Representation with Incomplete Votes

Daniel Halpern, Gregory Kehne, Ariel D. Procaccia, Jamie Tucker-Foltz and **Manuel Wüthrich**

Harvard University

Abstract

Platforms for online civic participation rely heavily on methods for condensing thousands of comments into a relevant handful based on whether participants agree or disagree with them. We argue that these methods should guarantee fair representation of the participants, as their outcomes may affect the health of the conversation and inform impactful downstream decisions. To that end, we draw on the literature on approval-based committee elections. Our setting is novel in that the approval votes are incomplete since participants will typically not vote on all comments. We prove that this complication renders non-adaptive algorithms impractical in terms of the amount of information they must gather. Therefore, we develop an adaptive algorithm that uses information more efficiently by presenting incoming participants with statements that appear promising based on votes by previous participants. We prove that this method satisfies commonly used notions of fair representation, even when participants only vote on a small fraction of comments. Finally, an empirical evaluation on real data shows that the proposed algorithm provides representative outcomes in practice.

1 Introduction

A recent surge of interest in empowering citizens through online civic participation has spurred the development of a number of platforms (Salganik and Levy 2015; Ito et al. 2020; Shibata et al. 2019; Fishkin et al. 2019; Aragón et al. 2017; Iandoli, Klein, and Zollo 2009).[1] A particularly successful example is *Polis* (Small et al. 2021),[2] an open-source "system for gathering, analyzing and understanding what large groups of people think in their own words." It has been widely used by local and national government agencies around the world. Most notably, it is the basis of vTaiwan, a system commissioned by the government of Taiwan, whose participatory process — involving thousands of ordinary citizens — has led to new regulation of ridesharing services and financial technology. A similar (albeit commercial) system, *Remesh*,[3] allows users to "save resources by democratizing insights in live, flexible conversations with up to 1,000 people at the same time".

The key idea underlying both systems is simple and broadly applicable. Participants can submit free-text comments about the discussion topic at hand, and they can also choose to agree or disagree with others' comments, which are presented to them by the platform. An essential part of the process is the aggregation of these opinions towards an "understanding of what large groups of people think." Polis, for instance, displays a list of comments that received the most support among participants to whom they were shown. But this aggregation method may fail to represent minority groups, even those that are very large: if 51% of participants agree with one set of comments, while 49% of participants agree with another set of comments, only comments from the first set will appear on this list. Polis has recognized this problem and sought to mitigate it by employing a second, more elaborate procedure (Small et al. 2021).[4] While this procedure has produced interesting results in practice, it does not guarantee summarizations that are representative of the discussion in any rigorous sense.

In this paper, we reexamine opinion aggregation in systems like Polis and Remesh through the lens of computational social choice (Brandt et al. 2016). We observe that *selecting a subset of comments based on agreements and disagreements is the same as electing a committee based on approval votes*. From this viewpoint, the primary aggregation method used by Polis corresponds to classical approval voting (AV). There is substantial work — starting with the paper of Aziz et al. (2017) — on *approval-based committee elections* that seeks to avoid the shortcomings of approval voting by guaranteeing that the selected committee satisfies fairness notions. To define one such notion (which is not satisfied by AV), note that if the size of the committee is k and the number of voters is n, a subset of n/k voters is large enough to demand a seat on the committee if they agree on at least one candidate. This intuition is captured by a property called *justified representation*, which guarantees that every such subset of voters has an approved candidate on the committee.

[1] Including Considerit (https://consider.it), Citizens.is (https://citizens.is/), Make.org (https://make.org), and Kialo (https://www.kialo.com).

[2] https://pol.is

[3] https://www.remesh.ai

[4] The idea is to find clusters of participants with similar opinions and then ensure that each cluster is represented by comments that distinguish it from the others.

There is a major gap, however, between the literature on approval-based committee elections and the reality of systems like Polis and Remesh: these systems only have access to partial votes. For example, in the discussion facilitated by Polis around ridesharing regulation in Taiwan, 197 comments were submitted, but each participant only voted on 10.57 comments on average — roughly 5% of all comments.

Our main conceptual insight is that we may overcome the partial-information gap via statistical estimation.[5] In other words, it is possible to define and achieve probabilistic representation guarantees in a query model that can be implemented in real systems. The question is whether the query complexity of these guarantees is such that this approach is practical; we provide largely positive answers to this question, both theoretically and experimentally.

Our approach and results. Suppose that each voter (user) can be counted upon to express an opinion (approval/disapproval) on at least t candidates (comments) shown to them. A *query* asks a randomly chosen voter for their approval votes on a subset of candidates S such that $|S| \leq t$. Note that this is consistent with how Polis works, as participants express their agreement or disagreement with the comments shown to them by the system.

There are two primary ways in which these queries are less informative than full approval votes. First, the number of queries is upper bounded by the number of voters, which is far too small to evaluate all possible committees of desired size k, since typically t is on the order of k and both are substantially smaller than the number of candidates. For example, in the ridesharing regulation instance, the number of voters is 4,837, t is roughly 10, and if we choose $k = 10$ also then there are on the order of 10^{16} possible committees of size k. Second, since we will not query all voters about the same committee, we obtain only estimates of approval rates.

In Section 3, we start by abstracting away the second issue and focusing on the first: A single query in this setting gives us precise approval rates for a subset of alternatives of size t; we refer to these queries as *exact queries*. Ideally, we would like to design an algorithm that is *non-adaptive*, i.e., that decides on its queries in advance, because under an adaptive algorithm, the influence voters exert on the outcome may depend on whether they arrived earlier or later. However, we find that a non-adaptive algorithm must ask at least $\Omega(m^{11})$ exact queries to achieve justified representation with non-negligible probability, where m is the number of candidates. This result means that non-adaptive algorithms are impractical, and so we restrict our attention to adaptive algorithms. We adapt a local search algorithm of Aziz et al. (2018) to the case of exact queries and show that it can achieve *extended justified representation (EJR)* (which is stronger than justified representation) and *proportionality* with $\mathcal{O}(mk^2 \log k)$ queries.

In Section 4, we return to the realistic query model, where a query corresponds to a single voter; we refer to these queries as *noisy queries*. Since we need to estimate the answer to each exact query using multiple noisy queries, and we must control uncertainty, the query complexity of the adaptive algorithm for the same guarantees increases to $\mathcal{O}\left(mk^6 \log k \log m\right)$. By applying martingale theory, we develop an extension of this algorithm that allows the reuse of votes in a statistically sound way.

In Section 5 we show empirically (on real datasets from Polis and Reddit) that this extension allows us to find committees satisfying (approximate) EJR and proportionality despite access to very limited information (i.e., few voters, each voting on only a small fraction of the comments), while AV often does not achieve satisfactory representation.

2 Preliminaries

We begin by introducing the standard approval-based committee selection setting (Aziz et al. 2017). For $s \in \mathbb{Z}_{\geq 1}$, we use the notation $[s] = \{1, \ldots, s\}$. We have a set $N = [n]$ of n voters and a set C of m candidates. Each voter $i \in N$ approves a set of candidates $A_i \subseteq C$. We refer to the vector $\mathbf{A} = (A_1, \ldots, A_n)$ as an *approval profile*. The goal is to choose a *committee* $W \subseteq C$ of size $k \leq m$. The value k is called the *target committee size*. We refer to an algorithm that takes as input the profile and candidates and outputs a committee of size k as a *k-committee-selection algorithm*.

