The Art of Calling the Winner by Asking Just Enough Questions

Nisarg Shah[1] **Ziqi Yu**[1]

[1]Department of Computer Science, University of Toronto, Canada

Abstract

We study active elicitation of agent preferences for collectively choosing among m alternatives using prominent voting rules. We focus on the next-best query model, in which an agent responds to a query by revealing their next favorite alternative, and measure the competitive ratio, which is the worst-case ratio between the number of queries made by the active elicitation algorithm and the minimum number of queries needed to reveal the winning alternative(s) in hindsight. We show that the best competitive ratio is sublinear in m for many positional scoring rules but linear in m for all Condorcet-consistent rules. Our analysis centers on a simple elicitation algorithm, which not only achieves optimal theoretical bounds, but also impressive empirical performance on real data.

1 INTRODUCTION

Voting is a fundamental primitive for multiagent decision-making. Beyond political elections, voting mechanisms are routinely invoked to aggregate preferences in human computation systems [Mao et al., 2013], to combine predictions in ensemble systems [Pennock et al., 2000], and increasingly to arbitrate among AI agents in multi-LLM architectures [Zhao et al., 2024]. In all these settings, a group must collectively select a single alternative from a set of m alternatives based on the ranked preferences of n agents.

Social choice theory provides a rich menu of voting rules, each justified by compelling normative principles [Brandt et al., 2016]. A central modeling assumption underlying this literature is that the complete preference ranking of every agent over the alternatives is available upfront. This assumption is reasonable when m is small, but unrealistic in modern decision-making environments that select from a large pool of alternatives: for example, deliberation platforms such as Polis and Remesh induce voting over thousands of proposed statements [Small et al., 2021, Konya et al., 2023, 2025] and generative systems such as the Habermas machine [Tessler et al., 2024] find a common ground by searching over an astronomically large space of statements.

This tension has motivated a growing body of work on voting with limited preference information. Given limited information, prior work studies possible and necessary winners—alternatives that win under some and all completions of the available preference information, respectively—under common voting rules [Konczak and Lang, 2005, Xia and Conitzer, 2011]. When such information can be actively elicited, prior work aims to make just enough queries to reveal the winners and measures the *communication complexity*, i.e., the worst-case number of queries needed [Conitzer and Sandholm, 2005, Service and Adams, 2012].

However, most interesting voting rules demand eliciting the entire preference profile in the worst case, unless one makes structural assumptions on the preferences [Conitzer, 2007]. Arguably, making many queries is less problematic on *hard instances* in which fewer queries could not have revealed the winners, but more problematic on *easy instances* in which a small number of queries could have revealed the winners in hindsight. This motivates the study of *competitive ratio* [Sleator and Tarjan, 1985, Borodin and El-Yaniv, 2005], which is the worst-case ratio between the number of queries made by the algorithm and the minimum possible number of queries needed to reveal the winners in hindsight.

Hosseini et al. [2021] are the first to adopt this approach to social choice. They seek a desirable one-to-one matching of n agents to n objects in the next-best query model, in which an algorithm is allowed to query an agent's k^{th} most preferred object only after querying her first $k-1$ favorite objects. This model captures sequential top-k interfaces, where the system gradually asks each voter to extend her revealed prefix by one additional alternative, and stops once the winner can be certified. Adapting this to voting, we initiate the study of the *competitive ratio for winner determi-*

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:6149–6164.

nation under prominent voting rules in the next-best query model.

With n voters and m alternatives, the competitive ratio is trivially upper bounded by $O(m)$ for most reasonable voting rules: the winners can certainly be determined by eliciting complete preference rankings in $n \cdot m$ queries, whereas any rule that reduces to a simple majority in case of only two alternatives—this is the case for almost all prominent rules—requires making at least one query to at least $n/2$ voters. This leads to our main research question:

Which voting rules admit an elicitation algorithm with competitive ratio sublinear in the number of alternatives?

1.1 OUR RESULTS

In Section 3, we warm up with a simple analysis proving that plurality, which selects the alternatives that are the top choice of the maximum number of voters, admits an optimal competitive ratio of 2. This is achieved by the simple rule that elicits all voters' top choices and returns the plurality winners. Then, we extend this analysis to the broader class of k-approval rules, which select the alternatives that are among the top k choices of the maximum number of voters for a given value of k; note that $k = 1$ yields plurality, whereas $k = m - 1$ is known as veto. Once again, simply eliciting the top k choices of all the voters and returning the k-approval winners turns out to be the best one can do with a competitive ratio of k for $k \in \{2, \dots, m-1\}$. This is sublinear when $k \in o(m)$.

In Section 4, we consider Borda count, a classical positional scoring rule in which alternatives are awarded linearly decreasing scores based on their position in voters' preference rankings and alternatives with the highest total scores are selected. We design a natural elicitation algorithm LEVELPRUNING, which elicits the preference rankings of all voters level-by-level and prunes redundant voters along the way. Our main result with an intricate analysis establishes that it achieves a sublinear competitive ratio between $\Omega(\sqrt{m})$ and $O(m^{2/3})$. We also establish a universal lower bound of 2.5 on all elicitation algorithms.

In Section 5, we show that no Condorcet-consistent voting rule admits a sublinear competitive ratio in m, indicating that this family of voting rules may demand more elicitation than the aforementioned positional scoring rules.

In Section 6, we conduct experiments with real datasets from PrefLib [Mattei and Walsh, 2013], which show that our elicitation algorithm achieves even better competitive ratios in practice than the theoretical bounds indicate.

All the missing proofs appear in the appendix.

1.2 RELATED WORK

Competitive analysis. While competitive-ratio analysis is standard in the online algorithms literature [Sleator and Tarjan, 1985, Borodin and El-Yaniv, 2005], its application to active preference elicitation—where the algorithm actively chooses which information to query rather than passively receiving it—is much more recent (see, e.g., Price and Zhou [2023]). Hosseini et al. [2021] introduce the next-best query model and study the competitive ratio for identifying optimal matchings of n agents to n objects. Specifically, they establish a tight bound of $\Theta(\sqrt{n})$ for the criterion of Pareto optimality, while establishing only a lower bound of approximately $4/3$ for a different criterion called rank maximality; Peters [2022] revises the latter to an optimal bound of $3/2$. Our work is the first to adopt this framework to winner determination in voting, where the need to pick a global winner for all n agents rather than matching each of them to a distinct object requires novel algorithmic strategies.

Competitive analysis has also been applied in similar settings. Oren and Lucier [2014] apply it to an online social choice problem, where the preferences arrive in an online fashion. Chen et al. [2018] apply it to analyze algorithms which can extract top K alternatives using noisy pairwise comparisons, with applications to recommender systems.

Partial information. In social choice theory, decision-making under the uncertainty induced by partial preference information (either given or actively elicited) is well-studied. Beyond the winner determination problem discussed above, it has also been applied to design novel voting rules that optimize welfare [Borodin et al., 2022, Kempe, 2020] or fairness [Halpern et al., 2026, Ebadian et al., 2024a].

2 MODEL

For $k \in \mathbb{N}$, define $[k] \triangleq \{1, \dots, k\}$.

Problem instances. Let $V = [n]$ be a set of n *voters* and A be a set of m *alternatives*. Each voter $v \in V$ has a *preference ranking* (strict total order) $\succ_v$ over the alternatives in A. We use $\pi_{\succ_v}(a)$ to denote the rank of alternative a in $\succ_v$, and $\pi^{-1}_{\succ_v}(k)$ to denote the alternative ranked k-th in $\succ_v$. For brevity, after this definition we write π_v instead of $\pi_{\succ_v}$. We call $\vec{\succ} = (\succ_v)_{v \in V}$ the *preference profile*. An instance is given by the tuple $I = (N, A, \vec{\succ})$; we drop I from the notation whenever it is clear from the context. Let $\mathcal{I}$ be the set of all instances over a fixed set of alternatives A (and any finite set of voters V). Effectively, we are fixing the number of alternatives m but allowing any number of voters $n \in \mathbb{N}$.

Voting rules. A *voting rule* $f : \mathcal{I} \to 2^A$ maps each instance to a set of (tied) alternatives, termed *winners*. We study the following prominent voting rules.

