participate in fewer matchups than well-established models. As a result, these new models can be ranked disproportionately higher based on limited data, leading to biased rankings. In this scenario, the Glicko system provides a superior solution. Glicko utilizes a rating deviation parameter to account for uncertainty in a model's ranking due to limited matchups. This allows the system to adjust the rankings more conservatively for models with fewer comparisons, preventing them from being ranked too high based only on a few matches. As data sets become larger and more uneven, Glicko's ability to incorporate rating deviation ensures that the rankings accurately reflect the true performance throughout the dataset, even when new models appear with limited match histories. This flexibility makes Glicko particularly effective for large-scale LLM assessments where uneven matchup distributions are prevalent.

Best Practice 3. Glicko is effective for large data sets with uneven matches due to its rating deviation parameter, which manages uncertainty and prevents models with few matches from being overrated.

5.4 Bradley-Terry for other Dataset Types

For small, uneven datasets, Bradley-Terry's simplicity and ability to model pairwise comparisons make it a robust choice, even when the data distribution is skewed. It does not require parameter estimation, which allows it to handle small datasets efficiently without the need for complex adjustments or hyperparameter tuning, making it ideal for situations where uneven distributions might otherwise require additional preprocessing.

In the case of large, evenly distributed datasets, Bradley-Terry is effective because of its scalability and ease of interpretation. For large data sets where the distribution of results is balanced, Bradley-Terry's ability to produce clear, interpretable rankings without adding excessive computational overhead is a significant advantage. Although more complex models such as Glicko can offer finer granularity in large, uneven data cases, Bradley-Terry's efficiency and interpretability make it a viable alternative when computational simplicity and transparency are prioritized.

Best Practice 4. Bradley-Terry's scalability and interpretability make it a viable choice for large, evenly distributed datasets, offering clear rankings with minimal computational overhead.

6 Related Work

Pairwise ranking has recently emerged as a popular method for evaluating LLMs, as it allows for direct comparisons between models and provides a more nuanced assessment of their performance than traditional metrics (Chiang et al., 2024). This evaluation involves presenting human judges with pairs of model outputs and asking them to select the better one. The outcomes of these pairwise comparisons are then aggregated into final rankings using ranking algorithms such as Elo (Elo, 1978). Despite their widespread use, there has been limited research on the applicability of these ranking algorithms to LLM evaluation. Works such as (Boubdir et al., 2023) have critically analyzed the compatibility of Elo with LLM evaluations by examining how well it adheres to reliability and transitivity. However, Elo is not the only ranking algorithm used in LLM evaluation. There exists a number of other algorithms that have been proposed to rank in different contexts, such as the Bradley-Terry model (Bradley and Terry, 1952) and Glicko (Glickman and Jones, 1999), which have been shown to outperform Elo in certain scenarios, but their performance in the context of LLM evaluation has not been systematically evaluated. However, several factors affect the performance of these algorithms, such as the distribution of the data, the number of models that are compared, and the number of pairwise comparisons. As such, there is a need for a systematic comparison of ranking algorithms in the context of LLM evaluation to identify the most suitable algorithm for different scenarios which this paper aims to address.

7 Conclusion

In this paper, we explore the application of several ranking algorithms in the context of evaluating LLMs. Through our experiments and analysis, we identify significant limitations in the ability of these algorithms to maintain stable and accurate ranking across varying dataset types and distribution patterns. To combat these challenges, we introduce a set of guidelines for selecting the most appropriate ranking system based on the specific characteristics of the evaluation task and available resources. We believe that our findings and recommendations will be valuable to researchers and practitioners working in the field of LLM evaluation, and we hope that our work will inspire further research in this area.

8 Limitations

While our study comprehensively explores the ranking of Large Language Models (LLMs) through head-to-head combat, several limitations warrant discussion.

Scalability Constraints One significant limitation arises from the scalability constraints inherent in pairwise evaluations. As the number of LLMs in the ecosystem expands, the number of required comparisons grows quadratically. This introduces computational challenges and resource constraints that may limit the feasibility of exhaustive head-to-head evaluations, particularly for larger sets of models.