Notions of representation. We say that a group of voters $V \subseteq N$ is *ℓ-large* if $|V| \geq \ell \cdot \frac{n}{k}$; V is *ℓ-cohesive* if $|\bigcap_{i \in V} A_i| \geq \ell$. Aziz et al. (2017) introduced the following two fairness notions:

Definition 2.1 (Justified Representation (JR))**.** *A committee W provides JR if for every 1-large, 1-cohesive group of voters V, there exists a voter $i \in V$ who approves a member of W, i.e., $|A_i \cap W| \geq 1$.*

Definition 2.2 (Extended Justified Representation (EJR))**.** *A committee W provides EJR if for every $\ell \in [k]$ and every ℓ-large, ℓ-cohesive group of voters V, there exists a voter $i \in V$ who approves at least ℓ members of W, i.e., $|A_i \cap W| \geq \ell$.*

We also study the following approximate version of EJR:

Definition 2.3 (α-Extended Justified Representation (α-EJR))**.** *A committee W provides α-EJR if for every $\ell \in [k]$ and every $\frac{\ell}{\alpha}$-large, ℓ-cohesive group of voters V, there exists a voter $i \in V$ who approves at least ℓ members of W, i.e., $|A_i \cap W| \geq \ell$.*

Fernández et al. (2017) proposed another notion of representation called the *average satisfaction* of a group of voters V for a committee W, defined as $\text{avs}_W(V) = \frac{1}{|V|} \sum_{i \in V} |A_i \cap W|$. Similarly to Skowron (2021) and Fernández et al. (2017), we define following desirable property:

Definition 2.4 (α-proportionality)**.** *A committee W provides α-proportionality if for every $\lambda \in [0, k]$ and every $\frac{\lambda+1}{\alpha}$-*

[5] There is a body of work in computational social choice that deals with incomplete votes (Xia and Conitzer 2011; Filmus and Oren 2014), but the input consists of rankings, not approval votes, and the ultimate goal is to predict the winner according to a given voting rule, in contrast to our aim of achieving (combinatorial) representation guarantees.

large, λ-cohesive group of voters V, the average satisfaction *of voters in V is at least λ, that is, $avs_W(V) \geq \lambda$.*

Proposition 2 of Aziz et al. (2018) implies that for any $\varepsilon > 0$, there exist approval profiles and target committee sizes where no committee satisfies $(1+\varepsilon)$-proportionality. Hence, 1-proportionality is the best guarantee possible, and we will refer to it simply as *proportionality*.

Proportional approval voting. *Proportional Approval Voting (PAV)* is a widely studied committee selection algorithm; it returns a committee maximizing the *PAV score*, defined as

$$\text{PAV-SC}(W) := \frac{1}{n} \sum_{i \in N} \sum_{j=1}^{|A_i \cap W|} \frac{1}{j}.$$

PAV satisfies EJR and proportionality (Fernández et al. 2017; Aziz et al. 2018), but is NP-hard to compute (Aziz et al. 2015). Consequently, Aziz et al. (2018) propose a local search approximation of PAV (LS-PAV), which continues to satisfy EJR and proportionality, but, unlike PAV, runs in polynomial time. As we shall see, LS-PAV is a useful basis for algorithms in our query model.

3 Exact Queries

In the exact-query setting, the response R to a query Q consists of a proportion p_S for every subset $S \subseteq Q$, where p_S is the proportion of voters who only approve the candidates in S among the queried candidates Q, i.e.,

$$p_S := \frac{1}{n} \sum_{i \in N} \mathbb{I}[A_i \cap Q = S].$$

We refer to an algorithm that makes queries of size t, receives this type of response, and outputs a committee of size k as a *(k,t)-committee selection algorithm with exact queries*. We say an algorithm is *adaptive* if the queries it chooses depend on responses from previous queries; otherwise, we call it *non-adaptive*. Note that we allow all of our algorithms to be randomized. In the following, we ask how many queries are needed to guarantee the notions of representation introduced in Section 2.

3.1 Nonadaptive Algorithms

In this section, we think of m as large (many comments will be submitted to the system), while we think of k and t as small constants (since we wish to select only a few comments and voters have limited time). Hence, we are interested in the asymptotic query complexity as m grows large. In addition, we are interested primarily in lower bounds on the query complexity of non-adaptive algorithms. Therefore we consider only the minimal fairness criterion of JR (which is implied by EJR and proportionality).

An initial observation is that JR can always be guaranteed with $O(m^k)$ queries, as simply querying every set of k candidates provides all the information necessary to run PAV. For $k = 1$, this bound is tight, as voters could all approve only a single candidate, which will take a linear number of queries to find. Our first result is a tight quadratic lower bound for $k = 2$.

Theorem 3.1. *For any constants k and t such that $k \geq 2$, and any $\varepsilon > 0$, any non-adaptive (k,t)-committee selection algorithm that makes fewer than $\Omega(m^2)$ queries satisfies JR with probability at most ε.*

This result provides a separation between the adaptive and non-adaptive settings. As we show in Section 3.2, there is an adaptive (k,t)-committee selection algorithm guaranteeing JR with only $O(m)$ queries for any k and t such that $k < t$.

Theorem 3.1 follows from a more general result (Lemma B.1), which we present formally in Appendix B. Here we illustrate the argument by considering the special case where $t = k = 2$ and $\varepsilon = \frac{5}{6}$. Consider the adversary (which is simply a distribution over instances) that picks a random set of 3 candidates, call them 1, 2, and 3, and answers queries according to the approval matrix visualized in Figure 1(a): half of the voters approve only candidate 1, and the other half of the voters approve only candidates 2 and 3. To satisfy JR, the algorithm needs to include candidate 1 in the committee. However, if the algorithm never queries $\{1,2\}$, $\{1,3\}$, or $\{2,3\}$, it receives no information that can distinguish candidates 1, 2, and 3 from each other, so it can do no better than selecting a random pair from these three candidates, which will succeed with probability $\frac{2}{3}$. Indeed, assuming a bounded number of queries, this will often be the case. Specifically, suppose the algorithm makes fewer than $\frac{1}{18} \cdot \binom{m}{2}$ queries. Since there are $\binom{m}{2}$ pairs of candidates, the probability that the algorithm queried any randomly selected pair of candidates is at most $\frac{1}{18}$. By the union bound, the probability that the algorithm queries any of $\{1,2\}$, $\{1,3\}$, or $\{2,3\}$ is at most $\frac{1}{18} + \frac{1}{18} + \frac{1}{18} = \frac{1}{6}$. To summarize, for the algorithm to succeed, it either needs to get lucky during the querying phase, which happens with probability at most $\frac{1}{6}$, or get lucky during the selection phase, which happens with probability at most $\frac{2}{3}$. By the union bound, the algorithm succeeds with probability at most $\frac{1}{6} + \frac{2}{3} = \frac{5}{6}$.

A natural follow-up question is whether the $O(m^k)$ upper bound is tight for larger k. Interestingly, this is *not* the case for $k \geq 3$, as we prove in Appendix A:

Theorem 3.2. *For any $t \geq k - \lfloor (k+1)/4 \rfloor$, there exists a (k,t)-committee selection algorithm guaranteeing JR with $O(m^t)$ exact queries.*

However, the exponent does have a dependence on k. In particular, we find that guaranteeing JR requires $\Omega(m^3)$ queries starting at $k = 6$. The adversary employs an analogous strategy, now picking 7 random candidates and imposing the approval matrix depicted in Figure 1(b). Satisfying JR requires that the algorithm include candidate 1, which may be indistinguishable from the other six candidates unless the algorithm makes $\Omega(m^3)$ queries, since every candidate is approved by $\frac{6}{18}$ of the voters and every pair of candidates is approved by $\frac{2}{18}$ of the voters.