- *Positional scoring rules.* A *scoring vector* $\vec{s} = (s_1, \ldots, s_m)$, with $s_1 \geqslant s_2 \geqslant \ldots \geqslant s_m \geqslant 0$ and $s_1 > s_m$, awards a score of s_r to each alternative each time it appears in position $r \in [m]$. The corresponding positional scoring rule selects the set of alternatives with the highest total scores, i.e., $\arg\max_{a \in A} \sum_{v \in V} s_{\pi_v(a)}$.
  - For $k \in [m]$, the *k-approval rule* ($f_{k\text{-approval}}$) is given by the scoring vector $(1, \ldots, 1, 0, \ldots, 0)$ containing k ones followed by zeros, of which *plurality* and *veto* are special cases corresponding to $k = 1$ and $k = m - 1$, respectively.
  - *Borda count* is given by the scoring vector $(m - 1, m - 2, \ldots, 0)$.
  - The *harmonic rule* is given by the scoring vector $(1, 1/2, \ldots, 1/m)$.
- *Condorcet-consistent rules.* Let $n_{a \succ b}(I)$ denote the number of voters who prefer alternative a to b in an instance I. Alternative a is a *Condorcet winner* in instance I if $n_{a \succ b}(I) > n/2$ for all $b \in A \setminus \{a\}$. A voting rule f is *Condorcet-consistent* if, on every instance I in which some alternative a is a Condorcet winner, it returns $f(I) = \{a\}$. Two examples are Copeland's rule and the minimax rule.
  - The Copeland score of an alternative a in I is obtained by adding 1 for every alternative b it wins against by pairwise majority ($n_{a \succ b}(I) > n_{b \succ a}(I)$) and 0.5 for every alternative b it is tied against ($n_{a \succ b}(I) = n_{b \succ a}(I)$). Copeland's rule selects the set of alternatives with the highest Copeland score.
  - *Minimax rule.* The minimax score of an alternative a in instance I is given by $\min_{b \in A} n_{a \succ b}(I) - n_{b \succ a}(I)$. The minimax rule selects the set of alternatives with the highest minimax score.

Sequential queries. A *query sequence* Q is a finite ordered list $((v_t, k_t))_{t \geqslant 1}$, where $v_t \in V$ and $k_t \in [m]$. Let $|Q|$ denote the length of the list, i.e., the total number of queries. For $t \in [|Q|]$, let $Q[t] \triangleq (v_t, k_t)$ be the t-th query, and $Q[1{:}t]$ be the sequence of first t queries. Query (v_t, k_t) elicits the k_t-th alternative in the preference ranking of voter v_t, denoted $R((v_t, k_t), I) \triangleq \pi_{v_t}^{-1}(k_t)$. With slight abuse of notation, let $R(Q[1{:}t], I) \triangleq (R(Q[1], I), \ldots, R(Q[t], I))$ be the responses to the first t queries.

Consistent instances and necessary winners. Given a query sequence Q, let $\mathcal{C}(Q, I)$ denote the set of all *consistent instances* I' that would have yielded the same responses as I, i.e., for which $R(Q, I') = R(Q, I)$. We say that alternative a is a *necessary winner* if $a \in f(I')$ for all $I' \in \mathcal{C}(Q, I)$.

Next-best query model. A *next-best query sequence* Q elicits any voter's preference ranking from top to bottom, i.e., it elicits any voter v's any k-th favorite alternative only after eliciting her top $k - 1$ favorite alternatives: formally, for all $t \in [|Q|]$, $v \in V$, and $k \in [m]$, if $Q[t] = (v, k)$, then for all $k' \in [k - 1]$, there exists $t' \in [t - 1]$ with $Q[t'] = (v, k')$. Let $d(v, Q)$ denote the number of queries posed to v under Q, which is the length of the elicited prefix of her preference ranking.

At any intermediate stage, the elicitation algorithm observes only a partial preference profile: for each voter, a prefix of her ranking has been revealed, while the remaining alternatives are still unelicited. Thus, the algorithm must reason under uncertainty about the unobserved parts of the profile and decide which queries are necessary to certify the winner.

Elicitation in the next-best query model. An *elicitation algorithm* $\mathcal{A}$ for voting rule f maps every instance $I \in \mathcal{I}$ to a query sequence $Q \triangleq \mathcal{A}(I) = ((v_1, k_1), (v_2, k_2), \ldots)$ of some length ℓ satisfying three conditions:

[C1] (Winner Determination) The query responses must determine the set of winners, i.e., in all instances I' with $R(Q, I) = R(Q, I')$, we must have $f(I') = f(I)$.

[C2] (Next-Best) Q must be a next-best query sequence.

[C3] (Online Elicitation) The next query must depend only on the elicited responses, i.e., for any $t \in [\ell - 1]$ and any consistent instance $I' \in \mathcal{C}(Q[1{:}t], I)$, we must have $|\mathcal{A}(I')| \geqslant t + 1$ and $\mathcal{A}(I')[t + 1] = \mathcal{A}(I)[t + 1]$.

Optimal queries and competitive ratio. Let $\mathrm{OPT}(I)$ be any shortest sequence satisfying conditions [C1] and [C2] above, but not necessarily [C3]. Note that in the absence of [C3], $\mathrm{OPT}(I)$ is characterized by the prefix length $d(v, \mathrm{OPT}(I))$ elicited from each voter v. While $|\mathrm{OPT}(I)|$ is the smallest number of queries that reveal the winners under I, an online elicitation algorithm $\mathcal{A}$ oblivious of the instance may end up making more queries before deducing the winners; its *competitive ratio* is given by:

$$\mathrm{CR}(\mathcal{A}) = \sup_{I \in \mathcal{I}} \frac{|\mathcal{A}(I)|}{|\mathrm{OPT}(I)|}.$$

We remark that all our upper bounds are obtained by elicitation algorithms that reveal the set of all (tied) winners, whereas all our lower bounds hold even if one wishes to reveal at least one necessary winner.

3 WARM UP: PLURALITY AND k-APPROVAL

Plurality is arguably the most widely used voting rule. Recall that plurality winners are simply the alternatives that are the top choices of the highest number of voters. Consequently, eliciting the top choices of all the voters (in any order) suffices to reveal them with n queries. *What is the competitive ratio of this algorithm? Is it the best possible?*

The following simple analysis shows that it is essentially 2 and indeed the best possible.

Theorem 1. *The algorithm that elicits the top choices of all the voters and returns the plurality winners has competitive ratio* 2*. No online algorithm for the plurality rule in the next-best query model has competitive ratio less than* 2*.*

Proof. Let us establish an upper bound on the competitive ratio of the algorithm in question and the same lower bound on the competitive ratio of all online algorithms.

Upper bound. Because the algorithm makes n queries on every instance, it is sufficient to establish that $|\operatorname{OPT}(I)| \geqslant n/2$ for all instances I. Suppose for contradiction that there is an instance I in which one can conclude that alternative a is a necessary plurality winner after making fewer than $n/2$ queries. Then, more than $n/2$ voters would have received no queries. If all these voters rank another alternative b as their top choice, it would make b the only plurality winner, contradicting the assumption that a is a necessary plurality winner. This implies that $|\operatorname{OPT}(I)| \geqslant n/2$, so the competitive ratio of the algorithm is at most 2.

Lower bound. Consider any online elicitation algorithm $\mathcal{A}$. Fix any two alternatives $a, b \in A$ and construct an instance adversarially as follows. The first $\lfloor n/2 \rfloor$ voters v receiving a query of $(v, 1)$ (for their top choice) respond with alternative a. The next $\lfloor n/2 \rfloor - 1$ voters v receiving a query of $(v, 1)$ (for their top choice) respond with alternative b. Any voter v receiving a query of (v, k) for $k > 1$ responds with an arbitrary consistent response (i.e., with any alternative that they have not revealed in response to a query (v, k') for any $k' < k$). It is easy to see that this information is not sufficient to deduce the plurality winner as the gap between the plurality scores of a and b is 1 and there are at least two voters who have not revealed their top choices. Thus, the algorithm must make at least $2\lfloor n/2 \rfloor - 1$ queries on all instances consistent with the revealed responses thus far. In contrast, consider the instance I in this family, in which a is the top choice of $\lfloor n/2 \rfloor + 1$ voters; revealing the top choices of these voters would suffice to deduce a as a necessary plurality winner in I. Hence, the competitive ratio of $\mathcal{A}$ is

$$\operatorname{CR}(\mathcal{A}) \geqslant \lim_{n \to \infty} \frac{2\lfloor n/2 \rfloor - 1}{\lfloor n/2 \rfloor + 1} = 2,$$

as desired. □

Recall that plurality is the 1-approval rule. Let us move on to the k-approval rule, for a given $k \in \{2, 3, \ldots, m-1\}$, which selects the alternatives that appear in the top k positions of the maximum number of voters. With careful analysis, the argument for plurality generalizes: simply eliciting the top k positions of all n voters and returning the set of k-approval winners yields an optimal competitive ratio of k. Note that this optimal ratio does not change when moving from $k = 1$ to $k = 2$, but increases linearly afterwards.