Human Feedback Variability Another concern is the variability in human feedback, on which our rankings are heavily based. Human judgments can be subjective and influenced by numerous factors, including the individual's background, expertise, and contextual understanding. This variability can introduce noise into the ranking system, affecting the evaluations' stability and reliability.

References

Meriem Boubdir, Edward Kim, Beyza Ermis, Sara Hooker, and Marzieh Fadaee. 2023. Elo uncovered: Robustness and best practices in language model evaluation. *arXiv preprint arXiv:2311.17295*.

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

Thomas Callaghan, Peter J Mucha, and Mason A Porter. 2003. Random walker ranking for ncaa division i-a football. *arXiv*.

Thomas Callaghan, Peter J Mucha, and Mason A Porter. 2004. The bowl championship series: A mathematical review. *arXiv*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

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

H. A. David. 1988. *The Method of Paired Comparisons*, 2nd edition. Griffin's Statistical Monographs & Courses, Oxford University Press, London : New York. Revised edition, Griffin's statistical monographs & courses ; no. 41.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.

A.E. Elo. 1978. *The Rating of Chessplayers, Past and Present*. Arco Pub.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Mark E Glickman. 1999. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 48(3):377–394.

Mark E Glickman and Albyn C Jones. 1999. Rating the chess rating system. *CHANCE-BERLIN THEN NEW YORK-*, 12:21–28.

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

Chandra Irugalbandara, Ashish Mahendra, Roland Daynauth, Tharuka Kasthuri Arachchige, Krisztian Flautner, Lingjia Tang, Yiping Kang, and Jason Mars. 2023. Scaling down to scale up: A cost-benefit analysis of replacing openai's gpt-4 with self-hosted open source slms in production. *arXiv preprint arXiv:2312.14972*.

Maurice G Kendall and B Babington Smith. 1940. On the method of paired comparisons. *Biometrika*, 31(3/4):324–345.

Gary King and Langche Zeng. 2001. Logistic regression in rare events data. *Political analysis*, 9(2):137–163.

Paul Kvam and Joel S Sokol. 2006. A logistic regression/markov chain model for ncaa basketball. *Naval Research Logistics (NrL)*, 53(8):788–803.

Ricky Sanjaya, Jun Wang, and Yaodong Yang. 2022. Measuring the Non-Transitivity in Chess. *Algorithms*, 15(5):152.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems. *Preprint*, arXiv:1905.00537.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Preprint*, arXiv:1804.07461.

Minghao Wu and Alham Fikri Aji. 2023. Style over substance: Evaluation biases for large language models. *arXiv*.

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

A Appendix

A.1 Glicko Algorithm Details

For model $i \in M$, the rating and the rating deviation are updated as:

$$\theta_i' = \theta_i + \frac{q}{\frac{1}{\sigma_i^2} + \frac{1}{d^2}} \sum_{j=1}^{J} g(\sigma_j)(S_{ij} - p_{ij}) \tag{6}$$

$$\sigma_i' = \left(\frac{1}{\sigma_i^2} + \frac{1}{d^2}\right)^{-1} \tag{7}$$

where $q = \log(10)/400$, J is the set of opponents for i and $g(\sigma) = 1/\sqrt{1 + 3q^2\sigma^2/\pi^2}$. Given the rating θ_i and θ_j of model i and j, the expected outcome p_{ij} is calculated as[1]:

$$p_{ij} = \frac{1}{1 + 10^{-g(\sigma_j)(\theta_i - \theta_j)/400}} \tag{8}$$

$$d^2 = \left(q^2 \sum_{j=1}^{J} g(\sigma_j)^2 p_{ij}(1 - p_{ij})\right)^{-1} \tag{9}$$

where d^2 is the scaling factor for the system's certainty.

