Efficient Portfolio Selection through Preference Aggregation with Quicksort and the Bradley–Terry Model

Yurun Ge, Lucas Böttcher, Tom Chou, Maria R. D'Orsogna

***Abstract*—How to allocate limited resources to projects that will yield the greatest long-term benefits is a problem that often arises in decision-making under uncertainty. For example, organizations may need to evaluate and select innovation projects with risky returns. Similarly, when allocating resources to research projects, funding agencies are tasked with identifying the most promising proposals based on idiosyncratic criteria. Finally, in participatory budgeting, a local community may need to select a subset of public projects to fund. Regardless of context, agents must estimate the uncertain values of a potentially large number of projects. Developing parsimonious methods to compare these projects, and aggregating agent evaluations so that the overall benefit is maximized, are critical in assembling the best project portfolio. Unlike in standard sorting algorithms, evaluating projects on the basis of uncertain long-term benefits introduces additional complexities. We propose comparison rules based on Quicksort and the Bradley–Terry model, which connects rankings to pairwise "win" probabilities. In our model, each agent determines win probabilities of a pair of projects based on his or her specific evaluation of the projects' long-term benefit. The win probabilities are then appropriately aggregated and used to rank projects. Several of the methods we propose perform better than the two most effective aggregation methods currently available. Additionally, our methods can be combined with sampling techniques to significantly reduce the number of pairwise comparisons. We also discuss how the Bradley–Terry portfolio-selection approach can be implemented in practice.**

***Index Terms*—Portfolio selection, participatory budgeting, preference aggregation, Bradley–Terry model, social choice.**

I. Introduction

THE problem of allocating limited resources to select projects that offer the greatest benefit to stakeholders arises in many decision-making tasks. One of the main issues in project selection under uncertainty that it is often difficult to estimate the long-term benefit (or "value") of any given project, resulting in a large heterogeneity of estimates when multiple agents are queried. One example is project selection in organizational contexts, where multiple members are called to evaluate a number of innovation projects with uncertain returns [1], [2]. The goal is to select the most promising options while balancing the diverse member inputs. A related example is participatory budgeting [3]–[6], where communities must choose which public projects should receive funding. Regardless of the specific context, agents must evaluate the benefit associated with each of a large set of projects under conditions of uncertainty. Given the uncertain and heterogeneous inputs from various stakeholders, devising effective comparison and aggregation methods is crucial for reducing cognitive load on agents while selecting an optimal portfolio.

Prior research has examined the effectiveness of various aggregation methods, such as voting, averaging, and delegation to experts. An existing model of organizational decision-making [1] assumes that agents evaluate and approve one project at a time, without comparisons. Projects are characterized by both type and intrinsic value, while agents' evaluations depend on their specific expertise with regard to the project types. This model has been extended in [2] to address portfolio selection under budget constraints. Evaluating multiple projects with uncertain values and choosing a subset considered most valuable by agents is closely related to the multiwinner voting problem within the field of social choice [7], [8]. Similar to the favorable properties of the Borda method [8], [9] in multiwinner voting, it also performs well in portfolio selection when project costs are uniform [10]. One challenge in portfolio selection is that agents may need to compare a potentially large number of projects, especially when ranking them as is done in Borda counting. Although this is an important problem in a variety of applications, to date there has been little research on aggregation methods based on pairwise comparisons, which provide a more practical approach to ranking items.

Pairwise comparisons can help mitigate the cognitive burden associated with directly ranking a large number of projects. The human short-term memory is limited to processing around seven items at once ("Miller's law") [11], making ranking tasks inefficient as the number of projects increases. Additionally, pairwise comparisons have proven valuable in eliciting preferences and are essential in cases where direct estimation of project value is difficult due to psychological factors [12]. In machine learning, pairwise comparison is a well-studied problem, with research focusing on reducing the complexity of comparisons needed to recover a full ranking of n items. For example, comparison methods that achieve an expected lower bound of $\Omega(n)$ comparisons under certain conditions have been proposed [13]. Additionally, active-learning techniques that require no more than $\mathcal{O}(n\,\mathrm{polylog}(n))$ queries have been introduced [14].

Most existing research on sorting assumes that comparison results are consistent with the true underlying rankings. There

Yurun Ge is with the Department of Mathematics, California State University at Northridge, Los Angeles, USA, and the Department of Computational Medicine, University of California, Los Angeles, USA (e-mail: yurun97@ucla.edu).

Lucas Böttcher is with the Department of Computational Science and Philosophy, Frankfurt School of Finance and Management, Frankfurt a. M., Germany, and the Laboratory for Systems Medicine, University of Florida, Gainesville, USA (e-mail: l.boettcher@fs.de).

Tom Chou is with the Department of Mathematics, University of California, Los Angeles, USA (e-mail: tomchou@ucla.edu).

Maria R. D'Orsogna is with the Department of Mathematics, California State University at Northridge, Los Angeles, USA, and the Department of Computational Medicine, University of California, Los Angeles, USA (e-mail: dorsogna@csun.edu).

has been relatively little work on deriving rankings from noisy information, such as in portfolio selection, where agents provide imprecise estimates of project values. The authors of [15], [16] were among the first to discuss algorithms for sorting based on noisy information. These methods were later extended in [17] to enable parallel processing. In this paper, we propose pairwise-comparison rules based on the Bradley–Terry model [18], [19], which links rankings to win probabilities.

In Section II, we give an overview of related work, while in Section III, we present the Bradley–Terry model and describe recent advances in its algorithmic development. In Section IV, we connect the Bradley–Terry model to portfolio-selection theory and describe various methods, including a Quicksort approach, for aggregating project evaluations from different agents in Section V. In Section VI, we examine the performance of these aggregation methods to show that several of the methods we propose perform better than existing aggregation methods. We also demonstrate how our approach can be integrated with sampling techniques to reduce the number of comparisons, alleviating cognitive load on agents. Finally, in Section VII, we summarize our findings and discuss our results.

II. Related work

a) Bradley–Terry model: The Bradley–Terry model is a statistical method used to rank n items based on repeated pairwise comparisons. The model was introduced in 1929 by Zermelo to study tournament outcomes [18] and re-introduced in 1952 by the eponymous Bradley and Terry [19]. A common application is ranking chess players according to their results in matches against one another. Due its versatility, the Bradley–Terry model has also been applied to sports rankings, electoral preferences, social choice modeling, psychological and healthcare studies, and to other settings where relative comparisons are more practical than isolated evaluations. Many iterative algorithms have been proposed to determine the maximum likelihood estimator parameters that can best fit existing data [20], [21] and to accelerate algorithm convergence, such as Newman's iteration [22].