Theorem 2. *For $k \in \{2, 3, \ldots, m-1\}$, the algorithm that elicits the top k choices of all the voters and returns the k-approval winners has competitive ratio k, and no online elicitation algorithm for the k-approval rule in the next-best query model has competitive ratio less than k.*

4 BORDA COUNT

Let us now move on to Borda count, a widely used and one of the most classical voting rules. Suddenly, we discover that the analysis gets a lot more complex than Theorem 1.

First, we can show a slightly improved lower bound for Borda count compared to the bound for plurality in Theorem 1, although this requires a significantly more involved construction.

Theorem 3. *The competitive ratio of any online elicitation algorithm for Borda count is at least* 2.5*.*

How do we achieve a low competitive ratio for Borda count? For plurality, recall that we simply elicited the top choices of all voters. This information does not suffice to identify necessary Borda count winners.

What if we elicit the entire preference profile with $n \cdot m$ queries? Then, we face a terrible competitive ratio of m due to the simple instance in which all voters happen to rank the same alternative first and $O(n)$ queries would suffice to reveal it as the unique Borda count winner.

This suggests the following refined approach. Elicit voters' preference rankings *level by level*, i.e., first everyone's top choices,[1] then everyone's second-best choices, and so on, and, crucially, *stop* as soon as the set of Borda winners is identified. This algorithm, which we term LEVEL, works well on the simple instance above, but still incurs the same $\Omega(m)$ competitive ratio.

Proposition 1. *The competitive ratio of* LEVEL *for Borda count is* $\Omega(m)$*.*

Proof. To see this, consider an instance with $n = 2p + 1$ voters, and a set A of m alternatives. Fix two special alternatives $a, b \in A$. Voters $1, \ldots, p$ rank a first, b second, and the remaining alternatives arbitrarily. Voters $p + 1, \ldots, 2p$ rank b first, a second, and the remaining alternatives arbitrarily. And voter $2p + 1$ ranks b *last*, a second-to-last, and the remaining alternatives arbitrarily in the first $m - 2$ positions. Note that a is the unique Borda count winner, but LEVEL will notice a tie between a and b after eliciting two levels in everyone's rankings, and will need to continue eliciting till level $m - 1$ before it sees a in the ranking of voter $2p + 1$.

Hence, LEVEL will terminate after $n(m - 1)$ queries, whereas $2n + (m - 2)$ queries in which all voters reveal

[1] Voters in each level are elicited in an arbitrary order.

their top two choices and voter $2p+1$ further reveals ranks $3, \ldots, m-1$ would suffice to establish a as the unique Borda count winner, implying that the competitive ratio of LEVEL is at least

$$\lim_{n\to\infty} \frac{n(m-1)}{2n+(m-2)} = \frac{m-1}{2} \in \Omega(m). \qquad \square$$

In the bad instance in the proof of Proposition 1, LEVEL performs terribly because after eliciting the first two levels, we can already narrow down to a and b as the only *possible winners* under Borda count. Since voters $1, \ldots, 2p$ have already revealed both a and b, it would make sense to *prune* them and not query them further. Yet, LEVEL queries all voters at each level, which leads to the bad competitive ratio.

Based on this insight, we next consider an algorithm, termed LEVELPRUNING and presented as Algorithm 1, which mimics LEVEL, but after each level, prunes (and makes no further queries to) voters who have revealed all alternatives that can plausibly be Borda count winners in the end. The following result characterizes a condition that allows us to rule out some alternatives that cannot possibly be a winner.

Lemma 1. *Given a next-best query sequence Q, its responses R, and an alternative a, define* $\mathrm{LB}(a; Q, R)$ *and* $\mathrm{UB}(a; Q, R)$ *to be the lowest and highest Borda scores of a, respectively, across all instances which yield responses R to query sequence Q. Then, alternative a is a possible Borda winner under some completion consistent with R only if* $\mathrm{UB}(a; Q, R) \geqslant \max_{b\in A} \mathrm{LB}(b; Q, R)$.

Proof. If there exists $b \in A$ such that $\mathrm{UB}(a; Q, R) < \mathrm{LB}(b; Q, R)$, then a would have a lower Borda score than b in every completion consistent with R. Hence, a cannot be a possible winner under any completion in this case. $\square$

Lemma 1 motivates LEVELPRUNING as follows. After each level, the algorithm computes $P = \{a \in A : \mathsf{UB}(a) \geqslant \max_{b\in A} \mathsf{LB}(b)\}$, a superset of the set of alternatives that can still be Borda count winners, and prunes active voters who have already revealed all alternatives in P. Since lower bounds can only increase and upper bounds can only decrease as more prefixes are revealed, P can only shrink, so a pruned voter has already revealed every alternative that may remain relevant. When no active voters remain, every alternative in P has an exact score (i.e., $\mathrm{UB}(a) = \mathrm{LB}(a)$ for all $a \in P$) and alternatives outside P cannot be possible winners. Further, all alternatives in P must have the same (exact) score and must all be winners because if $a \in P$ has a lower score than $b \in P$, then $\mathrm{UB}(a) < \mathrm{LB}(b)$ would hold, which would have eliminated a from P in the last update, a contradiction. Hence, in the end, the algorithm simply returns P.

The difficult argument, with an intricate proof, is that pruning through the necessary condition in Lemma 1 indeed helps LEVELPRUNING achieve a sublinear competitive ratio of $O(m^{2/3})$; compare this to the $\Theta(m)$ competitive ratio of LEVEL (Proposition 1).

ALGORITHM 1: LEVELPRUNING for Borda Count

Input: Voters N, alternatives A with $|A| = m$
Output: Borda winner(s)
Initialize each voter's revealed prefix to empty
$U \leftarrow N$ `// active voters`
for $d = 1, \ldots, m$ **do**
  Query each active voter $i \in U$ for her d-th most favorite alternative `// this is a next-best query; voters in U have revealed their top d − 1 alternatives in prior iterations`
  foreach $a \in A$ **do**
    $\mathsf{LB}(a) \leftarrow$ minimum possible Borda score of a `// place a last in every unrevealed position`
    $\mathsf{UB}(a) \leftarrow$ maximum possible Borda score of a `// place a as high as possible among unrevealed positions`
  $\mathsf{bestLB} \leftarrow \max_{b\in A} \mathsf{LB}(b)$
  $P \leftarrow \{a \in A : \mathsf{UB}(a) \geqslant \mathsf{bestLB}\}$ `// superset of possible winners`
  Remove every voter from U who has revealed all alternatives in P `// prune inactive voters`
  if $U = \emptyset$ **then** `// no active voters remain`
    return P

Theorem 4. *The competitive ratio of* LEVELPRUNING *for Borda count is $O(m^{2/3})$.*

Proof. Let I be the underlying instance and w be a Borda winner on I with $\mathrm{score}(w) = nm(1-\delta)$, where $\delta \in (0,1)$. Fix $\lambda > 1$, which we will set later, and $d^* = \lambda\delta m$.

Phase 1. Consider the execution of LEVELPRUNING until the end of the loop with $d = d^*$. Let V' be the subset of voters who do not reveal w by the end of Phase 1. Since w always remains a possible winner, Lemma 1 ensures that $w \in P$ throughout the execution of Phase 1. Hence, all voters who have not revealed w by the end of Phase 1 must be active in U (i.e., $V' \subseteq U$). The contribution of each such voter to the Borda score of w is at most $m - d^* = m(1-\lambda\delta)$. Hence, $nm(1-\delta) \leqslant (n - |V'|)\cdot m + |V'|\cdot m\cdot(1-\lambda\delta) = nm - |V'|m\lambda\delta$, so $|V'| \leqslant n/\lambda$. Therefore,

$$\begin{aligned}\mathrm{LB}(w; Q_{d^*}, R_{d^*}) &\geqslant \mathrm{score}(w) - |V'|m(1-\lambda\delta) \\ &\geqslant nm(1-\delta) - \tfrac{n}{\lambda}m(1-\lambda\delta) = nm\Big(1 - \tfrac{1}{\lambda}\Big),\end{aligned}$$

where Q_d and R_d are the queries and responses during Phase 1, respectively. The cost of this phase is at most $\lambda\delta nm$.