In Appendix B, we describe a computational search we conducted to find similar instances for larger values of k. The best lower bound obtained is as follows.

Theorem 3.3. *For any $\varepsilon > 0$, there exists a target committee size k with $k = \Theta(\log 1/\varepsilon)$ such that for all t, any non-*

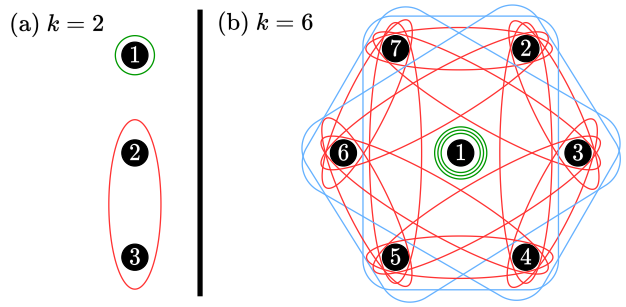

Figure 1: Adversarial approval matrices. Each region represents a disjoint set of voters of equal population who approve only the candidates within the region. In (a), queries of size $t \geq 2$ are needed to distinguish the candidates; in (b), we need $t \geq 3$.

adaptive (k,t)-committee selection algorithm with exact queries that makes fewer than $\Omega(m^{11})$ queries satisfies JR with probability at most ε.

This theorem closes the book on the (im)practicality of non-adaptive committee selection algorithms. We therefore turn our attention to adaptive algorithms.

3.2 An Efficient Adaptive Algorithm

In this section, we propose an adaptive algorithm based on LS-PAV (Aziz et al. 2018), and we show that it achieves proportionality and EJR with a practical number of queries.

For convenience, we introduce the following notation: For a committee W and candidates $c \in W$ and $c' \notin W$, let

$$\Delta(W, c', c) := \text{PAV-SC}(W \cup \{c'\} \setminus \{c\}) - \text{PAV-SC}(W)$$

denote the difference in PAV score obtained by replacing c with c' in W. Additionally, let

$$\Delta(W, c) := \text{PAV-SC}(W \cup \{c\}) - \text{PAV-SC}(W)$$

denote the marginal gain in PAV score by adding c to W.

LS-PAV starts with an arbitrary committee W and repeatedly replaces a committee member $c \in W$ with a candidate $c' \notin W$, provided the improvement to the PAV score satisfies $\Delta(W, c', c) \geq \frac{1}{k^2}$. Aziz et al. (2018) show that after at most $\mathcal{O}(k^2 \log k)$ swaps, no such swap pairs c, c' remain, at which point W satisfies proportionality and EJR.

We first observe that LS-PAV can be implemented using exact queries: For any set of candidates S, PAV-SC(S) can be computed using any query $Q \supseteq S$, as it is sufficient to know the proportion of voters that approve each subset of S. Hence, for any W, $c \in W$, and $c' \notin W$, $\Delta(W, c', c)$ can be computed using a query Q that contains both W and c'. Using $\left\lceil \frac{m-k}{t-k} \right\rceil$ queries of size t, we can cover all $m - k$ candidates that are not in W, which leads to an overall query complexity of $\mathcal{O}\left(mk^2 \log k\right)$.

We next present a version of LS-PAV, which we call α-PAV (Algorithm 1), that has the same query complexity as LS-PAV for finding a committee that satisfies EJR and proportionality, but lower query complexity for approximate ($\alpha < 1$) α-EJR and α-proportionality.

Algorithm 1: (k,t)-α-PAV

```
Choose W ∈ (C choose k), c ∈ W, and c' ∉ W arbitrarily
γ ← ∞
while γ ≥ 1/(αk) do
    W ← W ∪ {c'} \ {c}
    Choose Q = {Q_i}, with |Q_i| = t, s.t. W ⊆ ⋂Q and C ⊆ ⋃Q
    c' ← arg max_{x∉W} Δ(W, x)                ▷ (using Q)
    c ← arg max_{x∈W} Δ(W, c', x)             ▷ (using Q)
    γ ← Δ(W, c')
return W
```

Besides introducing the approximation parameter α, we make two other modifications to LS-PAV: First, Algorithm 1 terminates when there is no candidate c' such that $\Delta(W, c') \geq \frac{1}{k}$ (for $\alpha = 1$), while LS-PAV terminates when there is no pair c, c' such that $\Delta(W, c', c) \geq \frac{1}{k^2}$. As we shall see in Lemma 3.6, the termination condition of Algorithm 1 is weaker than that of LS-PAV, implying that it may terminate earlier. Second, instead of considering all possible swaps c, c', we only consider adding the candidate c' with the largest $\Delta(W, c')$. This modification makes the algorithm slightly simpler and more computationally efficient (by a factor of k).

Theorem 3.4. *For any $m \geq t > k$, Algorithm 1 yields a committee satisfying α-proportionality and α-EJR while making at most*

$$\left\lceil \frac{m-k}{t-k} \right\rceil \frac{\alpha k^2}{(1-\alpha)k+1} H_k$$

queries, where H_k is the k^{th} harmonic number. For $\alpha = 1$, this leads to a query complexity of $\mathcal{O}\left(mk^2 \log k\right)$ while for any fixed $\alpha < 1$, this leads to a query complexity of $\mathcal{O}\left(mk \log k\right)$.

The proof of Theorem 3.4 essentially follows from the following two lemmas, the first of which uses the notation

$$\Delta^*(W) := \max_{c \in C} \Delta(W, c).$$

Lemma 3.5. *If a committee W satisfies $\Delta^*(W) < \frac{1}{\alpha k}$, then W satisfies α-EJR and α-proportionality.*

Lemma 3.6. *For any committee W and $c \notin W$, we have that $\max_{x \in W} \Delta(W, c, x) \geq \frac{(k+1)\Delta(W,c)-1}{k}$. In particular, if $\Delta(W, c) \geq \frac{1}{\alpha k}$, then $\max_{x \in W} \Delta(W, c, x) \geq \frac{(1-\alpha)k+1}{\alpha k^2}$.*

Lemma 3.5 guarantees that when Algorithm 1 terminates the desired fairness properties are satisfied. Lemma 3.6 establishes that the PAV score increases over the algorithm's run. This bounds the number of swaps it performs since PAV-SC(W) is at most H_k.

Lemma 3.5 is a generalization of the lower bound from Lemma 1 of Skowron (2021). This generalization is useful because it states that to establish EJR and proportionality of

any given committee W (no matter how it is derived), it is sufficient to prove that $\Delta^*(W)$ is small; hence it can be used as a certificate of satisfaction. In Appendix E, we show that standard PAV and LS-PAV satisfy $\Delta^*(W) < \frac{n}{k}$, which is noteworthy in that it provides a simple proof of the known result that they satisfy EJR and proportionality.

We observe that for exact queries, an α-approximation with $\alpha < 1$ improves the query complexity by a factor of k. In the next section, we will see that such an approximation yields an even larger improvement in query complexity for noisy queries, as it also reduces the accuracy with which we need to estimate $\Delta(W, c', c)$.