A.2 Markov Chain Details

The Markov Chain Model was originally developed as a random walker model Callaghan et al. (2003, 2004) where ranks were calculated by solving a series of differential equations. This method was adapted into a Markov Chain Model by (Kvam and Sokol, 2006). When expressed as a Markov chain, the elements T_{ij} of the transition matrix T are the probabilities that a walker moves from model i to j in a single time step, with the initial values computed as follows:

$$t_{ij} = \frac{1}{N_i}\left[w_{ij}(1-p) + l_{ij}p\right], \quad \forall j \neq i \tag{10}$$

$$t_{ij} = \frac{1}{N_i}\left[W_i p + L_i(1-p)\right] \tag{11}$$

where W_i, L_i are the number of matches won by model i and w_{ij}, l_{ij} are the number of games won by model i against model j.

The steady-state probability vector $\pi = [\theta_i]$ represents the final ranking of each team $i \in M$ such that $\pi T = \pi$ and $\sum_{i=1}^{m} \theta_i = 1$.

A.3 F1 scores per model

In this paper, F1 scores are used as an evaluation metric to assess the consistency and predictive performance of the ranking algorithms. The F1 score is calculated based on the precision and recall of the correctly predicted pairwise outcomes compared to the observed outcomes in the dataset. In the following, we detail the methodology for computing the F1 scores:

Let the data set consist of m pairwise matches between models. For each pair (i, j), p_{ij} is the probability that the model i beats j in a random match. If model i plays M_i matches then E_i, the expected number of games won by model i is $E_i = \lfloor M_i p_{ij} \rfloor$ and the actual number of games won is given by A_i.

The Precision and Recall are calculated as:

$$\text{precision} = \frac{|E_i \cap A_i|}{|E_i|}, \quad \text{recall} = \frac{|E_i \cap A_i|}{|A_i|}$$

[1] For $\sigma = 0$, p_{ij} defaults to the classic Elo rating system.

The F1 score is calculated as the harmonic mean of precision and recall:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

The total F1 score for a ranking algorithm is computed by averaging the F1 scores for all pairwise matches (i, j):

$$\text{F1 Score}_{\text{Overall}} = \frac{1}{|M|} \sum_{(i,j)\in M} \text{F1 Score}_{(i,j)},$$

where M is the set of all pairwise matches.

Figures 3 and 4 show the F1 scores for both the SLAM and Arena datasets, and also their overall F1 score.

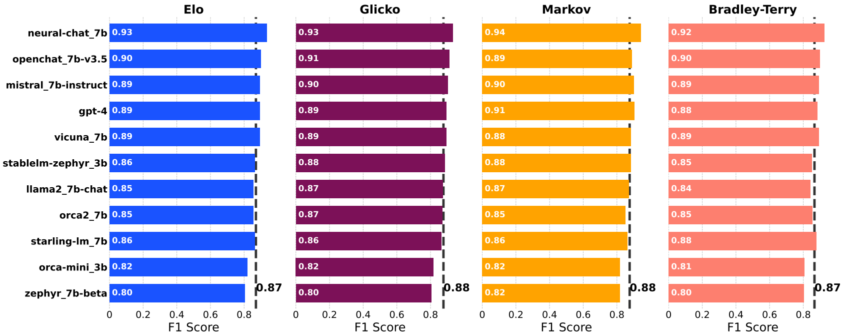


Figure 3: F1 scores for all models in the SLAM dataset.