Extensions include including ties between items, incorporating ordering-based advantages (such as playing on one's home-field in sports), or multiple (instead of pairwise) comparisons [23]–[27]. Bradley–Terry models are also used in machine learning and reinforcement learning, as useful tools in ranking, preference learning, learning from feedback, reward shaping, and other problems involving human choices [28]–[30]. They are also used to compare and rank large language models (LLMs) through crowdsourced open platforms, or other expert evaluators. Since direct pairwise LLM comparisons involve computations that are $\mathcal{O}(n^2)$, novel comparison methods that use only a subset of Bradley–Terry type pairings have been recently introduced [31]. Other modern applications include comparing the ideological positions of US politicians using specifically tuned LLMs [32] and ranking video-game players based on skill so they can be properly paired in virtual matches. For example, the proprietary TrueSkill ranking system utilizes a Bayesian framework based on the Bradley–Terry model to incorporate uncertainties in player skills. It can also be applied to matches with more than two players [33], [34].

b) Group decisions and portfolio selection: A common problem in decision-theory is how to effectively aggregate many individual inputs into a collective output. This issue arises in voting systems, social choice problems, and organizational decision-making. Typical aggregation methods include treating all inputs equally (*e.g.*, by using the arithmetic mean), delegating to specific individuals (at random or based on given criteria), using majority rules, or biasing the outcome in favor of specific subgroups [35]. Hierarchical team-decision making assumes distributed expertise among tiered group members whose judgment is aggregated using probabilistic methods [36]. In addition to developing computationally efficient aggregation methods, research has also focused on other important aspects such as the legitimacy of voting systems [37] and their capacity to mitigate polarization [38].

A decision problem that frequently arises in organizational settings is that groups are tasked with selecting a subset of projects from a portfolio. To model organizational decision-making, agents can be considered as having specific expertise, while projects are characterized by an intrinsic value (representing their long-term benefit and assumed to be ground truth) and a defining type [1]. Agents do not know this intrinsic value and must evaluate it; the larger the discrepancy between project type and agent expertise, the larger the uncertainty in the evaluation. This model has been extended to include project costs and budget limitations, and has been applied to social choice problems [2], [10].

c) Sorting methods: Developing robust algorithms that operate effectively despite unreliable or noisy information, without removing specific noisy elements, is a problem that arises in many computing applications [39]. Unlike standard sorting algorithms that focus on ordering a list of precisely known values [40], comparing items with values affected by various sources of uncertainty requires adapted sorting approaches [15]–[17]. Several strategies have been developed to improve on these so-called "dirty" comparisons, for example using the inaccurate results in parallel with a subset of exact, "clean" comparisons to improve efficiency. The number of the required clean comparisons can be tuned based on the accuracy of the dirty comparisons [41]–[43]. In some applications, it may be sufficient to use a relatively small number of pairwise comparisons to obtain an approximate ranking. A lower bound on the number of necessary pairwise comparisons has been derived in [44]. Related work has also considered problems like computing a longest increasing subsequence associated with a given sequence of elements in the presence of comparison errors [45], [46].

III. The Bradley–Terry Model

Here, we discuss the Bradley–Terry model in more mathematical detail. The goal is to assign "strength" parameters to all players and rank them accordingly. The strength parameters determine the probability of a win, tie, or loss when two players are placed in competition. Specifically, if π_i and π_j represent the strengths of competitors i and j, respectively, the probability that i wins over j is $\pi_i/(\pi_i + \pi_j)$.

Estimation of the strength parameters is usually performed using maximum likelihood estimation. The basic idea is to maximize the log-likelihood function of the observed competition outcomes to find the most likely values of the players' strengths. Mathematically, given a set of observed outcomes w_{ij}, where w_{ij} is the number of times competitor i wins over competitor j, the log-likelihood function for these outcomes under the Bradley–Terry model is

$$\begin{aligned} l(\boldsymbol{\pi}) &= \sum_{i\neq j} w_{ij} \ln\left(\frac{\pi_i}{\pi_i+\pi_j}\right) \\ &= \sum_{i\neq j} w_{ij}\left[\ln(\pi_i) - \ln(\pi_i+\pi_j)\right], \end{aligned} \tag{1}$$

where $\boldsymbol{\pi} = (\pi_1, \dots, \pi_n)^\top$ is the strength parameter vector.

Maximizing the log-likelihood function $l(\boldsymbol{\pi})$ in Eq. (1) involves iterative updates of the parameters $\boldsymbol{\pi}$. Zermelo proved [18] that this maximization has a unique solution under certain conditions. The maximum can be found by differentiating $l(\boldsymbol{\pi})$ with respect to each parameter π_i and setting the resulting expressions to zero, resulting in an implicit expression for the strength parameters

$$\pi_i = \frac{\sum_{j\neq i} w_{ij}}{\sum_{j\neq i} \frac{w_{ij}+w_{ji}}{\pi_i+\pi_j}}. \tag{2}$$

Zermelo also proposed the first iterative approach [18] to solve Eq. (2). In each iteration step, one calculates

$$\pi_i' = \frac{\sum_{j\neq i} w_{ij}}{\sum_{j\neq i} \frac{w_{ij}+w_{ji}}{\pi_i+\pi_j}}, \tag{3}$$

where π_i' denotes the updated strength of competitor i starting from strengths π_i and $\pi_{j\neq i}$. This method is simple but can be slow to converge. More recently, Newman proposed an alternative iterative process [22] that is substantially faster than Zermelo's algorithm. His proposal is based on the iteration

$$\pi_i' = \frac{\sum_{j\neq i} \frac{w_{ij}\pi_j}{\pi_i+\pi_j}}{\sum_{j\neq i} \frac{w_{ji}}{\pi_i+\pi_j}}, \tag{4}$$

which converges faster than Zermelo's algorithm by a factor of $\sim 3 - -100$. Newman's iteration can be improved by incorporating updated values after each iteration, enhancing both convergence speed and stability as in the Gauss–Seidel method [47]. We adapt this iteration process and represent it as

$$\pi_i' = \frac{\sum_{j\neq i} \frac{w_{ij}\pi_j'}{\pi_i+\pi_j'} + \sum_{j>i} \frac{w_{ij}\pi_j}{\pi_i+\pi_j}}{\sum_{j\neq i} \frac{w_{ji}}{\pi_i+\pi_j'} + \sum_{j>i} \frac{w_{ji}}{\pi_i+\pi_j}}. \tag{5}$$

Note that the strength parameters can become ill-defined if a player never wins or never loses. For example, consider three players 1, 2, and 3 with match results: 1 wins against 2, 1 wins against 3, and 2 wins against 3. In such cases, the algorithm may converge to values where π_1 grows towards infinity at a faster rate than π_2, while π_3 converges to 0.

TABLE I
AN OVERVIEW OF THE MAIN MODEL PARAMETERS. UNLESS OTHERWISE STATED, ALL PARAMETERS ARE REAL-VALUED.