4 Noisy Queries

We now turn to a query model that includes the noise abstracted away in Section 3. In order to represent voters arriving to the platform one-by-one, we assume that each time the algorithm performs a query $Q \subseteq C$ a voter $i \in N$ is selected independently and uniformly at random, and then the algorithm observes their votes on the queried candidates $Q \cap A_i$. We discuss this modeling choice more thoroughly in Appendix F. We refer to an algorithm that performs queries of size t, receives as a response the votes of a single voter, and outputs a committee of size k as a (k,t)-*committee selection algorithm with noisy queries*.

To see the connection between this query model and the previous one, note that an algorithm with noisy queries can approximate an exact query Q by estimating the values of p_S by taking the empirical proportion of repeated samples. By standard sample complexity bounds, using $\Theta\left(\log(2^t/\delta)/\varepsilon^2\right)$ queries, a noisy-query algorithm could guarantee $\pm\varepsilon$ estimates of p_S for all $S \subseteq Q$ with probability $1-\delta$. Hence if an exact-query algorithm requires no more than $\text{poly}(m)$ queries with additive ε error, then it can be implemented using a factor of $\Theta(\log m)$ more noisy queries and yield a correct result with probability $1-\delta$. What's more, this $\log$ factor is in some cases necessary when moving from the exact-query to the noisy-query setting. In Appendix C, we demonstrate instances for which a non-adaptive exact-query algorithm needs only $\Theta(m)$ queries, while in order to be correct with any fixed probability δ, a non-adaptive noisy-query algorithm requires $\Omega(m \log m)$ queries.

Conversely, notice that one can use exact queries to simulate noisy queries. Indeed, p_S is exactly the probability that an incoming voter will vote yes on candidates S and no on $Q \setminus S$ in response to a query Q. An algorithm with access to exact query values can simply sample a voter response and feed it to a noisy-query algorithm. Therefore, the lower bounds on the query complexity of exact-query, non-adaptive algorithms, in particular Theorem 3.3, apply to noisy-query non-adaptive algorithms as well. As the number of candidates becomes large, adaptivity is therefore necessary to attain theoretical guarantees — mirroring the approach of online platforms in practice.

A natural starting point is the exact-query adaptive algorithm, namely Algorithm 1. Indeed, it can be adapted to the noisy setting by replacing exact queries with a sufficient number of noisy queries, ℓ, to obtain high-probability bounds on Δ, yielding Algorithm 2.

Algorithm 2: (k,t)-noisy-α-PAV

$\ell \leftarrow \left\lceil 288 \left(\frac{\alpha k^2}{(1-\alpha)k+1}\right)^2 \log\left(\frac{8mk^4}{\delta}\right)\right\rceil$
Choose $W \in \binom{C}{k}$, $c \in W$, and $c' \notin W$ arbitrarily
$\gamma \leftarrow \infty$
while $\gamma \geq \frac{1}{\alpha k} - \frac{(1-\alpha)k+1}{12\alpha k^2}$ **do**
  $W \leftarrow W \cup \{c'\} \setminus \{c\}$
  Choose $\mathcal{Q} = \{Q_i\}$, with $|Q_i| = t$, such that $W \subseteq \bigcap \mathcal{Q}$ and $C \subseteq \bigcup \mathcal{Q}$
  Ask each query $Q \in \mathcal{Q}$ to ℓ new voters
  $\hat{\Delta}(W, x) \leftarrow$ estimate of $\Delta(W, x)$ using ℓ voters from query Q containing $W \cup \{x\}$ ▷ $\forall x \notin W$
  $\hat{\Delta}(W, x, y) \leftarrow$ estimate of $\Delta(W, x, y)$ using ℓ voters from Q containing $W \cup \{x\}$ ▷ $\forall x \notin W, \forall y \in W$
  $c' \leftarrow \arg\max_{x \notin W} \hat{\Delta}(W, x)$
  $c \leftarrow \arg\max_{x \in W} \hat{\Delta}(W, c', x)$
  $\gamma \leftarrow \hat{\Delta}(W, c')$
return W

The key is to choose ℓ large enough that if the termination condition is not met, i.e., we have $\hat{\Delta}(W, c') < \frac{1}{\alpha k} - \frac{(1-\alpha)k+1}{12\alpha k^2}$, the resulting swap is guaranteed to yield a positive improvement in the PAV-score, such that the number of steps of the algorithm is bounded. With the choice of ℓ in Algorithm 2, we obtain the following theorem, whose proof can be found in Appendix G.

Theorem 4.1. *For any $m \geq t > k$, with probability at least $1-\delta$, Algorithm 2 returns a committee that satisfies α-EJR and α-proportionality after querying no more than*

$$578 H_k \left\lceil \frac{m-k}{t-k} \right\rceil \left(\frac{\alpha k^2}{(1-\alpha)k+1}\right)^3 \log\left(\frac{4mk^4}{\delta}\right)$$

voters. For any fixed $\delta > 0$, if $\alpha = 1$, this leads to a query complexity of $\mathcal{O}\left(mk^6 \log k \log m\right)$ and if $\alpha < 1$, this leads to a query complextiy of $\mathcal{O}\left(mk^3 \log k \log m\right)$.

While Algorithm 2 achieves good worst-case query complexity, it may be suboptimal on certain instances because of two reasons: (i) after each swap, Algorithm 2 discards all previous information so each candidate is reassessed from scratch, and (ii) it presents each candidate $c \notin W$ to the same number of voters, even though it may quickly become apparent that some candidates are more promising than others.

To address issue (i), we can use all past votes to compute bounds on Δ. A difficulty with this approach is that past voters may not have voted on all candidates in W (which is necessary to directly estimate $\Delta(W, c)$), since they may have been queried on a different committee W'. But we can nonetheless use these past votes to obtain upper and lower bounds on estimated values.

To address issue (ii), we can present promising candidates to voters more often. Further, it is possible to perform swaps as soon as we are confident they yield an increase of the PAV-score of at least some value ε, rather than first querying

a predetermined number of voters as in Algorithm 2. This allows us to quickly add highly beneficial candidates since a lower bound estimate on Δ will be large enough after relatively few queries.

These ideas are incorporated into Algorithm 4, called ucb-α-PAV; see Appendix H for a formal description of the algorithm and an analysis of its query complexity.

5 Experiments

Since the analysis in the theoretical sections considers worst-case approval profiles, it is possible that, in practice, we may be able to find good committees with fewer queries than required by Theorem 4.1 and Theorem H.1. We investigate this question empirically on real data from online discussions with only a few hundred voters, each voting on only a fraction of all comments.

Datasets. Polis provides open-use data from real deliberations hosted on their platform.[6] These include, for instance, a discussion organized by the government of Taiwan, which led to the successful regulation of Uber. Since participants typically only vote on a fraction of comments, most votes are missing. To be able to simulate the proposed adaptive algorithms, we first infer these missing votes using a matrix factorization library, LensKit.[7] Importantly, we infer votes only for the purpose of the experiments; if our algorithms were executed during the discussion, they would adaptively query users about the relevant comments without relying on any inference method.

In most datasets, we observe several comments that are nearly universally approved. Since these comments make achieving EJR and proportionality trivial, we remove comments approved by more than 60% of participants to obtain nontrivial instances. This step may also be appropriate in practice to gain insights into participants' opinions beyond uncontroversial issues.

The number of queried voters L ranges from 87 to 1000 across the 13 datasets (see Appendix I for details). For all datasets, we assume that each voter votes on $t = 20$ comments. Since the total number of comments m ranges from 31 to 1719 across datasets, the percentage of comments each voter votes on, t/m, ranges from 1% to 65%. For each dataset, we experiment on target committee sizes $k = 5, 7, 10$. Hence, there are a total of $13 \cdot 3 = 39$ instances (times 10 random seeds).