Pool size. At any depth d,

$$C^d_{\text{pool}} := \sum_{a \neq w} \max\{0, \text{UB}(a; Q_d, R_d) - \text{LB}(w; Q_d, R_d)\}.$$

At the start of Phase 2, we begin with a pool surplus of $C^{d^*}_{\text{pool}}$. Using the above lower bound and relaxing to all $a \in \mathcal{A}$, $C^{d^*}_{\text{pool}} \leqslant \sum_{i \in \mathcal{I}} \sum_{a \in \mathcal{A}} \max\{0,\ s^{\text{UB}}_{i,a} - m(1 - \frac{1}{\lambda})\}$, where $s^{\text{UB}}_{i,a}$ is the Borda score that a can attain by being placed in the highest possible position in the ranking of voter i. Borda scores follow $s_j = m - j$ for all $j \in [m]$. The surplus over the baseline $m(1 - \frac{1}{\lambda})$ at rank j is $\frac{m}{\lambda} - j$, which is positive only for $j \leqslant m/\lambda$. Since each rank is occupied by one alternative, for each voter i we have $\sum_{a \in \mathcal{A}} \max\{0,\ s^{\text{UB}}_{i,a} - m(1 - \frac{1}{\lambda})\} \leqslant \sum_{j=1}^{\lfloor m/\lambda \rfloor} \left(\frac{m}{\lambda} - j\right) \leqslant \frac{1}{2}\left(\frac{m}{\lambda}\right)^2$. Thus, $C^{d^*}_{\text{pool}} \leqslant \frac{n}{2}\left(\frac{m}{\lambda}\right)^2$.

Phase 2. At the end of depth d elicitation in Phase 2, let $P_d = \{a \in A : \text{UB}(a; Q_d, R_d) \geqslant \max_b \text{LB}(b; Q_d, R_d)\}$ be the set computed by LEVELPRUNING that contains all possible-winners. Let $P_{d,i} \subseteq P_d$ be the alternatives in P_d which voter i has not revealed yet. In the next iteration, suppose the algorithm queries an active voter i to reveal the alternative at rank $d+1$. Since voter i was active, we know that $|P_{d,i}| \geqslant 1$. There are three possibilities.

- Voter i reveals an alternative not in P_d at rank $d+1$. Then, each alternative $a \in P_{d,i}$ incurs a decrease of at least 1 in its Borda upper bound (i.e., $\text{UB}(a; Q_{d+1}, R_{d+1}) \leqslant \text{UB}(a; Q_d, R_d) - 1$) because it is no longer possible to place it at rank $d+1$. Since $|P_{d,i}| \geqslant 1$, this reduces the pool surplus by at least 1 (i.e., $C^{d+1}_{\text{pool}} \leqslant C^d_{\text{pool}} - 1$).
- Voter i reveals an alternative $a \in P_d$ at rank $d+1$ and $|P_{d,i}| \geqslant 2$. The same argument above still applies to all alternatives in $P_{d,i}$ except a. Since $|P_{d,i}| \geqslant 2$, the pool surplus still reduces by at least 1.
- Voter i reveals the unique alternative $a \in P_{d,i}$. Then, the voter is removed from the set of active voters at the end of the loop. Note that each voter can be asked at most one query that falls under this category.

Because $C^{d^*}_{\text{pool}} \geqslant C^d_{\text{pool}} \geqslant 0$, the total number of queries of types 1 and 2 is upper bounded by $C^{d^*}_{\text{pool}}$. And the total number of queries of type 3 is naturally upper bounded by n. Hence, the total cost of Phase 2 is at most $C^{d^*}_{\text{pool}} + n$.

Optimizing the Total Cost. The total cost is therefore at most $\lambda\delta nm + \frac{nm^2}{2\lambda^2} + n$. Balancing the first two terms gives $\lambda^3 = m/\delta$, i.e., $\lambda = (m/\delta)^{1/3}$, and substituting that allows us to upper bound the total cost by $\frac{3}{2}\, nm^{4/3}\delta^{2/3} + n$.

Competitive ratio. Since $\text{OPT} \geqslant nm\delta$, we obtain $\text{CR} \leqslant \frac{\frac{3}{2}\, nm^{4/3}\delta^{2/3} + n}{nm\delta} = \frac{3}{2}\left(\frac{m}{\delta}\right)^{1/3} + \frac{1}{m\delta}$. As $\delta \geqslant 1/m$, this is $O(m^{2/3})$. □

It is unclear if our analysis in Theorem 4 is tight. Although, we are able to establish the following lower bound, showing that LEVELPRUNING cannot achieve a constant competitive ratio matching the universal lower bound of Theorem 3.

Theorem 5. *The competitive ratio of* LEVELPRUNING *for Borda count is* $\Omega(\sqrt{m})$.

This raises a natural open question, which we leave for future work.

> **Open Question:** Does there exist an elicitation rule with constant competitive ratio for Borda count?

5 CONDORCET METHODS

A popular family of voting rules, disjoint from the family of positional scoring rules [Fishburn, 1974], is that of Condorcet-consistent rules, which includes prominent rules such as Copeland's rule, the minimax rule, Kemeny's rule, ranked pairs, Schulze's method, and Dodgson's method. While we showed that many positional scoring rules admit $o(m)$ competitive ratio, the following result shows that all Condorcet-consistent rules admit $\Omega(m)$ competitive ratio, indicating that determining a Condorcet winner can incur significant cost due to the online nature of the elicitation.

Theorem 6. *The competitive ratio of any elicitation algorithm for any Condorcet-consistent voting rule is* $\Omega(m)$.

Proof. Suppose n and m are even. Consider a set of alternatives $A = \{w, c\} \cup X \cup Y$, where $|X| = |Y| = m/2 - 1$. We construct an adversarial class of problem instances using Table 1 as our guide; whenever X or Y appear in the preference ranking, we substitute an arbitrary order among the alternatives in that set instead.

Voter type	No. of voters	Preference ranking
(T1)	$n/2 - 1$ voters:	$w \succ c \succ X \succ Y$
(T2)	2 voters:	$Y \succ c \succ w \succ X$
(T3)	2 voters:	$X \succ c \succ w \succ Y$
(T4)	$n/2 - 5$ voters:	$X \succ Y \succ c \succ w$
(T5)	2 voters:	$X \succ Y \succ w \succ c$

Table 1: Adversarial problem instances for Condorcet-consistent rules.

The adversary acts as follows. For the first $n/2 - 1$ voters receiving their first query, the adversary assigns type (T1) as soon as the first query is received and responds to all their queries as per the table. The same holds for the next 2 voters receiving their first query, who are assigned type (T2), and the subsequent 2 voters receiving their first query, who are assigned type (T3). For all the remaining voters,

who may be of type (T4) or (T5), the adversary continues revealing $X \succ Y$ for the first $m-2$ positional queries without determining the type. For the first $n/2-5$ of these voters receiving a query at position $m-1$, the adversary assigns type (T4) and reveals alternative c (and responds to a query at position m, if ever posed, with alternative w). The last 2 voters in this group receiving a query at position $m-1$ are assigned type (T5) and their queries at positions $m-1$ and m are responded to accordingly.

First, notice that w is the Condorcet winner: the majority (T1)+(T2) prefer $w \succ X$, the majority (T1)+(T3) prefer $w \succ Y$, and the majority (T1)+(T5) prefer $w \succ c$.

Next, we argue that any elicitation algorithm for any Condorcet-consistent rule must make at least $(n/2-4)\cdot(m-1)$ queries before terminating (with w as the necessary winner). Suppose for contradiction that it terminates with fewer queries. Consider the group of (T4)+(T5) voters. Because the adversary only assigns type (T5) to the $(n/2-4)^{\text{th}}$ and $(n/2-3)^{\text{rd}}$ voter in this group receiving a query at position $m-1$, the adversary must not have assigned type (T5) to any voters yet. Then, the algorithm cannot rule out the possibility that all voters in the group (T4)+(T5) have preference ranking $X \succ Y \succ c \succ w$. For voters of types (T4) and (T5), the relative order between c and w remains unrevealed at this point, so the adversary can complete their rankings with $c \succ w$ without contradicting any answer given so far. However, in this possibility, c becomes the Condorcet winner: the majorities (T1)+(T2) and (T1)+(T3) prefer $c \succ X$ and $c \succ Y$, respectively, just like w, but now the majority $(T2)+(T3)+(T4)+(T5)$ would also prefer $c \succ w$. But then, the algorithm cannot possibly terminate with w as the necessary winner under any Condorcet-consistent voting rule, a contradiction.

Let us contrast this with the optimal number of queries. Note that OPT can reveal the Condorcet winner w by querying (T1) voters up to rank 1, (T2) and (T3), and (T5) voters up to rank $m-1$. This amounts to $O(n+m)$ queries.

Taking the ratio of the $\Omega(nm)$ lower bound on the algorithm and $O(n+m)$ upper bound on OPT, followed by the limit of $n \to \infty$, yields the desired $\Omega(m)$ lower bound. □

6 EXPERIMENTS

In this section, we evaluate the empirical performance of our proposed query strategies across various positional scoring rules and tournament semantics. The implementation and experiment scripts are available in the NextBest repository.