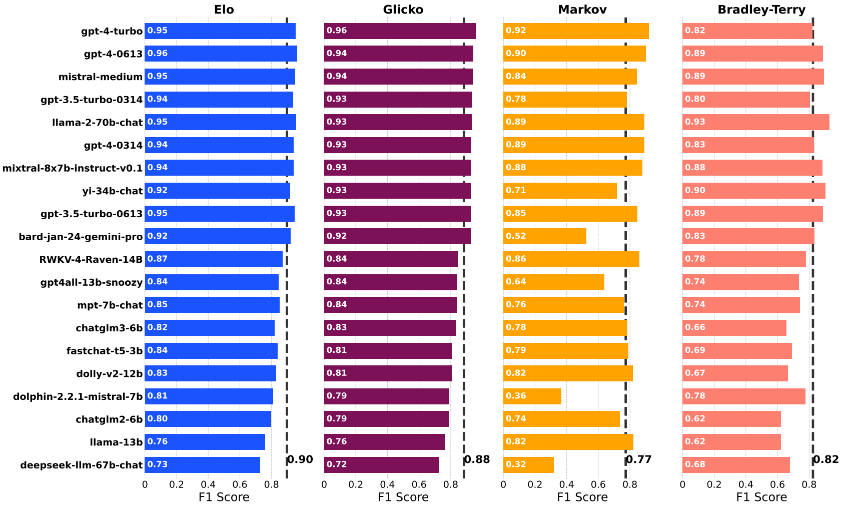


Figure 4: F1 scores of the top ten (10) highest and lowest ranked models using the Elo rating system on the Chatbot Arena dataset. In general, Elo provides the best prediction accuracy by achieving an F1 score of 0.90.

A.3.1 Experimental Setup

To evaluate the predictive accuracy of the ranking algorithms, each data set was split into a 75% training set and a 25% test set.

A.3.2 Procedure

The ranking algorithms were applied to the training set to compute model rankings based on pairwise matchups. Using these rankings, the algorithms predicted the outcomes of matchups in the test set, expressed as the number of games each model was expected to win. These predictions were compared with the actual results of the match and the precision, recall and F1 scores were calculated for each match.

The F1 scores, aggregated across all matches in the test set, quantified the alignment between the predicted and observed results, providing a measure of the ability of each algorithm to generalize to unseen data.

A.4 Elo's rank at $k = 3$ vs $k = 5$

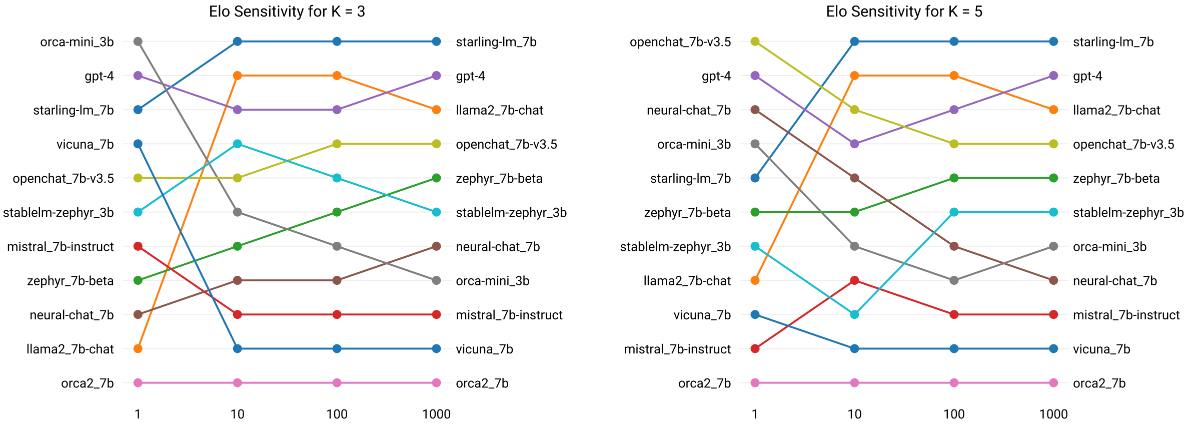
To analyze the sensitivity of Elo rankings to hyperparameter settings, we conducted experiments using two versions of Elo, with $K = 3$ and $K = 5$, on the SLAM dataset.

A.4.1 Experimental Setup

For each K-factor, we evaluated Elo rankings under varying numbers of permutations: $1, 10, 100$, and 1000. Permutations involve randomizing the order of matchups, a strategy commonly employed to reduce the effect of matchup order on final rankings and improve stability.

A.4.2 Procedure

For each K factor ($K = 3$ and $K = 5$), Elo rankings were calculated using the raw SLAM matchup data without permutations. Matchup orders were permuted $1, 10, 100$, and 1000 times for each factor K, with Elo rankings recalculated after each set of permutations. Our findings (shown in figure 5), show volatility of Elo's ranking for each of the k value.



(a) Ranks produced by Elo for $k = 3$

(b) Ranks produced by Elo for $k = 5$

Figure 5: Elo produces different ranks based on the value of the hyperparameter k. Increasing the number of permutation can lead of more stable ratings, however, model ranks may still be unstable as is the case with *orca-min_3b* and *neural-chat_7b*.