The second dataset we consider consists of Reddit discussions.[8] To obtain an interesting dataset, we combined voting data from two subreddits, r/politics and r/Conservative, which are arguably situated at opposite ends of the political spectrum. More details about this dataset can also be found in Appendix I.

Algorithms. We evaluate noisy-α-PAV (Algorithm 2) and ucb-α-PAV (Algorithm 4). Both query L voters (see Table 3) in random order, each of whom votes on $t = 20$ comments. To enable these algorithms to swap candidates after querying only a small number of voters, we make the following modifications: In Algorithm 2, we treat ℓ, the number of times we ask voters about each candidate, as a parameter. Similarly, in Algorithm 4, we replace the numerator in the confidence intervals of $\Delta^-(W, c', c)$ with a parameter θ, and we set $\varepsilon = 0$. Both ℓ and θ were chosen based on validation on a separate datasets, see Appendix I for details. We run both algorithms on all the L voters, rather than terminating as soon as we can guarantee $\Delta^*(W) < \frac{1}{\alpha k}$ (and hence EJR and proportionality).

To obtain an upper bound on the attainable performance, we execute α-PAV (Algorithm 1) with access to exact queries. To obtain the best possible α, we let Algorithm 1 run as long as the swap increases the PAV score, i.e., $\Delta(W, c', c) > 0$, instead of terminating as soon as $\Delta(W, c') < 1/k$ (which would be sufficient to guarantee EJR and proportionality).

To verify that the proposed algorithms do indeed take the complementarity of different candidates into account, we also compare against standard approval voting (AV) with access to all votes, which simply selects the k candidates with the most approval votes.

Performance Metric. As a performance metric, we use $\hat{\alpha} := \frac{1}{k\Delta^*(W)}$, where W is the committee selected by the respective algorithm. According to Lemma 3.5, $\alpha > \hat{\alpha}$, so this implies $\hat{\alpha}$-EJR and $\hat{\alpha}$-proportionality. As discussed in Section 2, $\alpha = 1$ is the best that can be guaranteed across all possible approval profiles. Note that α may be larger than $\hat{\alpha}$, hence obtaining $\hat{\alpha} = 1$ is a sufficient, but not a necessary condition for proportionality and EJR. Nevertheless, we will use $\hat{\alpha}$ as a metric for two reasons: first, verifying whether $\alpha \geq 1$ (i.e., whether a committee satisfies EJR and proportionality) is computationally hard (Aziz et al. 2017), which makes it impractical for evaluation; and the stronger condition $\hat{\alpha} \geq 1$ provides the additional benefit that EJR and proportionality can easily be verified through Lemma 3.5. Second, one could argue that $\hat{\alpha}$ is a meaningful quantity in its own right since it (or rather its inverse $1/\hat{\alpha}$) measures how much voter satisfaction could be improved by adding another candidate (giving lower weight to voters who already have many approved candidates).

Polis Results. In Figure 2, we show the $\hat{\alpha}$ achieved on all the Polis datasets for each of the four algorithms.

(Recall that higher $\hat{\alpha}$ is better and that $\hat{\alpha} \geq 1$ implies proportionality and EJR.) As expected, α-PAV performs best since it has access to exact queries. Note that it often achieves an α substantially larger than 1, which means that better representation can be guaranteed for the corresponding dataset than in the worst case. AV performs surprisingly well in most experiments, but in 38% of the cases, it yields $\hat{\alpha}$ smaller than 1 (and sometimes much smaller). We conclude that for some datasets, it is important to take the complementarity of candidates into account rather than selecting them individually. The challenge for the proposed algorithms is to do so while being sample-efficient. We see that noisy-α-PAV often fails to achieve

[6] https://github.com/compdemocracy/openData

[7] https://lenskit.org

[8] https://www.kaggle.com/datasets/josephleake/huge-collection-of-reddit-votes

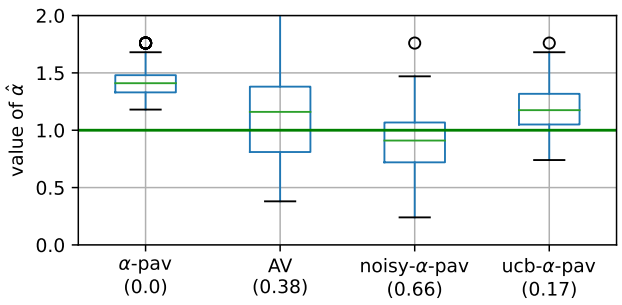

Figure 2: Boxplots where datapoints correspond to the 39 Polis problems ($\times 10$ random seeds). The top / bottom whiskers indicate the maximal / minimal points (except outliers, which are marked by circles), the line in the middle is the median, and the bottom and top of the boxes are the 1st and 3rd quartiles, respectively. The numbers in parenthesis are the fractions of problems where the respective algorithm yields a $\hat{\alpha} \leq 1$.

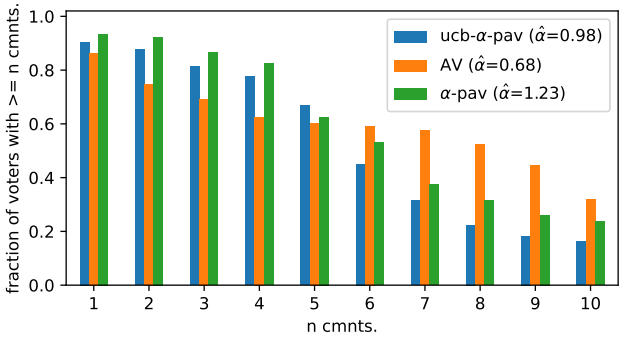

Figure 3: Results on Reddit dataset (with $L = 608, m = 2135, k = 10$): the fraction of voters (y-axis) that approve of at least $1, 2, .., 10$ candidates (x-axis) among the selected committee of size $k = 10$.

an $\hat{\alpha} \geq 1$. We know from Theorem 4.1 that given enough queries, noisy-α-PAV achieves $\hat{\alpha} \geq 1$, so this failure is due to the low number of queried voters. By contrast, ucb-α-PAV yields $\hat{\alpha} \geq 1$ in 83% of the cases, and $\hat{\alpha} \geq 0.75$ in all cases, which indicates that the proposed extensions (i.e., querying promising candidates more often, swapping as soon as possible, and reusing voters) indeed lead to more efficient use of data.

Reddit Results. To illustrate why approval voting can perform poorly despite having access to the full votes, we execute the algorithms on the Reddit dataset described above. In this experiment, AV achieves only $\hat{\alpha} = 0.68$. To understand why this may happen, we show in Figure 3 the fraction of voters who have at least $1, \ldots, 10$ approved comments in the committee. We see that AV yields a committee where a high fraction of voters approve many candidates, e.g., about 60% of voters approve 7 or more candidates, whereas for α-PAV this is the case for only about 40%. This comes at the cost of a high fraction of voters who are poorly represented by AV, e.g., about 25% of voters get at most one approved candidate, whereas for α-PAV, this percentage is less than 10%. This is to be expected as approval voting does not take the complementarity of candidates into account and can therefore lead to less equitable results. Finally, we observe that ucb-α-PAV achieves an $\hat{\alpha}$ close to 1, and its approval fractions look similar to α-PAV, i.e., more equitable than AV. It is interesting that ucb-α-PAV performs well on this example, since it only has access to $t = 20$ votes for each of the $L = 608$ queried candidates, while it has to select from a large number of comments, $m = 2135$.