6.1 EXPERIMENTAL SETUP

Datasets. We conduct our experiments using real-world preference datasets from **PrefLib** [Mattei and Walsh, 2013] under the Strictly Ordered Complete (SOC) category. Figure 1 depicts the distribution of the number of voters n and the number of alternatives m in these datasets collectively.

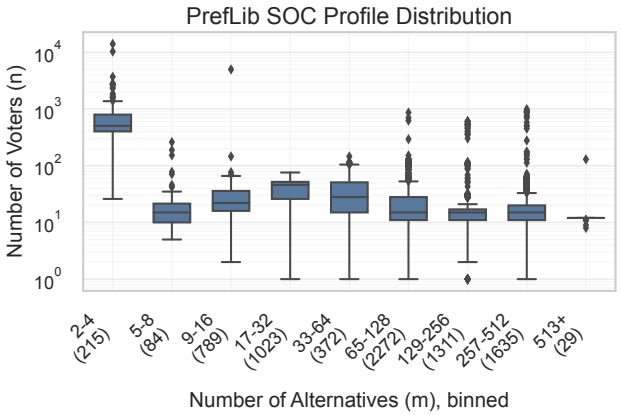

Figure 1: Distribution of PrefLib SOC profiles used in our experiments. Profiles are grouped by the number of alternatives (m), and each boxplot shows the distribution of the number of voters (n) within that group. Parenthesized values on the x-axis give the number of profiles in each bin.

Voting Rules. We consider a diverse and representative set of prominent voting rules:

- Positional scoring rules (PSRs): **Borda**, **Harmonic**, and three instantiations of k**-Approval**, namely **Plurality** ($k=1$), **Half-Approval** ($k=\lfloor m/2 \rfloor$), and **Veto** ($k=m-1$), and
- Condorcet-consistent rules: **Copeland** and **Minimax**.

Baselines for Comparison. For these voting rules, we compare the performance of the following three approaches:

- **OPT (Optimal):** The absolute minimum number of queries required to prove the winner. This is computed via an Integer Linear Programming (ILP) formulation provided in Appendix C using the Gurobi solver.
- **LevelPruning:** Our proposed algorithm (Algorithm 1) that queries voters level-by-level while pruning those who have revealed all possible-winners.
- **Level:** A standard baseline that queries voters level-by-level, without any pruning, until one candidate emerges as the necessary winner. Note that the maximum depth $d^{\max}$ queried under Level and LevelPruning is identical, but LevelPruning stops querying some voters at depth lower than $d^{\max}$.

To ensure a reasonable timeframe, we impose a 60-second time limit on the ILP for OPT; for each rule, this excluded less than 1% of the instances, leaving at least 7,000 instances on which OPT (and hence, the competitive ratios of Level and LevelPruning) could be computed.

Evaluation Metrics. We assess the algorithms based on three primary metrics:

- **Competitive Ratio:** The ratio of the queries used by LEVELPRUNING or LEVEL to the optimal number of queries OPT. This sheds light on the overall performance of the two algorithms.
- **Pearson Correlation:** The Pearson correlation coefficient between the depth to which voter i is queried under LEVELPRUNING or OPT and the position of the true winner in voter i's ranking. This sheds light on the probing strategy of an algorithm. Ideally, an algorithm should query each voter approximately up to where she can reveal the true winner, resulting in a high correlation coefficient.

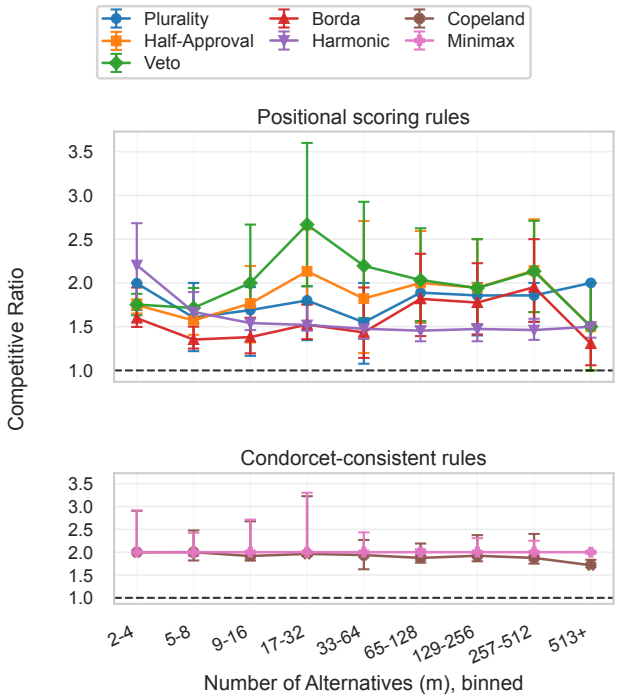

Figure 2: Scalability of the competitive ratio (LEVELPRUNING relative to the exact OPT solver) as a function of the number of alternatives (m), with m grouped into logarithmic-size bins. Points show the median ratio within each bin and error bars show the interquartile range.

6.2 RESULTS AND ANALYSIS

Performance of LEVELPRUNING. Figure 2 shows that, on average, the competitive ratio of LEVELPRUNING remains rather small (median typically close to 2) and does not exhibit any clear upward trend as the number of alternatives grows. Its performance is fairly consistent across Condorcet-consistent rules, slightly better on Borda, and slightly worse on Veto. The empirical comparison between Borda and Veto is consistent with their worst-case analyses.

Does Pruning Help? Figure 3 compares the competitive ratios of LEVELPRUNING and LEVEL. For k-approval rules (Plurality, Half-Approval, and Veto), they are identical for a simple reason: until all voters are queried for their top k positions (or a unanimous winner with a score of n is observed), no alternative (and, thus, no voter) can be pruned as such an alternative can possibly appear in the k-th position of all voters who have not revealed it yet, making it a possible winner. Pruning provides modest savings for Harmonic, Copeland, and Minimax, and noticeably more for Borda.

Probing Strategy. Figure 4 depicts the correlation between ranks to which voters are queried in LEVELPRUNING and OPT and ranks at which they place the true winner. LEVELPRUNING maintains a near-zero correlation because it largely queries level-by-level, querying most voters to the same maximum depth $d^{\max}$; indeed, for k-approval rules, we observed above that it never prunes any voters, resulting in a Pearson correlation coefficient of exactly 0 on every instance. The probing strategy of OPT, on the other hand, is highly complex, and shows a tension between two competing tendencies:

- **Negative correlation:** For Condorcet-consistent rules such as Copeland and Minimax, the negative correlation comes largely from the fact that OPT chooses to make no queries to many detractors who rank the winner low; instead, it verifies the winner by making a few queries to its supporters, thereby uncovering it at high ranks and deducing that it defeats the large number of alternatives placed below it in those rankings.
- **Positive correlation with peaks at 1 and 0:** For k-approval rules with a large k, OPT has a dichotomous behavior. When a voter ranks the winner high (thus in the top k), OPT makes just as many queries as required to uncover the winner, confirming that the winner gets a score of 1 from that voter in the k-approval tally; the number of queries being equal to the rank of the winner creates a peak at 1. In contrast, when a voter ranks the winner low (but still within the top k), OPT queries the voter till the k-th position regardless of the rank of the winner as making a few extra queries after uncovering the winner is worth it to also establish that all alternatives who do not appear in the top k do not get a score of 1 from that voter in the k-approval tally; the number of queries being exactly k regardless of the rank of the winner creates a peak around 0. This pattern shows up for rules such as Half-Approval and Veto.

Plurality is special: voters who place the winner at rank 1 are queried till rank 1, and other voters who place the winner at rank greater than 1 are queried till rank either

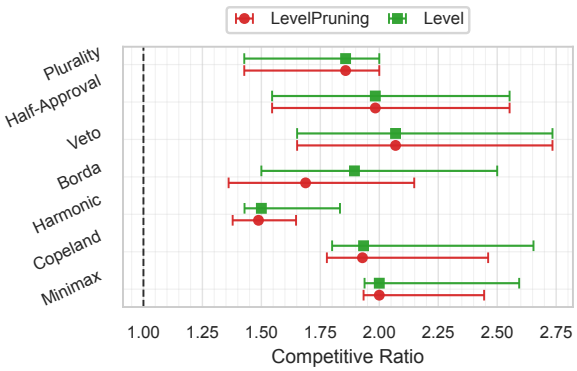

Figure 3: Performance comparison between the proposed LEVELPRUNING and the LEVEL baseline. Points show the median competitive ratio across PrefLib instances and horizontal bars show the interquartile range.