Symbol	Description
$N \in \mathbb{Z}^+$	Number of agents
$n \in \mathbb{Z}^+$	Number of items (or projects)
$n^* \in \mathbb{Z}^+$	Budget constraint
$i, j \in \{1, \dots, n\}$	Project label
$\ell \in \{1, \dots, N\}$	Agent label
$v_i \in \mathbb{R}^+$	Value of project i
$t_i \in [t_{\min}, t_{\max}]$	Type of project i
$e_\ell \in [e_{\min}, e_{\max}]$	Expertise of agent ℓ
$\beta \geq 0$	Knowledge breadth of agents
e_{M}	Mean expertise level; $e_{\mathrm{M}} = (t_{\min} + t_{\max})/2$
$v_{i\ell}$	Value of project i as evaluated by agent ℓ
$\eta_{i\ell} = v_{i\ell} - v_i$	Noise in perceived value of project i associated with agent ℓ
$\sigma_{i\ell} > 0$	Uncertainty of value of project i associated with agent ℓ
v_i'	Aggregate value of project i over all N agents
$w_{ij}^\ell \in (0,1)$	Win probability of project i outperforming project j as predicted by agent ℓ
$W^\ell \in (0,1)^{n\times n}$	Matrix of all win probabilities w_{ij}^ℓ as predicted by agent ℓ
$w_{ij}' \in (0,1)$	Aggregated win probability of project i outperforming project j
$W' \in (0,1)^{n\times n}$	Matrix of all aggregated win probabilities w_{ij}'

IV. PORTFOLIO SELECTION

To model the selection of projects from a portfolio under cost constraints, we build on the framework proposed in [2]. Each project $i \in \{1, \dots, n\}$ is characterized by two parameters: its type $t_i \in [t_{\min}, t_{\max}]$ and value $v_i \in \mathbb{R}^+$. In the project selection context, the values v_i define the true benefit of project i over a specific time horizon, if chosen. The true benefit may evolve and be uncertain over time due to societal value shifts, environmental changes, and complex interactions with other selected projects $j \neq i$. We do not consider these sources of uncertainty in v_i (which is here considered the ground truth) and restrict ourselves to each agent's uncertainty in the estimation of v_i at the time of evaluation. This leads to subjective evaluations $v_{i\ell}$ of project i from each agent $\ell \in \{1, \dots, N\}$. To represent $v_{i\ell}$ we first assume that each agent ℓ involved in the decision-making process has a level of expertise $e_\ell \in [e_{\min}, e_{\max}]$ given by

$$e_\ell = e_{\mathrm{M}} - \frac{N+1-2\ell}{N-1}\beta. \tag{6}$$

According to Eq. (6) the e_ℓ values are evenly spaced across the interval $[e_{\mathrm{M}} - \beta, e_{\mathrm{M}} + \beta] \equiv [e_{\min}, e_{\max}]$. Here, e_{M} represents the mean expertise level and β denotes the knowledge breadth that determines the expertise spread. For mathematical convenience, we set $e_{\mathrm{M}} = (t_{\min} + t_{\max})/2$ so that the mean expertise coincides with the mean project type.

The values t_i and e_ℓ do not have any specific meaning; they are simply labels used to differentiate between various types and expertise levels. However, the alignment between t_i and e_ℓ

affects the accuracy of $v_{i\ell}$, agent ℓ's evaluation of project i's value.[1] Specifically, we assume that the noise in the "perceived value," $\eta_{i\ell} = v_{i\ell} - v_i$, follows a normal distribution with standard deviation $\sigma_{i\ell} = |t_i - e_\ell|$. That is, $\eta_{i\ell} \sim \mathcal{N}(0, \sigma_{i\ell}^2)$, meaning that the closer the agent's expertise is to the project type, the lower the uncertainty. Each project is evaluated by N agents, and their aggregated preferences determine the final selection. Given a limited amount of resources, only a fixed number $n^* < n$ projects can be selected.

In this paper, we extend the described model of portfolio selection to pairwise comparisons between projects. Suppose agent ℓ is evaluating projects i and j, with perceived values $v_{i\ell}$ and $v_{j\ell}$, respectively, and corresponding noise terms $\eta_{i\ell}$ and $\eta_{j\ell}$. How would this agent evaluate the win probability w_{ij}^ℓ that project i is better than project j? As a starting point, we define

$$\begin{aligned} w_{ij}^\ell &:= \Pr\left(v_i > v_j\right) = \Pr\left((v_{i\ell} - \eta_{i\ell}) > (v_{j\ell} - \eta_{j\ell})\right) \\ &= \Pr\left((\eta_{i\ell} - \eta_{j\ell}) < (v_{i\ell} - v_{j\ell})\right). \end{aligned} \tag{7}$$

Under the assumption that the noise in the perceived value is independently and normally distributed, the quantity $\eta_{i\ell} - \eta_{j\ell}$ follows a normal distribution with a mean of 0 and a standard deviation of $\sqrt{\sigma_{i\ell}^2 + \sigma_{j\ell}^2}$. We can thus rewrite Eq. (7) as

$$w_{ij}^\ell = \Phi\left(\frac{v_{i\ell} - v_{j\ell}}{\sqrt{\sigma_{i\ell}^2 + \sigma_{j\ell}^2}}\right), \tag{8}$$

where Φ is the cumulative distribution function of the standard normal distribution. An immediate consequence of the above equation is that $w_{ij}^\ell = 1 - w_{ji}^\ell$.

In Table I, we provide an overview of the main model parameters used in this work. Some parameters, such as aggregated values and win probabilities, will be introduced in the next section, where we discuss various aggregation methods for identifying the most valuable projects within a given portfolio based on their performance $E(\beta; N, n, n^*)$. This quantity is defined as the expected value over n^* out of n projects that are evaluated by N agents with knowledge breadth β. We compute the expected value over a given type distribution.

As an example, we consider $N = 3$ agents, each with a knowledge breadth of $\beta = 0$, and $n = 3$ projects, from which $n^* = 2$ must be selected. The project values are $v_1 = 1$, $v_2 = 2$, and $v_3 = 3$. We assume that agents perceive the true project values (*i.e.*, $v_{i\ell} = v_i$ for $\ell \in \{1, 2, 3\}$). The performance for this example is calculated as $E(\beta = 0; N = 3, n = 3, n^* = 2) = v_2 + v_3 = 5$.

Some of the aggregation approaches that we study in this work will be based on project value estimates $v_{i\ell}$ while others will employ win probabilities w_{ij}^ℓ. We will show that aggregation methods using win probabilities instead of value estimates typically perform better.

[1] Our approach to describing domain-specific expertise is similar to that used in Hotelling models, where preferences are represented as distances along a line [48], [49].

V. Aggregation Methods

In the portfolio-selection model that we consider in this work, projects are chosen based on information collected from multiple agents. This information contains noise, and a key challenge is to design aggregation methods that effectively integrate all the agents' inputs to maximize the expected value of the selected projects. One possibility is to aggregate the estimated values provided by the agents using the arithmetic mean. However, value estimates may be difficult to ascertain in practice. Additionally, outliers can easily bias the arithmetic mean towards inaccurate value estimates. Previous research [2] has shown that a ranking-based method using Borda scores is more robust to outliers than the value-based arithmetic mean. Here, we leverage the win probabilities expressed in Eq. (8) and incorporate them into existing aggregation methods. The main advantage of this method is that rankings can be predicted directly from win probabilities that are associated with pairwise comparisons of projects, so that project selection can proceed without relying on explicit value estimates.

We will proceed with an overview of the aggregation methods that we employ in this work.