6 Discussion

This work bridges the gap between online civic-participation systems, such as Polis, and committee-election methods by enabling them to handle incomplete votes. To deploy the proposed algorithms on such platforms, two practical issues must be considered.

First, our adaptive approach requires control over what the Polis creators call *comment routing* (Small et al. 2021): the algorithm that decides which comments are shown to which participants. However, if there is an existing comment routing algorithm in place, shared control is possible: each algorithm could determine part of the slate of comments shown to a participant, or the participants themselves can be divided between the algorithms.

Second, in our analysis, we assumed that all comments have been submitted — or all candidates are known — at the time we run our algorithms. Note that they could be extended straightforwardly to a growing set of comments, but we would inevitably lose the representation guarantees for comments that were submitted late if not enough participants could vote on them. In practice, this could be resolved by setting a comment submission deadline, which has been done previously by Polis.

An alternative to our approach would be to complete partial approval votes using collaborative filtering (Resnick and Varian 1997). The completed approval votes can then be aggregated through any approval-based committee election rule, such as PAV. The disadvantage of this approach is that it is unlikely to lead to worst-case guarantees of the type we establish in this paper.

Finally, we emphasize that our approach may be applicable to social media more generally. For instance, as mentioned in Section 5, Reddit users also approve or disapprove comments through upvotes and downvotes. However, Reddit uses these inputs to produce a *ranking* of the comments, in contrast to our goal of selecting a subset. There is work on obtaining justified-representation-type guarantees for rankings (Skowron et al. 2017), which could possibly be extended to the setting of incomplete votes using the techniques developed in this paper. Even more broadly, this article provides insights into how to fairly represent opinions of groups given incomplete information, which may be relevant for the design of more constructive online ecosystems.

References

Aragón, P.; Kaltenbrunner, A.; Calleja-López, A.; Pereira, A.; Monterde, A.; Barandiaran, X. E.; and Gómez, V. 2017. Deliberative platform design: The case study of the online discussions in decidim Barcelona. In *Proceedings of the 9th International Conference on Social Informatics (SocInf)*, 277–287.

Ash, R. B. 1990. *Information Theory*. Dover Publications.

Aziz, H.; Brill, M.; Elkind, E.; Freeman, R.; and Walsh, T. 2017. Justified Representation in Approval-Based Committee Voting. *Social Choice and Welfare*, 42(2): 461–485.

Aziz, H.; Elkind, E.; Huang, S.; Lackner, M.; Sánchez-Fernández, L.; and Skowron, P. 2018. On the Complexity of Extended and Proportional Justified Representation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, 902–909.

Aziz, H.; Gaspers, S.; Gudmundsson, J.; Mackenzie, S.; Mattei, N.; and Walsh, T. 2015. Computational Aspects of Multi-Winner Approval Voting. In *Proceedings of the 14th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 107–115.

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

Fernández, L. S.; Elkind, E.; Lackner, M.; García, N. F.; Arias-Fisteus, J.; Basanta-Val, P.; and Skowron, P. 2017. Proportional Justified Representation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 670–676.

Filmus, Y.; and Oren, J. 2014. Efficient Voting via the Top-k Elicitation Scheme: A Probabilistic Approach. In *Proceedings of the 15th ACM Conference on Economics and Computation (EC)*, 295–312.

Fishkin, J.; Garg, N.; Gelauff, L.; Goel, A.; Munagala, K.; Sakshuwong, S.; Siu, A.; and Yandamuri, S. 2019. Deliberative democracy with the Online Deliberation platform. In *Proceedings of the 7th AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*, 1–2.

Iandoli, L.; Klein, M.; and Zollo, G. 2009. Enabling On-Line Deliberation and Collective Decision-Making through Large-Scale Argumentation: A New Approach to the Design of an Internet-Based Mass Collaboration Platform. *International Journal of Decision Support System Technology (IJDSST)*, 1(1): 69–92.

Ito, T.; Suzuki, S.; Yamaguchi, N.; Nishida, T.; Hiraishi, K.; and Yoshino, K. 2020. D-Agree: Crowd Discussion Support System Based on Automated Facilitation Agent. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, 13614–13615.

Resnick, P.; and Varian, H. R. 1997. Recommender Systems. *Communications of the ACM*, 40(3): 56–58.

Salganik, M. J.; and Levy, K. E. C. 2015. Wiki surveys: open and quantifiable social data collection. *PloS one*, 10(5): e0123483.

Shibata, D.; Moustafa, A.; Ito, T.; and Suzuki, S. 2019. On Facilitating Large-Scale Online Discussions. In *PRICAI 2019: Trends in Artificial Intelligence*, 608–620. Springer International Publishing.

Skowron, P. 2021. Proportionality Degree of Multiwinner Rules. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, 820–840.

Skowron, P.; Lackner, M.; Brill, M.; Peters, D.; and Elkind, E. 2017. Proportional Rankings. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 409–415.

Small, C.; Bjorkegren, M.; Erkkilä, T.; Shaw, L.; and Megill, C. 2021. Polis: Scaling Deliberation by Mapping High Dimensional Opinion Spaces. *Revista De Pensament I Anàlisi*, 26(2).

Xia, L.; and Conitzer, V. 2011. Determining Possible and Necessary Winners Given Partial Orders. *Journal of Artificial Intelligence Research*, 41: 25–67.

Appendix

A Proof of Theorem 3.2

Theorem 3.2. *For any $t \geq k - \lfloor (k+1)/4 \rfloor$, there exists a (k,t)-committee selection algorithm guaranteeing JR with $O(m^t)$ exact queries.*

Proof. Consider Algorithm 3, described below.

Algorithm 3: (k,t)-non-adaptive (for $t \geq k - \lfloor (k+1)/4 \rfloor$)

```
Query every set of candidates of size t
for i ← 1, 2, ..., t do
    c_i ← approval winner among voters not approving c_1, c_2, ..., c_{i-1}
for i ← 1, 2, ..., ⌊(k+1)/4⌋ do
    c_{t+i} ← arbitrary default candidate
    for c ∈ C do
        A ← set of voters approving of c but not any of c_1, c_2, ..., c_{i+⌊(t-i)/2⌋}, c_{t+1}, c_{t+2}, ..., c_{t+i-1}
        B ← set of voters approving of c but not any of c_1, c_2, ..., c_{i-1}, c_{i+⌊(t-i)/2⌋+1}, c_{i+⌊(t-i)/2⌋+2}, ... c_{t+i-1}
        if |A| ≥ n/k and |B| ≥ n/k then
            c_{t+i} ← c
return {c_1, c_2, ..., c_k}
```

It is straightforward (but tedious) to verify that the **if** condition can be checked using only information about sets of voters of size t as long as $t \geq k - \lfloor (k+1)/4 \rfloor$ (this bound is tight). Thus, Algorithm 3 is indeed a non-adaptive (k,t)-committee selection algorithm with exact queries.

For each $i \in \{1, 2, \ldots, t + \lfloor (k+1)/4 \rfloor\}$, we say that a voter is *satisfied on round i* if it approves of c_i, but none of the previously selected candidates $c_1, c_2, \ldots, c_{i-1}$, and we say that a voter is *satisfied by round i* if it was satisfied on some round $j \leq i$. We prove that the final committee satisfies JR by counting the fraction of voters that are satisfied on each round. Indeed, JR is equivalent to the property that there is no 1-cohesive $\frac{1}{k}$-fraction of voters that is left unsatisfied by the k^{th} round.