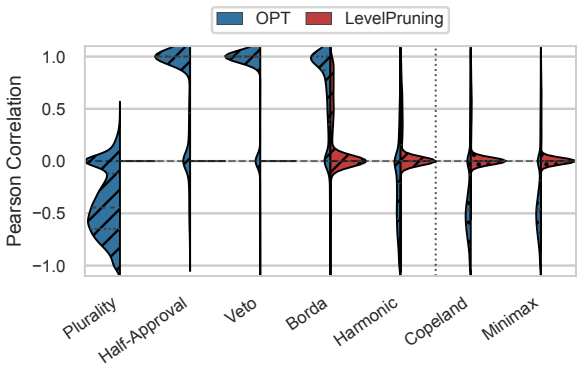

Figure 4: Violin plots illustrating the Pearson correlation between a voter's true rank of the winning candidate and their required query depth under both OPT and LEVELPRUNING paradigms. The vertical dashed line separates positional scoring rules on the left from Condorcet-based rules on the right. The distinct distributions reveal the underlying algorithmic strategies for each voting rule.

0 or 1, resulting in the largely negative correlation. Borda and Harmonic lie in-between with Borda acting similarly to k-approval with a larger k while Harmonic, with its sharply dropping scores, acts more like k-approval with a smaller k.

7 DISCUSSION

Our work shows that active elicitation of preferences coupled with the use of competitive ratio to measure performance has remained severely underexplored in social choice theory—a paradigm whose relevance continues to grow with online platforms deciding between a large number of alternatives by actively eliciting stakeholder preferences.

The optimal competitive ratio for Borda. An immediate direction for future work is to settle the optimal competitive ratio for Borda count. Our results show that LEVELPRUNING achieves an $O(m^{2/3})$ competitive ratio and that the same algorithm cannot achieve better than $\Omega(\sqrt{m})$ in the worst case, while the best lower bound against arbitrary elicitation algorithms is only constant. It remains open whether Borda admits a constant-competitive elicitation algorithm, or whether every algorithm must have a competitive ratio that grows with m. Extending the theoretical analysis to other scoring rules of interest, such as the harmonic rule, would also be interesting.

Alternative query models. Our guarantees are specific to the next-best query model, where voters gradually extend a revealed prefix of their preference rankings. Other preference-elicitation interfaces are also common. For example, platforms such as Polis and Remesh often collect approval preferences or pairwise comparisons. Optimizing the elicitation of such preferences via competitive ratio analysis is an interesting future direction.

Extension to other models. More broadly, competitive analysis may be useful beyond winner determination under popular rules. For example, there is a rapidly expanding literature on distortion, which studies how limited preference information affects welfare approximation in voting [Anshelevich et al., 2021]. In particular, communication-distortion tradeoffs ask how many bits or queries suffice in the worst case over all instances to achieve a desired approximation guarantee [Mandal et al., 2019, 2020, Kempe, 2020] whereas our competitive-ratio perspective asks how many queries are needed in a given instance, relative to the instance-specific optimum, to achieve a given target. In this paper, the target was winner determination under popular voting rules, but future work can replace this by a desired distortion guarantee or a desired level of fairness, e.g., proportional fairness [Ebadian et al., 2024b] or justified representation [Lackner and Skowron, 2023].

Acknowledgements

This work was supported by an NSERC Discovery Grant and an NSERC-CSE Research Communities Grant. Researchers funded through the NSERC-CSE Research Communities Grants do not represent the Communications Security Establishment Canada or the Government of Canada. Any research, opinions or positions they produce as part of this initiative do not represent the official views of the Government of Canada.

References

Elliot Anshelevich, Aris Filos-Ratsikas, Nisarg Shah, and Alexandros A Voudouris. Distortion in social choice problems: The first 15 years and beyond. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4294–4301, 2021.

Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 2005.

Allan Borodin, Daniel Halpern, Mohamad Latifian, and Nisarg Shah. Distortion in voting with top-t preferences. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 116–122, 2022.

F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

Xi Chen, Sivakanth Gopi, Jieming Mao, and Jon Schneider. Optimal instance adaptive algorithm for the top-k ranking problem. *IEEE Transactions on Information Theory*, 64 (9):6139–6160, 2018.

Vincent Conitzer. Eliciting single-peaked preferences using comparison queries. In *Proceedings of the 6th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1–8, 2007.

Vincent Conitzer and Tuomas Sandholm. Communication complexity of common voting rules. In *Proceedings of the 6th ACM Conference on Economics and Computation (EC)*, pages 78–87, 2005.

Soroush Ebadian, Daniel Halpern, and Evi Micha. Metric distortion with elicited pairwise comparisons. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2791–2798, 2024a.

Soroush Ebadian, Anson Kahng, Dominik Peters, and Nisarg Shah. Optimized distortion and proportional fairness in voting. *ACM Transactions on Economics and Computation*, 12(1):1–39, 2024b.

P. C. Fishburn. Paradoxes of voting. *American Political Science Review*, 68(2):537–546, 1974.

Daniel Halpern, Gregory Kehne, Ariel D. Procaccia, Jamie Tucker-Foltz, and Manuel Wüthrich. Representation with incomplete votes. *Theory and Decision*, pages 257–296, 2026.

Hadi Hosseini, Vijay Menon, Nisarg Shah, and Sujoy Sikdar. Necessarily optimal one-sided matchings. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5481–5488, 2021.

David Kempe. Communication, distortion, and randomness in metric voting. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2087–2094, 2020.

Kathrin Konczak and Jérôme Lang. Voting procedures with incomplete preferences. Presented at the IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling, 2005. Unpublished.

A. Konya, L. Schirch, C. Irwin, and A. Ovadya. Democratic policy development using collective dialogues and AI. arXiv:2311.02242, 2023.

Andrew Konya, Luke Thorburn, Wasim Almasri, Oded Adomi Leshem, Ariel D. Procaccia, Lisa Schirch, and Michiel A. Bakker. Using collective dialogues and AI to find common ground between Israeli and Palestinian peacebuilders. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 312–333, 2025.

Martin Lackner and Piotr Skowron. *Multi-Winner Voting with Approval Preferences*. Springer, 2023.

D. Mandal, A. D. Procaccia, N. Shah, and D. P. Woodruff. Efficient and thrifty voting by any means necessary. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 7178–7189, 2019.

Debmalya Mandal, Nisarg Shah, and David P Woodruff. Optimal communication-distortion tradeoff in voting. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 795–813, 2020.

A. Mao, A. D. Procaccia, and Y. Chen. Better human computation through principled voting. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1142–1148, 2013.

N. Mattei and T. Walsh. PrefLib: A library for preferences. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT)*, pages 259—270, 2013.

Joel Oren and Brendan Lucier. Online (budgeted) social choice. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1456–1462, 2014.

David M Pennock, Pedrito Maynard-Reid, C Lee Giles, and Eric Horvitz. A normative examination of ensemble learning algorithms. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 735–742, 2000.

Jannik Peters. Online elicitation of necessarily optimal matchings. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, pages 5164–5172, 2022.

Eric Price and Yihan Zhou. A competitive algorithm for agnostic active learning. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 58670–58691, 2023.

Travis C Service and Julie A Adams. Communication complexity of approximating voting rules. In *Proceedings of the 11th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 593–602, 2012.

Daniel D Sleator and Robert E Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.

C. Small, M. Bjorkegren, T. Erkkilä, L. Shaw, and C. Megill. Polis: Scaling deliberation by mapping high dimensional opinion spaces. *Revista De Pensament I Anàlisi*, 26(2), 2021.

Michael Henry Tessler, Michiel A. Bakker, Daniel Jarrett, Hannah Sheahan, Martin J. Chadwick, Raphael Koster, Georgina Evans, Lucy Campbell-Gillingham, Tantum Collins, David C. Parkes, Matthew Botvinick, and Christopher Summerfield. AI can help humans find common ground in democratic deliberation. *Science*, 386 (6719):eadq2852, 2024.

Lirong Xia and Vincent Conitzer. Determining possible and necessary winners given partial orders. *Journal of Artificial Intelligence Research (JAIR)*, 41:25–67, 2011.

Xiutian Zhao, Ke Wang, and Wei Peng. An electoral approach to diversify llm-based multi-agent collective decision-making. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2712–2727, 2024.

APPENDIX

A MISSING PROOFS FROM SECTION 3

Theorem 2. *For $k \in \{2, 3, \ldots, m-1\}$, the algorithm that elicits the top k choices of all the voters and returns the k-approval winners has competitive ratio k, and no online elicitation algorithm for the k-approval rule in the next-best query model has competitive ratio less than k.*

Proof. Fix k and m, and let $n \gg m$. Let $\mathcal{A} = \{a, b\} \cup \{c_1, \ldots, c_{m-2}\}$.