A. Overview

a) Arithmetic Mean: This method requires all agents to provide precise perceived values $v_{i\ell}$, which are then averaged using the arithmetic mean to obtain the aggregated value v_i'. That is,

$$v_i' = \frac{1}{N}\sum_{\ell=1}^{N} v_{i\ell}. \tag{9}$$

The n^* projects with largest aggregate values v_i' are then selected.

b) Borda Count: The Borda Count is based on the eponymous method introduced by Jean-Charles de Borda in the late 18th century [9]. In this approach, each agent ℓ ranks the n projects in descending order based on their perceived values $v_{i\ell}$. For each project i, we denote its position in agent ℓ's preference list by $\mathrm{pos}_\ell(i)$. The aggregated score s_i for project i is then calculated as the sum of the reversed ranks across all N agents. That is,

$$s_i = \sum_{\ell=1}^{N} (n - \mathrm{pos}_\ell(i)). \tag{10}$$

The top n^* projects with the highest aggregated scores are selected. According to [2], this method is particularly robust against misclassification and often outperforms the Arithmetic Mean, especially in scenarios with high uncertainty.

c) Quicksort: Quicksort is a widely used sorting algorithm, first introduced in [50], which employs a divide-and-conquer approach to sort elements. Its average-case time complexity is $\mathcal{O}(n \log(n))$, making it one of the most efficient sorting algorithms to date [51]. When applied to project selection, as demonstrated in Algorithm 1, the algorithm selects a "pivot" project from the middle of the list of available projects and partitions the remaining projects into two sub-lists: one containing projects ranked worse than the pivot, and the other containing projects ranked better than or equal to the pivot.

This partitioning process is recursively applied to each sublist. In our implementation, a project is considered better than the pivot if its aggregated win probability against the pivot exceeds 0.5. The aggregated win probability associated with projects i and j is

$$w'_{ij} = \frac{1}{N} \sum_{\ell=1}^{N} w^{\ell}_{ij}, \tag{11}$$

where w^{ℓ}_{ij} is given by Eq. (8).

The Quicksort method produces a list of projects ranked based on their aggregated win probabilities in ascending order, from which the last n^* projects are selected.

Algorithm 1 Quicksort with aggregated win-probability matrix

Require: Aggregated win-probability matrix W' of size $n \times n$
Ensure: Sorted index array idx

$idx \leftarrow$ list of integers from 0 to $n-1$
function PARTITION(low, $high$)
$i \leftarrow low - 1$
for $j \leftarrow low$ **to** $high - 1$ **do**
  if $W'[idx[j], idx[high]] < 0.5$ **then**
    $i \leftarrow i + 1$
    Swap($idx[i]$, $idx[j]$)
  end if
end for
Swap($idx[i+1]$, $idx[high]$)
return $i + 1$
end function
function QUICKSORTRECURSIVE(low, $high$)
if $low < high$ **then**
  $pi \leftarrow$ PARTITION(low, $high$)
  QUICKSORTRECURSIVE(low, $pi - 1$)
  QUICKSORTRECURSIVE($pi + 1$, $high$)
end if
end function
QUICKSORTRECURSIVE(0, $n - 1$)
return idx

d) Bradley–Terry Method: The Bradley–Terry model is usually employed in tournament settings, where the quantities w_{ij} are integers and represent the number of times that competitor i wins over competitor j. We use the Bradley–Terry model to devise a portfolio selection method that includes real-valued probabilities w^{ℓ}_{ij} as follows:

- First, each agent ℓ provides their predicted probabilities w^{ℓ}_{ij} for all pairwise comparison results. We use $W^{\ell} \in (0,1)^{n \times n}$ to denote the corresponding win-probability matrix.
- Then, aggregated win probabilities w'_{ij} are computed according to Eq. (11). We use $W' \in (0,1)^{n \times n}$ to denote the corresponding aggregated win-probability matrix.
- Next, Newman's iteration is used to determine the relative strength of each project using Eqs. (4) and (5).
- Finally, projects are selected in descending order of relative strength until the desired number of projects n^* is reached.

Since it may not be feasible for each agent to perform pairwise comparisons for all projects in the first step, we propose sampling approaches in Section V-C so that only a subset of pairwise comparisons are performed. In the second step, one may consider win-probability aggregation methods different from Eq. (11).

Given that the aggregation methods presented here involve different quantities (*i.e.*, values, scores, and win probabilities), we will now discuss some advantages and pitfalls associated with the (a-d) methods outlined above.

B. Values, scores, or win probabilities?

Aggregating win probabilities using Eq. (11) offers an advantage over employing the Arithmetic Mean as per Eq. (9), particularly when handling outliers in project-value evaluations. To illustrate this point, we consider three agents evaluating two projects, Project 1 and Project 2. The first agent holds a highly favorable view of Project 1, while the other two agents assign lower value estimates to it. If the first agent's evaluation is an outlier—say v_{11} approaches infinity—this outlier's effect differs significantly between the two methods.

With the arithmetic mean, the aggregated value for Project 1, v'_1, becomes highly skewed by the outlier and may approach infinity as well. This disproportionate influence from a single agent distorts the collective assessment of Project 1's value.

In contrast, when using the win probability aggregation method, the outlier's impact is mitigated. We assume that the extreme value from the first agent translates into a win probability of $w^1_{12} = 0.98$, indicating a strong preference. If the other two agents provide negative assessments of Project 1 with respect to Project 2, such as $w^2_{12} = w^3_{12} = 0.2$, the aggregated win probability, calculated using Eq. (11), results in $w'_{12} = 0.46$. This result is more closely aligned with the agents' evaluations than the one obtained with the Arithmetic Mean shown in Eq. (9).

Using win probabilities also offers an advantage over the Borda Count, as it more precisely captures individual preferences through real-valued probabilities. For example, consider two agents evaluating two projects, Project 1 and Project 2. The first agent strongly prefers Project 1 over Project 2 ($w^1_{12} = 0.8$), while the second agent only slightly favors Project 2 over Project 1 ($w^2_{12} = 0.46$). The aggregated win probability, $w'_{12} = 0.63$, indicates that Project 1 is the preferred choice overall, reflecting the stronger preference of the first agent. This approach takes into account the intensity of each agent's preference.

On the other hand, if the Borda Count is used, each project would receive a Borda score of 1, resulting in a tie. This outcome fails to differentiate between the strong preference expressed by the first agent and the more moderate preference of the second agent.

C. Sampling pairwise comparisons

When applying the Bradley–Terry method to portfolio selection, sampling pairwise comparisons (*i.e.*, selecting a subset of win probabilities w^{ℓ}_{ij}) can be a cost-effective strategy in practical implementations, as each additional comparison requires

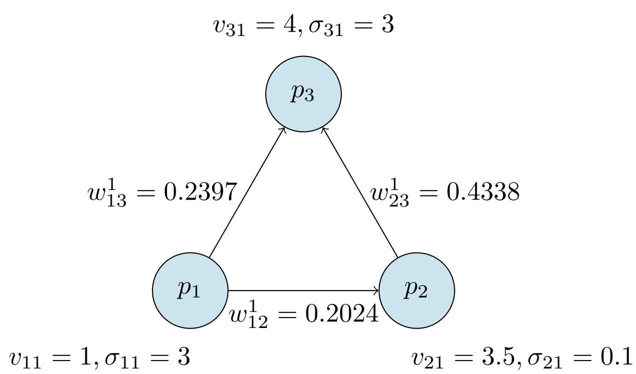

Fig. 1. Comparison of three projects p_1, p_2, and p_3 by a single agent. Each node represents a project and each directed edge represents a pairwise comparison between projects.

more resources. Moreover, we will show that incorporating sampling protocols into aggregation methods can improve performance.