A.5 Elo's Instability at higher permutation

To analyze the performance of Elo at higher permutations, we conducted two separate experiments on the SLAM dataset.

A.5.1 Experimental Setup

For a single K factor of 25, we evaluated Elo rankings by performing two separate evaluations of each $10,000$ permutation using the strategy of the previous section.

A.5.2 Procedure

We observed the ranks produced by each experiment, showing that even with a large permutation, ELO can produce two different ranks for the same two models6.

A.6 Evaluating the consistency of crowd-workers evaluators

In this section, we recommend a method to evaluate the consistency of crowd-worker evaluations based on the Analytic hierarchy process (AHP) practices. We begin by formally defining cycles in a pairwise ranking dataset.

Definition A.1. For models $i_1, i_2, \cdots i_k \in M$, if $p_{i_1 i_2} > 0.5, p_{i_2 i_3} > 0.5, \cdots, p_{i_k i_1} > 0.5$, then their games is said to be *cyclic*. A cycle of length three is called a Rock, Paper, Scissors (RPS) cycle.

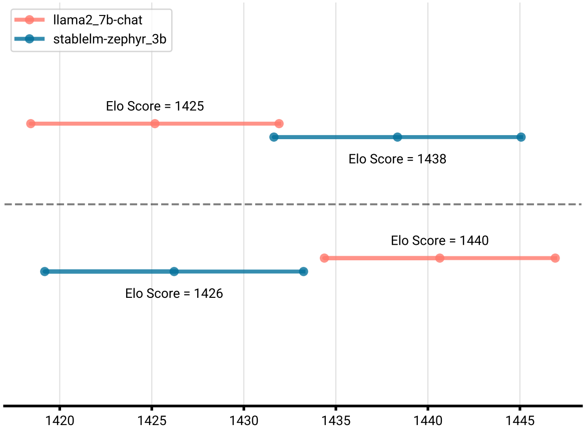


Figure 6: Two separate ranks produced by ELO for permutations of 10,000 each.

For any three models i, j, and k, it is expected that if a reviewer prefers i over j and j over k, they prefer i over k (transitivity). However, this may not always be due to personal preference, evaluation error, or evaluator negligence. If the transitive property does not hold, we say that the three models are intransitive and the evaluation is inconsistent.

Measuring the consistency of pairwise ranking falls under the domain of AHP. According to AHP theory, if the number of RPS cycles remains small, it reflects the complexity of the ranking problem rather than the judges' complacency. However, a high number of cycles represents a low quality of judgments.

According to (Kendall and Smith, 1940), the expected number of RPS cycles for a pairwise comparison dataset of m items follow a χ^2 distribution, and if the actual number of RPS cycles is significantly less than the expected number, we can be reasonably confident that the dataset is consistent.

The number of RPS cycles can be calculated by first generating the win rate matrix P as defined in section 4.2. This is followed by the adjacency matrix A, where $a_{ij} = 1$ if $p_{ij} > 0.5$ and 0 otherwise. Finally, the total number of RPS cycles is calculated as $\rho = 1/3 \sum_{i=1}^{M} a^*_{i,i}$, where a^*_{ii} are diagonals of A^3(Sanjaya et al., 2022).

The χ^2 statistic is calculated as:

$$\chi^2 = \frac{8}{k-4}\left\{\frac{1}{4}\binom{k}{3} - d - \frac{1}{3} + \rho\right\} \tag{12}$$

$$v = \frac{k(k-1)(k-2)}{(k-4)^2} \tag{13}$$

where v is the degrees of freedom.

The RPS count and χ^2 statistic for both the SLAM and Arena datasets are shown in table 6. At a p-value close to 0, we can be confident that inconsistencies are not due to the quality of the judges.

Table 6: Non-Transitivity statistic for the two datasets that were used. The p-value is calculated based on the χ^2 estimated via Kendal's Chi-squared approximation to the circular triad distribution.

Dataset	RPS Count	χ^2	p-value
SLAM	19	45.06	0.001
Arena	393	1107.25	0.000