Profile. Voter v_n ranks c_1 first; all other $n-1$ voters rank a first. Ranks $2, \ldots, k-1$ contain a cyclic permutation of $\{c_1, \ldots, c_{m-2}\}$ across voters. At rank k, every voter ranks b except a single voter v_1, whose k-th ranked alternative is not b.

Under k-approval, $\mathrm{score}(a) = \mathrm{score}(b) = n-1$, so a and b tie. If v_1 also ranked b in position k, then $\mathrm{score}(b) = n$ and b would be the unique winner. Thus identifying v_1 is necessary to certify that b does not win.

Lower bound for any online algorithm. Consider an adversary that, whenever a k-th position is queried, returns b whenever consistent. Since $n-1$ voters truly rank b at position k, the adversary can answer b for the first $n-1$ distinct voters queried at depth k. Until all but one voter have been queried at depth k, there remains a completion in which every queried voter ranks b in the top k, and b is the unique winner. Hence any algorithm must query the k-th position of at least $n-1$ voters. As each such query requires revealing k positions: $\mathrm{cost}_{\mathcal{A}} \geqslant (n-1)k$.

Upper bound on OPT**.** An optimal (profile-aware) scheme proceeds as follows.

- Query the top position of the $n-1$ voters who rank a first, establishing $\mathrm{LB}(a) = n-1$.
- Fully reveal voter v_1 (cost $k-1$ additional queries), establishing that b is absent from one top-k list and thus $\mathrm{UB}(b) \leqslant n-1$.
- Fully reveal a constant number $\lambda \leqslant m-2$ of additional voters to ensure $\mathrm{UB}(c_i) \leqslant n-1$ for all i.

The total cost is

$$\mathrm{OPT} \leqslant (n-1) + \lambda(k-1).$$

Competitive ratio. For the above profile,

$$\mathrm{CR} \geqslant \frac{(n-1)k}{(n-1)+\lambda(k-1)} = \frac{k}{1+\frac{\lambda(k-1)}{n-1}} \xrightarrow[n\to\infty]{} k.$$

Thus any online elicitation algorithm for k-approval has competitive ratio at least k. □

B MISSING PROOFS FROM SECTION 4

Theorem 3. *The competitive ratio of any online elicitation algorithm for Borda count is at least* 2.5.

Proof. We prove the lower bound already for $m = 3$ with Borda scores $(2, 1, 0)$.

Profile family. Let n be a multiple of 3 and consider the base profile P with

$$\underbrace{a \succ b \succ c}_{\frac{2n}{3}-1 \text{ voters}}, \qquad \underbrace{b \succ c \succ a}_{\frac{n}{3} \text{ voters}}, \qquad \underbrace{a \succ c \succ b}_{1 \text{ voter}}.$$

A direct calculation gives $\mathrm{score}(a) = \frac{4n}{3}$ and $\mathrm{score}(b) = \frac{4n}{3} - 1$, so a is the unique Borda winner.

Let $\mathcal{F}$ be the family obtained by permuting voters arbitrarily; i.e., every profile in $\mathcal{F}$ has the same multiset of rankings as P.

Upper bound on OPT**.** For any $P' \in \mathcal{F}$, there is a query sequence of cost at most $\frac{2n}{3} + 2$ that certifies a as the unique winner: query depth 1 for all but one $a \succ b \succ c$ voter, query depth 2 for that voter and for the single $a \succ c \succ b$ voter, and do not query the $b \succ c \succ a$ voters. Then

$$\mathrm{LB}(a) = \frac{4n}{3}, \qquad \mathrm{UB}(b) \leqslant \frac{4n}{3} - 1,$$

so $\mathrm{OPT}(P') \leqslant \frac{2n}{3} + 2$ for all $P' \in \mathcal{F}$.

Lower bound for any online algorithm. Fix any deterministic online algorithm $\mathcal{A}$. An adversary answers queries so as to keep two completions in $\mathcal{F}$ consistent until $\mathcal{A}$ spends at least $\frac{5n}{3}$ queries.

First-position queries: for the first $\frac{n}{3}$ distinct voters queried at depth 1, answer b; for all remaining voters queried at depth 1, answer a. Unless $\mathcal{A}$ queries the first position of *every* voter, there remains a completion in which all unqueried voters are of type $b \succ c \succ a$, making b the winner. Hence $\mathcal{A}$ must use at least n queries at depth 1.

Second-position queries among voters revealed to rank a first: for the first $\frac{2n}{3} - 1$ such queries, answer b. Unless $\mathcal{A}$ observes a c in second position among these voters, there remains a completion in which all such voters are $a \succ b \succ c$, yielding $\mathrm{score}(a) = \mathrm{score}(b)$. Thus $\mathcal{A}$ must issue at least $\frac{2n}{3}$ such second-position queries.

Consequently, for some $P' \in \mathcal{F}$ consistent with the adversary,

$$\mathrm{cost}_{\mathcal{A}}(P') \geqslant n + \frac{2n}{3} = \frac{5n}{3}.$$

Competitive ratio. For this P',

$$\mathrm{CR}(\mathcal{A}) \;\geqslant\; \frac{\frac{5n}{3}}{\frac{2n}{3} + 2} \;\xrightarrow[n\to\infty]{}\; \frac{5}{2}.$$

Hence any online elicitation algorithm for Borda count has competitive ratio at least 2.5. □

Theorem 5. *The competitive ratio of* LevelPruning *for Borda count is* $\Omega(\sqrt{m})$.

Proof. Let m be a perfect square and n be divisible by $2\sqrt{m} - 1$. Partition the voters into $n_1 = \frac{2\sqrt{m}-2}{2\sqrt{m}-1} n$ and $n_2 = \frac{n}{2\sqrt{m}-1}$ voters.

Profile.

- w is ranked 1 by the n_1 voters and $2\sqrt{m}$ by the n_2 voters.
- Let $k = \sqrt{m}/2$. The challengers $c_1, \ldots, c_k$ occupy ranks $2, \ldots, 1 + \sqrt{m}/2$ at every voter, distributed cyclically so that each challenger appears equally often at each such rank.
- All other alternatives fill lower ranks.

True scores.

$$\mathrm{score}(w) = n_1(m-1) + n_2(m - 2\sqrt{m}) = n(m-2).$$

Each challenger has average rank $\frac{2+(1+\sqrt{m}/2)}{2} = 1.5 + \sqrt{m}/4$, hence

$$\mathrm{score}(c_i) = n\left(m - 1.5 - \frac{\sqrt{m}}{4}\right).$$

Thus $\mathrm{score}(w) - \mathrm{score}(c_i) = \Theta(n\sqrt{m})$, so w is the unique winner.

Bounds at depth $d = \sqrt{m}/2$**.** Query the top d positions of every voter. Since $2\sqrt{m} > d$, w is unrevealed in all n_2 voters, and

$$\mathrm{LB}(w; Q_d) = n_1(m-1) = \frac{2\sqrt{m}-2}{2\sqrt{m}-1} n(m-1) = nm - \Theta(n\sqrt{m}).$$

All challengers lie within the queried prefix, so their upper bounds equal their true scores:

$$\mathrm{UB}(c_i; Q_d) = n\left(m - 1.5 - \frac{\sqrt{m}}{4}\right) = nm - \Theta(n\sqrt{m}).$$

No pruning. For sufficiently large m,

$$\mathrm{UB}(c_i; Q_d) - \mathrm{LB}(w; Q_d) = \Theta(n\sqrt{m}) > 0,$$

so none of the $k = \Theta(\sqrt{m})$ challengers is eliminated. Hence after $d = \Theta(\sqrt{m})$ levels, the potential-winner set still has size $\Theta(\sqrt{m})$.

Since each further level requires $\Theta(n)$ additional queries before any challenger can be pruned, LEVELPRUNING must spend $\Omega(n\sqrt{m})$ queries on this instance, while $\mathrm{OPT} = \Theta(n)$. Therefore the competitive ratio is $\Omega(\sqrt{m})$. □

C ILP FORMULATIONS FOR COMPUTING OPT

This appendix gives the integer linear programs used to compute the exact offline optimum in our experiments. These formulations are implemented in `next_best_query_tied.py`. Throughout, let $\mathcal{I}$ be the set of voters, $\mathcal{A}$ the set of alternatives, $n = |\mathcal{I}|$, and $m = |\mathcal{A}|$. For voter i, let $\pi_i(c) \in \{1, \ldots, m\}$ denote the true rank of alternative c, with rank 1 being most preferred. The ILPs use binary variables $z_{i,k}$, where $z_{i,k} = 1$ means that voter i is queried to depth at least k. The telescoping constraints

$$z_{i,k} \geqslant z_{i,k+1} \qquad \forall i \in \mathcal{I},\ k = 1, \ldots, m-1$$

ensure that queries reveal prefixes. In all formulations the objective is

$$\min \sum_{i \in \mathcal{I}} \sum_{k=1}^{m} z_{i,k}.$$

If the true voting rule has multiple winners, the ILP is allowed to certify any one of the tied winners; this matches the experimental definition of the minimum number of queries needed to prove a correct winner.