To illustrate how different methods of comparing projects can influence the resulting rankings, consider a simple example involving a single agent evaluating three projects. The value estimates for these projects are $v_{11} = 1$, $v_{21} = 3.5$, and $v_{31} = 4$, with corresponding uncertainties $\sigma_{11} = 3$, $\sigma_{21} = 0.1$, and $\sigma_{31} = 3$. Depending on the sampling strategy employed, these pairwise comparisons can yield different rankings (see Figure 1). For instance, comparing Project 1 with Project 2 results in a win probability of $w_{12}^1 = 0.2024$, while comparing Project 2 with Project 3 yields $w_{23}^1 = 0.4338$. This sequence of comparisons leads to the ranking: Project 3 $\succ$ Project 2 $\succ$ Project 1, where $x \succ y$ indicates that x is strictly preferred over y. However, if we compare Project 1 with both Project 2 and Project 3, the win probabilities $w_{12}^1 = 0.2024$ and $w_{13}^1 = 0.2397$ produce a different ranking: Project 2 $\succ$ Project 3 $\succ$ Project 1. Thus, different methods of performing pairwise comparisons can lead to varying rankings.

Instead of performing all $\mathcal{O}(n^2)$ comparisons, we employ an $\mathcal{O}(n)$ cyclic graph sampling method. This technique can be visualized as extracting a subgraph from the complete graph generated by n projects. Given an ordered list of all p_i projects such as $(p_1, p_2, \ldots, p_n)$, the cyclic graph sampling of pairwise comparisons is defined as

$$((p_1, p_2), (p_2, p_3), \ldots, (p_{n-1}, p_n), (p_n, p_1)). \tag{12}$$

In this notation, the pairs in parentheses represent the pairwise comparisons conducted between these projects.

The performance of the cyclic graph sampling approach is influenced by how projects are initially ordered in the list. In this work, we employ a two-phase approach. In the first phase, an initial project-ranking approximation can be obtained through random sampling or by using existing ranking algorithms. This preliminary ranking then serves as the input for the second phase, where cyclic graph sampling is employed to refine the rankings through an optimization step based on the Bradley–Terry method.

We propose two additional aggregation methods using the cyclic graph sampling shown in Eq. (12). Both are based on a two-phase approach and are discussed below

e) Two-Phase Bradley–Terry method:

- In the first phase, we begin by generating an initial ranking using a list in which each project p_i $(i \in 1, \ldots, n)$ is selected uniformly at random without replacement from the n available projects. Next, the values of w'_{ij} are calculated via Eq. (11) according to the cyclic graph sampling of the randomly ordered list as shown in (12). Finally, Newman's iteration illustrated in Eqs. (4) and (5) is applied to the win probabilities w'_{ij} to obtain an approximate ranking.
- In the second phase, starting from the approximate ranking, we compute the corresponding win probabilities w'_{ij} using cyclic graph sampling (12). To further refine the ranking, we apply Newman's iteration again, while also incorporating the win probabilities obtained in the first phase. Win probabilities that were not calculated in either phase one or phase two are set to 0.

f) Two-Phase Quicksort:

- In the first phase, we approximate the project ranking using the Quicksort algorithm. Instead of relying on randomly selected pairwise comparisons, here we apply the Quicksort algorithm to the matrix of aggregated win probabilities W', with elements w'_{ij} (as shown in Eq. (11)), to generate an initial ranking of items. During this step, only the necessary entries of the aggregated win probabilities W' are sampled, resulting in an $\mathcal{O}(n \log(n))$ complexity. In addition to calculating w'_{ij}, the Quicksort algorithm also produces an initial ranking. However, because the underlying perceived values are noisy observations, the output of the Quicksort algorithm does not represent the true ranking as it would in the absence of uncertainty.
- In the second phase, starting from the Quicksort ranking, we compute the corresponding win probabilities w'_{ij} using cyclic graph sampling (12). Newman's iteration shown in Eqs. (4) and (5)) is then applied to determine a refined ranking. Unlike in the Two-Phase Bradley–Terry method, we only consider win probabilities associated with the cyclic graph structure and not those obtained in the first phase.

VI. Simulation Results

We now compare the effectiveness of the aggregation methods (a–f) in achieving a high expected value for the selected projects, as quantified by the performance measure $E(\beta; N, n, n^*)$. This is a measure that quantifies the expected value of the n^* (out of n) selected projects, each evaluated by N agents with with knowledge breadth β. Recall that the knowledge breadth determines the spread in agents' expertise according to Eq. (6). Following prior work [1], [2], [10], we calculate the expected value over the project-type distribution $\mathcal{U}(0, 10)$. The expertise value of the central decision maker is set at $e_{\mathrm{M}} = (t_{\min} + t_{\max})/2 = 5$. In our simulations, we consider a scenario with $n = 30$ projects, $N = 3$ agents, and a target of selecting $n^* = 15$ projects. We define the value of project i as $v_i = i$ $(i \in \{1, \ldots, 30\})$. The uncertainty in agent ℓ's project evaluations is quantified by additive Gaussian

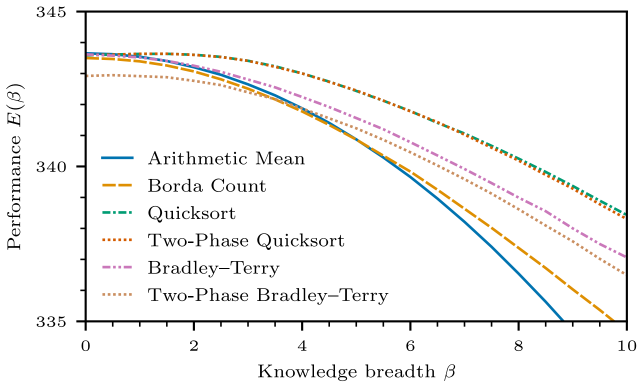

Fig. 2. Portfolio selection with $n = 30$ projects, $N = 3$ agents, and a target of selecting $n^* = 15$ projects. We show the performance $E(\beta; N, n, n^*)$ as a function of the knowledge breadth β for the six aggregation methods (a–f).

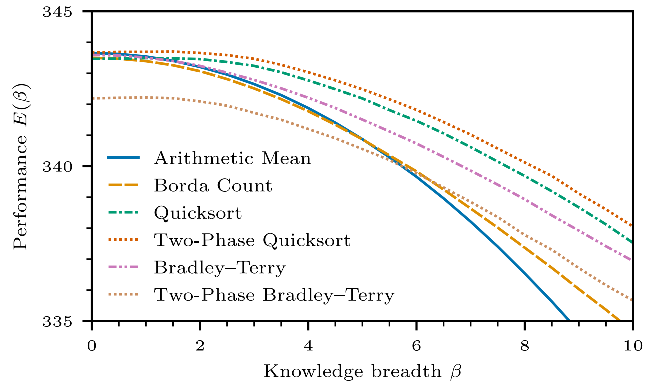

Fig. 3. Portfolio selection with $n = 30$ projects, $N = 3$ agents, and a target of selecting $n^* = 15$ projects. We show the performance $E(\beta; N, n, n^*)$ as a function of the knowledge breadth β for the six aggregation methods (a–f). In all approaches that are based on win probabilities, agents select one of the following values $\{0.01, 0.1, 0.2, \ldots, 0.9, 0.99\}$.

noise with zero mean and standard deviation $\sigma_{i\ell} = |t_i - e_\ell|$, where the expertise level e_ℓ of agent ℓ is given by Eq. (6). Prior work [2] has demonstrated that variations in value distribution, type distribution, and other parameters rarely affect the relative ordering of aggregation-rule performance.