The case where $k \leq t$ is easy: on each of the first k rounds, either we satisfy a $\frac{1}{k}$ fraction of voters, or there is no 1-cohesive set of $\frac{1}{k}$ unsatisfied voters. Thus, by round k, either all voters are satisfied, or the remaining set of unsatisfied voters has no 1-cohesive set of size $\frac{1}{k}$.

Now suppose that $k > t$. For each $1 \leq i \leq t$, let x_i denote the fraction of voters that are satisfied on round i. Again, if any $x_i < \frac{1}{k}$, it means that there is no 1-cohesive $\frac{1}{k}$-fraction of unsatisfied voters after round i, so JR is already satisfied. So assume each $x_i \geq \frac{1}{k}$. Further, if on any round $t+i$ where $t+1 \leq t+i \leq k$, we fail to find a candidate c making the **if** condition true, we claim that JR is already satisfied. For if JR were not satisfied, then there would be some candidate c approved by a $\frac{1}{k}$-fraction of voters S who approve of no previous candidates. Clearly, we would then have $S \subseteq A$ and $S \subseteq B$, so A and B both contain at least $\frac{1}{k}$ fractions of voters.

Thus, we may assume that, for each $i \in \{1, 2, \ldots, \lfloor (k+1)/4 \rfloor\}$, candidate c_{t+i} satisfies the **if** statement on round $t+i$. Consider an arbitrary round $t+i$. Let A and B denote the respective values of the variables on the iteration of the inner loop where c_{t+i} was set to its ultimate value. Observe that the candidates enumerated in the definitions of A and B cover all previously selected candidates. This means that $A \cap B$ is precisely the set of voters approving c_{t+i} and not any of the previous candidates; in other words, $A \cap B$ is the set of voters satisfied on round i. On the other hand, since candidates $c_1, c_2, \ldots, c_{i-1}$ are enumerated in the definitions of both sets, it follows that $A \cup B$ is a set of voters approving c_{t+i} but not any of $c_1, c_2, \ldots, c_{i-1}$. This means that $A \cup B$ can contain at most an x_i fraction of voters, for otherwise candidate c_{t+i} should have been selected earlier, on round i. Thus, we may lower bound the fraction of voters satisfied on round $t+i$ as

$$\frac{1}{n}\left(|A \cap B|\right) = \frac{1}{n}\left(|A| + |B| - |A \cup B|\right) \geq \frac{1}{n}\left(\frac{n}{k} + \frac{n}{k} - nx_i\right) = \frac{2}{k} - x_i.$$

Summing over each of the first k rounds, the number of satisfied voters is

$$\begin{aligned}
\sum_{i=1}^{k}(\#\text{ satisfied voters on round } i) &\geq \sum_{i=1}^{t} x_i + \sum_{i=1}^{k-t}\left(\frac{2}{k} - x_i\right) \\
&= \sum_{i=1}^{k-t}\left(x_i + \frac{2}{k} - x_i\right) + \sum_{i=k-t+1}^{t} x_i \\
&\geq \sum_{i=1}^{k-t}\frac{2}{k} + \sum_{i=k-t+1}^{t}\frac{1}{k} \\
&= (k-t)\cdot\frac{2}{k} + (t-(k-t))\cdot\frac{1}{k} \\
&= \frac{2(k-t)+(2t-k)}{k} \\
&= \frac{k}{k} = 1.
\end{aligned}$$

Since all voters are satisfied by round k, the final committee satisfies JR. □

B Proofs of lower bounds for non-adaptive algorithms with exact queries

In this section, we prove Theorems 3.1 and 3.3. Both theorems can be derived as applications of the following lemma, which formalizes the properties we require of the adversarial instances shown in Figure 1. We only use statement (ii) in this paper, but we include statement (i) as well because we believe it may be of independent interest.

Lemma B.1. *Suppose that, for some integers $0 \leq h \leq k_0 \leq \ell$, there exists a probability distribution $\{x_S\}_{S\subseteq[\ell]}$ over subsets of $[\ell]$ such that:*

(1) For any sets $T_1, T_2 \subseteq [\ell]$ such that $|T_1| = |T_2| \leq h$,

$$\sum_{S\supseteq T_1} x_S = \sum_{S\supseteq T_2} x_S.$$

(2) For some $s^* \in [\ell]$, $x_{\{s^*\}} \geq \frac{1}{k_0}$.

Then there exist exact query adversaries for which:

(i) For any $t \leq h$, any non-adaptive (k,t)-committee selection algorithm satisfies JR with probability at most $\left(\frac{k_0}{\ell}\right)^{\lfloor k/k_0\rfloor}$.

(ii) For any $t > h$, for any $\delta > 0$, any non-adaptive (k,t)-committee selection algorithm that makes fewer than $\Omega(m^{h+1})$ queries satisfies JR with probability at most $\left(\frac{k_0}{\ell}\right)^{\lfloor k/k_0\rfloor} + \delta$.

Proof. Given such a probability distribution $\{x_S\}_{S\subseteq[\ell]}$, we define the query adversary as follows. This adversary will be a distribution over profiles over $[m]$ candidates, for some sufficiently large m to be determined later. We denote a given non-adaptive (k,t)-committee selection algorithm by $\mathcal{A}$.

First let $k = pk_0 + r$, where p is a nonnegative integer and $0 \leq r < k_0$. Partition the candidates into $[m] = C_1 \cup C_2 \cup \cdots \cup C_p \cup D$ where, for each $i \in [p]$, $|C_i| = \lfloor (m-r)/p \rfloor$, and D contains the remaining candidates, of which there are at least r. For each i, the adversary will randomly select a subset of ℓ distinct candidates $S^i := \{c_1^i, c_2^i, \ldots, c_\ell^i\} \subseteq C_i$. The adversary chooses all subsets and orderings with equal probability, independently for each C_i. The adversary will then respond to all queries according to the following approval matrix.

We partition the voters into $p + r$ distinct "parties" $P_1, \ldots, P_p$ and $Q_1, \ldots, Q_r$, and every voter is a member of exactly one party. Each party P_i contains a k_0/k proportion of voters, and voters in P_i approve only of some subset of the candidates contained in $S^i \subseteq C_i$, and none of the other candidates. For these P_i, for all $S \subseteq [\ell]$, let the fraction of voters whose approval set is exactly $\{c_s^i \mid s \in S\}$ be equal to x_S. Every party Q_j is a $1/k$ proportion of the voters, and each voter belonging to Q_j approves only of one candidate $d_j \in D$ which is specific to Q_j.

Let us say that $\mathcal{A}$ *h-covers* a given set of candidates $S \subset [m]$ if $\mathcal{A}$ ever submits a query $T \subseteq [m]$ such that $|T \cap S| > h$. If, for any of the parties P_i with $i \in [p]$, the algorithm fails to h-cover the set S^i, then condition (1) implies that all ℓ of these candidates are completely symmetric (i.e. indistinguishable) to $\mathcal{A}$ given all of its query responses. Since each of the distinguished candidates $c_{s^*}^i$ is distributed uniformly at random among the candidates S^i, $\mathcal{A}$ selects $c_{s^*}^i$ to be part of its chosen committee with probability at most $\min(k_i/\ell, 1)$, where k_i is the number of candidates that $\mathcal{A}$ selects from P_i.