C.1 POSITIONAL SCORING RULES

This formulation is used for Borda, Harmonic, Plurality, Half-Approval, and Veto by changing only the scoring vector $\alpha = (\alpha_1, \ldots, \alpha_m)$, where $\alpha_1 \geqslant \cdots \geqslant \alpha_m$. Let $\mathcal{W} \subseteq \mathcal{A}$ be the set of true winners. For every $a \in \mathcal{W}$, introduce a binary variable w_a indicating which tied winner is certified, and impose

$$\sum_{a \in \mathcal{W}} w_a = 1.$$

Here w_a does not indicate whether a is the unique true winner; all alternatives in $\mathcal{W}$ are true winners by definition. Instead, $w_a = 1$ means that the ILP chooses a as the single winner to certify. Hence $\sum_{a \in \mathcal{W}} w_a = 1$ makes the formulation choose exactly one certification target among the true tied winners. For each alternative c, define lower and upper score bounds:

$$L_c = \sum_{i \in \mathcal{I}} \left[\alpha_m + \left(\alpha_{\pi_i(c)} - \alpha_m \right) z_{i,\pi_i(c)} \right], \tag{1}$$

$$U_c = \sum_{i \in \mathcal{I}} \left[\alpha_1 - \sum_{k=1}^{\pi_i(c)-1} \left(\alpha_k - \alpha_{k+1} \right) z_{i,k} \right]. \tag{2}$$

The lower bound gives candidate c its true score from voter i only if c has been revealed; otherwise it assigns the worst possible score α_m. The upper bound starts from the best possible score α_1 and decreases whenever revealed alternatives above c rule out higher positions for c. The winner-certification constraints are

$$w_a = 1 \quad \Longrightarrow \quad L_a \geqslant U_b \qquad \forall a \in \mathcal{W},\ \forall b \in \mathcal{A} \setminus \{a\}.$$

Thus, if a is selected as the certified winner, even its worst-case score under the revealed prefixes is at least every other alternative's best-case score. Thus, the ILP certifies one selected member of the true winner set, rather than necessarily certifying the entire tied winner set.

C.2 COPELAND

For Copeland, the implementation solves one ILP for each tied true winner $a \in \mathcal{W}$ and returns the minimum-cost solution among them. Fix such a target winner a. For two alternatives x, y, let

$$P_{xy} = \{i \in \mathcal{I} : \pi_i(x) < \pi_i(y)\}$$

be the voters who truly prefer x to y, and define the number of revealed witnesses for $x \succ y$ as

$$V_{xy} = \sum_{i \in P_{xy}} z_{i,\pi_i(x)}.$$

Let $\tau_+ = \lfloor n/2 \rfloor + 1$ be the strict-majority threshold. If n is even, let $\tau_0 = n/2$ be the tie threshold; if n is odd, set $\tau_0 = \tau_+$. For each ordered pair (x, y) used by the formulation, introduce binary variables p_{xy} and t_{xy} with

$$V_{xy} \geqslant \tau_+ p_{xy}, \qquad V_{xy} \geqslant \tau_0 t_{xy}.$$

When n is odd, the code sets $t_{xy} = p_{xy}$. The certified pairwise Copeland contribution of x against y is

$$s_{xy} = p_{xy} + t_{xy} - 1.$$

Hence $s_{xy} = 1$ certifies a pairwise win for x, $s_{xy} = 0$ certifies a pairwise tie when ties are possible, and $s_{xy} = -1$ leaves open the possibility that y beats x. The constraints defining p_{xy} and t_{xy} are one-directional: the ILP may set these variables to 1 only when enough witnesses have been revealed, but it is not forced to set them to 1 whenever this is possible. This relaxation is sound for certification, because increasing s_{xy} can only strengthen the certificate. It improves the lower bound on the selected winner's Copeland score, or decreases the upper bound on a challenger's Copeland score, and hence can never make an invalid certificate feasible.

The target winner's lower Copeland score is

$$L_a^{\mathrm{Cop}} = \sum_{x \in \mathcal{A} \setminus \{a\}} s_{ax}.$$

For any challenger c, an upper bound on c's Copeland score is

$$U_c^{\mathrm{Cop}} = \sum_{y \in \mathcal{A} \setminus \{c\}} -s_{yc}.$$

The certification constraints are

$$L_a^{\mathrm{Cop}} \geqslant U_c^{\mathrm{Cop}} \qquad \forall c \in \mathcal{A} \setminus \{a\}.$$

These constraints require the revealed prefixes to certify that the target winner's worst-case Copeland score is at least every challenger's best-case Copeland score.

C.3 MINIMAX

Equivalently, since preferences are strict and complete, maximizing

$$\min_{x \neq c} (n_{c \succ x} - n_{x \succ c})$$

is the same as minimizing

$$\max_{x \neq c} n_{x \succ c}.$$

We use the latter defeat-count formulation in the ILP, so lower values are better. The implementation uses a single ILP over all tied true winners. Let $\mathcal{W}$ be the true Minimax winner set. For each $a \in \mathcal{W}$, the variable $w_a \in \{0, 1\}$ indicates whether a is selected as the Minimax winner to certify. The constraint

$$\sum_{a \in \mathcal{W}} w_a = 1.$$

ensures that exactly one true Minimax winner is selected as the certification target. For each ordered pair (x, c) with $x \neq c$, define

$$P_{xc} = \{i \in \mathcal{I} : \pi_i(x) < \pi_i(c)\}.$$

The lower bound on the number of voters certified to prefer x over c is

$$L_{xc} = \sum_{i \in P_{xc}} z_{i,\pi_i(x)}.$$

The lower bound L_{xc} counts voters whose revealed prefixes already certify $x \succ_i c$. For a voter with $x \succ_i c$, revealing x is enough to certify this comparison, since all unrevealed alternatives must lie below the revealed prefix. The upper bound on the number of voters who may prefer x over c is

$$U_{xc} = |P_{xc}| + \sum_{i \in P_{cx}} \big(1 - z_{i,\pi_i(c)}\big).$$

The upper bound U_{xc} counts the largest possible number of voters who could still prefer x to c under some completion consistent with the revealed prefixes. All voters in P_{xc} can contribute to this upper bound. For voters in P_{cx}, the true order is $c \succ_i x$; such a voter can be ruled out as supporting $x \succ_i c$ only once c has been revealed, which explains the term $1 - z_{i,\pi_i(c)}$.

For each candidate c, introduce an integer variable M_c^U satisfying

$$M_c^U \geqslant U_{xc} \qquad \forall x \in \mathcal{A} \setminus \{c\}.$$

The variable M_c^U upper-bounds c's maximum pairwise defeat over all consistent completions. Since lower defeat-count scores are better under the Minimax formulation used here, this is the quantity we need to upper-bound for the selected winner.

To compare a selected target winner against a challenger b, the ILP also chooses one opponent witnessing the challenger's lower-bound maximum defeat. Introduce binary variables $q_{b,y}$ for $y \neq b$ and impose

$$\sum_{y \in \mathcal{A} \setminus \{b\}} q_{b,y} = 1.$$

The variables $q_{b,y}$ select, for each challenger b, one opponent y that witnesses a lower bound on b's maximum defeat. Since b's Minimax defeat is a maximum over opponents, one such witness is sufficient.

With $M_{\text{big}} = n + 1$, the certification constraints are

$$M_a^U - L_{yb} \leqslant M_{\text{big}}\big(2 - w_a - q_{b,y}\big)$$

for all $a \in \mathcal{W}$, $b \in \mathcal{A} \setminus \{a\}$, and $y \in \mathcal{A} \setminus \{b\}$. When $w_a = 1$ and $q_{b,y} = 1$, the big-M constraint reduces to

$$M_a^U \leqslant L_{yb}.$$

Thus, even the worst-case maximum defeat of the selected winner a is no larger than a certified lower bound on challenger b's maximum defeat. Therefore b cannot have a strictly better Minimax defeat-count score than a in any completion consistent with the revealed prefixes.

In short, the ILP certifies a selected true Minimax winner a by upper-bounding a's maximum defeat and, for every challenger b, finding one certified pairwise defeat of b that is at least as large.