All our results are based on Monte Carlo simulations. For methods based on pairwise comparisons and win probabilities, we use 100,000 independent and identically distributed samples. For the remaining two methods, Arithmetic Mean and Borda Count, which are computationally less demanding, we increase the sample size to 500,000. The theoretical maximum performance is $\sum_{i=16}^{30} v_i = \sum_{i=16}^{30} i = 345$.

We consider two scenarios for computing the win probabilities w'_{ij}. In the first scenario, the probabilities are calculated according to Eqs. (8) and (11). However, in real-world applications of aggregation methods based on pairwise comparisons and win probabilities, assigning probabilities with several decimal places may be impractical. Therefore, in the second scenario, we prespecify a set of win probabilities from which agents can choose when making pairwise comparisons.

A. Continuous win probabilities

In Figure 2, we show the performance $E(\beta; N = 3, n = 30, n^* = 15)$ for the six aggregation methods (a–f) as a function of knowledge breadth β. Prior work [2] has highlighted that both the Arithmetic Mean and the Borda Count are effective in identifying high-value projects within a portfolio. In particular, the Borda Count is more robust to evaluation outliers than the Arithmetic Mean and performs well across a wide range of model parameters. Our results in Figure 2 show that Quicksort (c), Two-Phase Quicksort (f), and the Bradley–Terry method using all pairwise comparisons (d), outperform both the Arithmetic Mean (a) and Borda Count (b) methods, particularly at higher values of β. The Two-Phase Bradley–Terry method (e), which employs a cyclic graph sampling approach of pairwise comparisons, performs worse than both the Arithmetic Mean (a) and Borda Count (b) for knowledge breadths $\beta \lesssim 5.5$.

As a robustness check, we also conducted simulations for $N = 15$ and $N = 30$ agents. We observed that the performance of all methods and observed that for all six methods the performance increases with increasing N, while their relative performance remains similar. Additionally, the performance gap between the Two-Phase Bradley–Terry method and the other methods widens. This is because the Two-Phase Bradley–Terry method uses a sampling protocol that leaves more entries in the aggregated win probability matrix W' empty, compared to other methods based on win probabilities.

B. Discrete win probabilities

In practical applications of the Bradley–Terry method, it may be necessary to prespecify a set of win probabilities w_{ij}^{ℓ} from which agents can choose. For example, one could use a set of prespecified probabilities such as $\{0.01, 0.1, 0.2, \ldots, 0.9, 0.99\}$. This approach simplifies the win-probability values by limiting them to a finite and manageable set, which is useful in decision-making scenarios where achieving a high degree of precision is not feasible.

In Figure 3, we show a comparison of the aggregation methods (a–f) where the win probabilities are restricted to values taken from the set $\{0.01, 0.1, 0.2, \ldots, 0.9, 0.99\}$. The relative performance ranking of the methods remains unchanged. However, the performance values of Quicksort and Two-Phase Quicksort exhibit a greater difference compared to the continuous case shown in Figure 2. Recall that the Two-Phase Quicksort method employs a refinement phase in which the final ranking is computed according to Newman's iteration (see Eqs. (4) and (5)). While this second phase had little impact on performance in the continuous case, it substantially affected results when using the prespecified win probabilities listed above. This aligns with the intuition that Newman's iteration (or similar iterative methods used in the Bradley–Terry method) performs well in scenarios where rankings are derived from a limited set of tournament outcomes.

The Arithmetic Mean and Borda Count methods rely on value estimates and scores, respectively, rather than win probabilities. Assigning value estimates can, in principle, be done di-

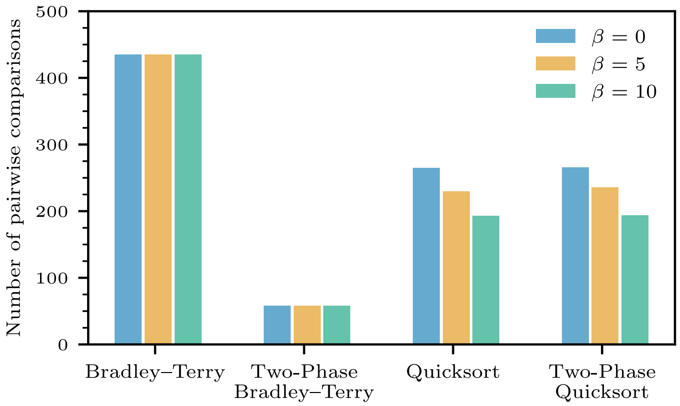

Fig. 4. Number of pairwise project comparisons across aggregation methods (c–f) for knowledge breadths $\beta \in \{0, 5, 10\}$ used to determine the performance $E(\beta)$ in Figure 3.

rectly without comparing projects. However, assigning ranking scores, as in the Borda Count, requires project comparisons. In our implementation of the Borda Count, we sort the agents' value estimate lists to generate the final rankings based on Eq. (10). When using Quicksort to obtain the scores, the average number of comparisons is $\mathcal{O}(n\ \log(n))$.

For the remaining methods that are based on win probabilities, the average number of comparisons are as follows:

- Bradley–Terry (all pairwise comparisons): $\mathcal{O}(n^2)$
- Two-Phase Bradley–Terry: $\mathcal{O}(n)$
- Quicksort (without a second refinement phase): $\mathcal{O}(n\ \log(n))$
- Two-Phase Quicksort: $\mathcal{O}(n\ \log(n))$

In Figure 4, we show the number of pairwise comparisons for each of these four approaches. The standard Bradley–Terry aggregation method considers all $30(30-1)/2 = 435$ possible comparisons, regardless of the value of β. The Two-Phase Bradley–Terry method involves approximately 58 comparisons for the given values of β. For Quicksort without a second refinement phase, the number of pairwise comparisons decreases from 265 for $\beta = 0$ to 193 for $\beta = 10$. The Two-Phase Quicksort approach results in slightly more comparisons, with 266 for $\beta = 0$ and 194 for $\beta = 10$.

The observed decrease in the number of pairwise comparisons results from the interplay between noise in the perceived values and the divide-and-conquer nature of Quicksort. The algorithm compares elements against a pivot to split the dataset into two parts. The fastest case occurs when the two parts contain an equal number of projects, while the slowest case happens when one part contains all the remaining projects, and the other part contains none. As the noise in project values increases with β, it becomes less likely that highly imbalanced sublists arise during Quicksort recursion. Therefore, the algorithm is expected to be more efficient for large values of β.

Although the Bradley–Terry, Quicksort, and Two-Phase Quicksort methods demonstrate the best performance in the simulations considered, the number of pairwise comparisons they require is likely too high for practical applications. In contrast, the Two-Phase Bradley–Terry method achieves favorable performance with a relatively small number of pairwise comparisons, making it the most practical approach for real-world use.

VII. Conclusions

In this work, we compared and contrasted six aggregation methods (a-f) for portfolio selection of projects with uncertain values. Of these, four novel methods (c-f) are based on pairwise comparisons. Agents are tasked with selecting a subset of available projects. The accuracy of their evaluations improves when the agents' expertise aligns well with the project types. However, when there is a misalignment between expertise and project types, evaluations are more prone to errors.

Agents may assign estimated values to projects, which can then be aggregated to make final decisions about which projects to select. When value estimates are difficult to ascertain, or when there is a risk of outlier evaluations due to mismatches between expertise and project types, it may be more appropriate to use Borda-type methods. These methods rank projects based on their perceived value, and the rankings are aggregated. However, ranking projects can also be challenging, especially when agents must evaluate a large number of projects relative to each other.

To address this challenge, we established a connection between portfolio selection and the Bradley–Terry model in which rankings of items are derived from pairwise comparisons and the associated win probabilities. We proposed two main aggregation methods based on win probabilities. The first method uses an extension of Quicksort to produce project rankings based on aggregate win probabilities, with a computational complexity of $\mathcal{O}(n \log n)$. The second approach employs Newman's iteration to compute rankings from a set of pairwise comparisons and their corresponding win probabilities. To reduce the number of comparisons, we introduced a cyclic graph sampling method, which achieved favorable performance with $\mathcal{O}(n)$ comparisons instead of the $\mathcal{O}(n^2)$ required for all possible pairwise comparisons. Similar graph structures have also been studied in the context of subgraph matching [52].

The methods we propose have applications in participatory budgeting, social choice, organizational decision-making, and other resource allocation problems involving decision-making under uncertainty. Furthermore, our sampling and ranking methods can be effectively applied to benchmark foundation models such as LLMs [31], [53], [54].

An interesting direction for future research is to extend the proposed Bradley–Terry aggregation methods by incorporating delegation strategies, where only agents with suitable expertise are queried, or by using approaches based on the median instead of the arithmetic mean. A related promising direction is to examine how interactions between agents shape their evaluations, using models of social influence on networks [55]–[57].

Another avenue for future work is the study of sampling methods akin to our cyclic graph sampling method that can achieve good performance with a subset of all pairwise comparisons. Identifying ways to sparsify the aggregate win probability matrix W' can make our proposed method more practical.

Finally, our work focused on uniform project costs, a single type distribution, and a single set of project values. These assumptions could be relaxed to examine the impact of heterogeneous project costs and of other type and value distributions. A broader analysis along these lines would provide further insight into the performance of aggregation methods based on pairwise comparisons and corresponding win probabilities.

ACKNOWLEDGMENTS

The authors thank Ronald Klingebiel for helpful discussions. This work was supported by the Army Research Office through W911NF-23-1-0129 (YG, LB, MRD) and by the National Science Foundation through grant MRI-2320846 (MRD). LB also acknowledges funding from hessian.AI.

REFERENCES

[1] F. A. Csaszar and J. Eggers, "Organizational decision making: An information aggregation view," *Manag. Sci.*, vol. 59, no. 10, pp. 2257–2277, 2013.

[2] L. Böttcher and R. Klingebiel, "Organizational selection of innovation," *Organ. Sci.*, 2024.

[3] B. Wampler, "A guide to participatory budgeting," in *Participatory Budgeting* (A. Shah, ed.), pp. 21–54, Washington, DC: The World Bank, 2007.

[4] H. Aziz and N. Shah, "Participatory budgeting: Models and approaches," in *Pathways Between Social Science and Computational Social Science: Theories, Methods, and Interpretations* (T. Rudas and G. Péli, eds.), pp. 215–236, Cham, Switzerland: Springer International Publishing, 2021.

[5] G. Benade, S. Nath, A. D. Procaccia, and N. Shah, "Preference elicitation for participatory budgeting," *Manag. Sci.*, vol. 67, pp. 2813–2827, 2021.

[6] J. C. Yang, C. I. Hausladen, D. Peters, E. Pournaras, R. Hnggli Fricker, and D. Helbing, "Designing digital voting systems for citizens: Achieving fairness and legitimacy in participatory budgeting," *Digit. Gov.: Res. Pract.*, vol. 5, no. 3, pp. 1–30, 2024.

[7] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko, "Properties of multiwinner voting rules," *Soc. Choice Welfare*, vol. 48, pp. 599–632, 2017.

[8] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, eds., *Handbook of Computational Social Choice*. Cambridge, UK: Cambridge University Press, 2016.

[9] J.-C. de Borda, "Mémoire sur les élections au scrutin," *Histoire et Mémoires de l'Académie royale des sciences*, pp. 657–665, 1781.

[10] Y. Ge, L. Böttcher, T. Chou, and M. R. D'Orsogna, "A knapsack for collective decision-making," 2024. arXiv:2409.13236.

[11] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychol. Rev.*, vol. 63, no. 2, p. 81, 1956.

[12] W. S. Torgerson, *Theory and Methods of Scaling*. New York, NY, USA: Wiley, 1958.

[13] F. Wauthier, M. Jordan, and N. Jojic, "Efficient ranking from pairwise comparisons," in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 109–117, PMLR, 17–19 Jun 2013.

[14] N. Ailon, "An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity," *J. Mach. Learn. Res.*, vol. 13, pp. 137–164, 2012.

[15] M. Braverman and E. Mossel, "Noisy sorting without resampling," in *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20-22, 2008* (S. Teng, ed.), (Philadelphia, PA), pp. 268–276, SIAM, 2008.

[16] M. Braverman and E. Mossel, "Sorting from noisy information," 2009. arXiv preprint arXiv:0910.1191.

[17] M. Braverman, J. Mao, and S. M. Weinberg, "Parallel algorithms for select and partition with noisy comparisons," in *Proceedings of the 48th annual ACM symposium on Theory of Computing*, (New York, NY, USA), pp. 851–862, Association for Computing Machinery, 2016.

[18] E. Zermelo, "Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung," *Math. Z.*, vol. 29, no. 1, pp. 436–460, 1929.

[19] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[20] L. R. Ford, "Solution of a ranking problem from binary comparisons," *Am. Math. Mon.*, vol. 64, pp. 28–33, 1957.

[21] D. R. Hunter, "MM algorithms for generalized Bradley-Terry models," *Ann. Stat.*, vol. 32, pp. 384–406, 2004.

[22] M. Newman, "Efficient computation of rankings from pairwise comparisons," *J. Mach. Learn. Res.*, vol. 24, no. 238, pp. 1–25, 2023.

[23] R. N. Pendergrass and R. A. Bradley, "Ranking in triple comparisons," in *Contributions to probability and statistics: Essays in honor of Harold Hotelling* (I. Olkin, S. G. Ghurye, W. Hoeffding, W. G. Madow, and H. B. Mann, eds.), pp. 331–351, Palo Alto, CA, USA: Stanford University Press, 1960.

[24] P. V. Rao and L. L. Kupper, "Ties in paired-comparison experiments: A generalization of the Bradley–Terry model," *J. Am. Stat. Assoc.*, vol. 62, pp. 194–204, 1967.

[25] R. R. Davidson, "On extending the bradley-terry model to accommodate ties in paired comparison experiments," *J. Am. Stat. Assoc.*, vol. 65, pp. 317–328, 1970.

[26] A. Agresti, *Categorical Data Analysis*. New York, NY, USA: Wiley, 1990.

[27] J. I. Marden, *Analyzing and Modeling Rank Data*. London, UK: Chapman and Hall, 1995.

[28] C. Muslimani and M. E. Taylor, "Leveraging sub-optimal data for human-in-the-loop reinforcement learning," in *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, (Richland, SC), pp. 2399–2401, International Foundation for Autonomous Agents and Multiagent Systems, 2024.

[29] J. Wainer, "A Bayesian Bradley-Terry model to compare multiple ML algorithms on multiple data sets," *J. Mach. Learn. Res.*, vol. 24, no. 341, pp. 1–34, 2023.

[30] M. Peschl, A. Zgonnikov, F. A. Oliehoek, and L. C. Siebert, "Moral: Aligning ai with human norms through multi-objective reinforced active learning," in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, (Richland, SC), p. 1038–1046, International Foundation for Autonomous Agents and Multiagent Systems, 2022.

[31] A. Liusie, V. Raina, Y. Fathullah, and M. Gales, "Efficient LLM Comparative Assessment: a Product of Experts Framework for Pairwise Comparisons," 2024. arXiv preprint arXiv:arXiv:2405.05894.

[32] P. Y. We, J. Nagler, J. A. Tucker, and S. Messing, "Large language models can be used to estimate the latent positions of politicians," 2023. arXiv preprint arXiv:2303.12057v4.

[33] R. Herbrich, T. P. Minka, and T. Graepel, "TrueSkill™: A Bayesian Skill Rating System," in *Advances in Neural Information Processing Systems* (B. Schölkopf, J. Platt, and T. Hoffman, eds.), vol. 19, MIT Press, 2006.

[34] T. P. Minka, R. Cleven, and Y. Zaykov, "TrueSkill 2: An improved Bayesian skill rating system," 2018.

[35] H. J. Einhorn, R. M. Hogarth, and E. Klempner, "Quality of group judgment.," *Psychological Bulletin*, vol. 84, no. 1, p. 158, 1977.

[36] S. E. Humphrey, J. R. Hollenbeck, C. J. Meyer, and D. R. Ilgen, "Hierarchical team decision-making," *Research in Personnel and Human Resources Management*, vol. 21, pp. 175–213, 2002.

[37] C. I. Hausladen, R. Hänggli Fricker, D. Helbing, R. Kunz, J. Wang, and E. Pournaras, "How voting rules impact legitimacy," *Humanit. Soc. Sci. Commun.*, vol. 11, no. 1, pp. 1–10, 2024.

[38] C. Alós-Ferrer and J. Buckenmaier, "Voting for compromises: alternative voting methods in polarized societies," *University of Zurich, Department of Economics, Working Paper*, no. 394, 2021.

[39] U. Feige, P. Raghavan, D. Peleg, and E. Upfal, "Computing with noisy information," *SIAM Journal on Computing*, vol. 23, no. 5, pp. 1001–1018, 1994.

[40] C. Zopounidis and M. Doumpos, "Multicriteria classification and sorting methods: A literature review," *Eur. J. Oper. Res.*, vol. 138, pp. 229–246, 2002.

[41] X. Bai and C. Coester, "Sorting with predictions," in *Proceedings of the 37th International Conference on Neural Information Processing Systems* (A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds.), NIPS '23, (Red Hook, NY, USA), pp. 26563–26584, Curran Associates Inc., 2024.

[42] P. Lu, X. Ren, E. Sun, and Y. Zhang, "Generalized sorting with predictions," in *4th Symposium on Simplicity in Algorithms, SOSA* (H. V.

Le and V. King, eds.), (Philadelphia, PA), pp. 111–117, Society for Industrial and Applied Mathematics, 2021.

[43] T. H. Chan, E. Sun, and B. Wang, "Generalized sorting with predictions revisited," in *Frontiers of Algorithmics - 17th International Joint Conference, IJTCS-FAW 2023 Macau, China, August 14-18, 2023, Proceedings* (M. Li, X. Sun, and X. Wu, eds.), vol. 13933 of *Lecture Notes in Computer Science*, (Cham, Switzerland), pp. 29–41, Springer International Publishing, 2023.

[44] J. Giesen, E. Schuberth, and M. Stojaković, "Approximate sorting," *Fundamenta Informaticae*, vol. 90, no. 1–2, p. 67–72, 2009.

[45] B. Geissmann, "Longest increasing subsequence under persistent comparison errors," *Theory Comput. Syst.*, vol. 64, pp. 662–680, 2020.

[46] B. Geissmann, "Longest increasing subsequence under persistent comparison errors," *Theory Comput. Syst.*, vol. 64, no. 4, pp. 662–680, 2020.

[47] R. Kress, *Numerical Analysis*, vol. 181. Berlin, Germany: Springer Science & Business Media, 2012.

[48] H. Hotelling, "Stability in competition," *Econ. J.*, vol. 39, no. 153, pp. 41–57, 1929.

[49] W. Novshek, "Equilibrium in simple spatial (or differentiated product) models," in *Noncooperative Approaches to the Theory of Perfect Competition* (A. Mas-Colell, ed.), pp. 199–212, Cambridge, MA, USA: Academic Press, 1982.

[50] C. A. Hoare, "Quicksort," *Comput. J.*, vol. 5, pp. 10–16, 1962.

[51] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. Cambridge, MA, USA: MIT Press, 2022.

[52] Y. Ge, D. Yang, and A. L. Bertozzi, "Iterative active learning strategies for subgraph matching," *Pattern Recognition*, vol. 158, p. 110797, 2025.

[53] G. Zhang and M. Hardt, "Inherent trade-offs between diversity and stability in multi-task benchmark," *arXiv preprint arXiv:2405.01719*, 2024.

[54] A. Ghosh, S. Dziadzio, A. Prabhu, V. Udandarao, S. Albanie, and M. Bethge, "ONEBench to test them all: Sample-level benchmarking over open-ended capabilities," *arXiv preprint arXiv:2412.06745*, 2024.

[55] A. Stomakhin, M. B. Short, and A. L. Bertozzi, "Reconstruction of missing data in social networks based on temporal patterns of interactions," *Inverse Problems*, vol. 27, no. 11, p. 115013, 2011.

[56] L. Böttcher, J. Nagler, and H. J. Herrmann, "Critical behaviors in contagion dynamics," *Physical Review Letters*, vol. 118, no. 8, p. 088301, 2017.

[57] J. R. Zipkin, F. P. Schoenberg, K. Coronges, and A. L. Bertozzi, "Point-process models of social network interactions: Parameter estimation and missing data recovery," *European Journal of Applied Mathematics*, vol. 27, no. 3, pp. 502–529, 2016.