However, in order to satisfy JR, $\mathcal{A}$ must select at least r candidates from D, since there are r distinct candidates in D approved by the r parties Q_j, which are disjoint fractions of $1/k$ of the voters. In order to satisfy JR $\mathcal{A}$ must also select the distinguished

candidate $c^i_{s^*} \in C_i$ for each party P_i, since condition (2) implies that for each P_i at least a $\frac{1}{k_0} \cdot \frac{k_0}{k} = \frac{1}{k}$ fraction of the voters approve only $c^i_{s^*}$ and none of the other candidates.

This already implies (i): assuming that $\mathcal{A}$ selects at least r candidates from D, then if $t \leq h$, it is impossible for $\mathcal{A}$ to h-cover S^i with any number of queries, and thus $\mathcal{A}$ succeeds in satisfying JR with probability at most

$$\Pr[\mathcal{A} \text{ selects } c^i_{s^*}, \text{ for all } i \in [p]] \leq \frac{k_1}{\ell} \cdot \frac{k_2}{\ell} \cdot \dots \cdot \frac{k_p}{\ell} = \frac{k_1 k_2 \dots k_p}{\ell^p} \leq \frac{k_0^p}{\ell^p} = \left(\frac{k_0}{\ell}\right)^p.$$

Here the second inequality holds due to the constraint that $k_1 + k_2 + \dots + k_p \leq k - r = pk_0$, since $\mathcal{A}$ must select at least r candidates from D.

To prove (ii), we must analyze the likelihood that an algorithm $\mathcal{A}$ making a small number of queries h-covers any given S^i. Let us suppose that $\mathcal{A}$ knows the partition of candidates into $C_1 \cup C_2 \cup \dots \cup C_p \cup D$, knows everything about the approval matrix except for which sets S^i were chosen within each party P_i, and is allowed to make at most cm^{h+1} queries within each party P_i, separately, where

$$c := \frac{\delta}{2^{t+\ell}\, \ell!\, p^{h+2}}.$$

Clearly, these assumptions only make the algorithm $\mathcal{A}$ stronger: an impossibility for this kind of algorithm implies the desired lower bound. Fix a party P_i. For sufficiently large m, every set $S \subseteq C$ of ℓ candidates that is h-covered by a query $T \subseteq [m]$ of size t can be decomposed into two parts: a set of size j (where $h+1 \leq j \leq t$) that is contained in T, and a set of size $\ell - j$ that is contained in $C_i \setminus T$. Thus, the number of sets S of size ℓ within C_i that any single query can h-cover is exactly

$$\sum_{j=h+1}^{t} \binom{t}{j} \binom{\lfloor (m-r)/p \rfloor - t}{\ell - j} \leq 2^t \binom{m/p}{\ell - h - 1} \leq 2^t \left(\frac{m}{p}\right)^{\ell-(h+1)}$$

provided that $\ell - h - 1 \leq m/2p$, which holds for sufficiently large m.

Since we have that $\mathcal{A}$ made at most cm^{h+1} queries within party P_i by assumption, at most

$$cm^{h+1} \cdot 2^t \left(\frac{m}{p}\right)^{\ell-(h+1)} = \frac{2^t}{p^{\ell-(h+1)}} cm^\ell$$

sets of size ℓ can be h-covered. Since, within each party P_i there are a total of

$$\binom{\lfloor (m-r)/p \rfloor}{\ell} \geq \frac{(\lfloor (m-r)/p \rfloor - \ell)^\ell}{\ell!} \geq \frac{(m/(2p))^\ell}{\ell!}$$

(for sufficiently large m) sets of size ℓ in total, and each one of them is chosen to be S_i by the adversary with equal probability, the likelihood that $\mathcal{A}$ h-covered S_i is at most

$$\frac{2^t cm^\ell \ell!}{p^{\ell-(h+1)}(m/(2p))^\ell} = 2^{t+\ell} \ell! p^{h+1} c = \frac{\delta}{p}.$$

It follows from a union bound over all p of the parties P_i that the probability that $\mathcal{A}$ h-covered *any* of the S_i is at most δ. For $\mathcal{A}$ to satisfy JR, it is necessary for it to either h-cover some S_i with the initial queries or subsequently select every $c^i_{s^*}$ after having failed to h-cover any S_i. By the union bound, the probability that $\mathcal{A}$ satisfies JR is at most the sum of the probabilities of these two events, which is at most $\left(\frac{k_0}{\ell}\right)^p + \delta$. □

Thus, to prove lower bounds against non-adaptive algorithms with exact queries, it suffices to construct probability distributions over subsets of a finite set $[\ell]$ with certain special properties. To prove our $\Omega(m^2)$ lower bound, which holds for any $k \geq 2$, we generalize the construction from Figure 1 (a) by simply adding more candidates to the larger approval set:

Theorem 3.1. *For any constants k and t such that $k \geq 2$, and any $\varepsilon > 0$, any non-adaptive (k,t)-committee selection algorithm that makes fewer than $\Omega(m^2)$ queries satisfies JR with probability at most ε.*

Proof. Given any t and $\varepsilon \in (0,1]$, let $h = 1$, $k_0 = 2$, and $\ell = \lceil 4/\varepsilon \rceil$. Consider the probability distribution over subsets of $[\ell]$ where $x_{\{1\}} = \frac{1}{2}$, $x_{\{2,3,4,\dots,\ell\}} = \frac{1}{2}$, and all other sets have probability zero (Figure 1 (a) shows the special case of this distribution where $\ell = 3$). Notice that these parameters meet all the requirements of Lemma B.1 with $s^* = 1$. Letting $\delta := \varepsilon/2$, it follows that, for any $k \geq k_0 = 2$, any (k,t)-committee selection algorithm that makes fewer than $\Omega(m^2)$ queries satisfies JR with probability at most

$$\left(\frac{k_0}{\ell}\right)^{\lfloor k/k_0 \rfloor} + \delta \leq \left(\frac{k_0}{\ell}\right)^1 + \delta = \frac{2}{\ell} + \delta \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

□

To prove stronger lower bounds we need to increase the h parameter. Probability distributions $\{x_S\}_{S\subseteq[\ell]}$ satisfying the hypotheses of Lemma B.1 prove difficult to construct by hand for $h > 1$, so we conducted a computational search. By a straightforward averaging argument, one can see that it is without loss of generality to consider "symmetric" distributions, where for any sets $S, T \subseteq [\ell]$ of the same size that either both contain s^* or both do not contain s^*, $x_S = x_T$. Thus, it suffices to consider solutions encoded as points in the following polyhedron, which we refer to as $P(h, k_0, \ell) \subseteq \mathbb{R}^{2\ell}$. We parameterize the space by the 2ℓ variables

$$\{x_{i,j} \mid i \in \{0,1\}, j \in \{0,1,2,\ldots,\ell-1\}\},$$

where $x_{0,j}$ encodes the value of x_S for all S of size j that do not include s^*, and $x_{1,j}$ encodes the value of x_S for all S containing s^* and j other elements from $[\ell]$. For a solution to be in $P(h, k_0, \ell)$, there are four kinds of constraints it must satisfy.

- All probabilities must be nonnegative: for all $i \in \{0,1\}$ and $j \in \{0,1,2,\ldots,\ell-1\}$,

$$x_{i,j} \geq 0.$$

- Probabilities must all sum to 1:

$$\sum_{i=0}^{1}\sum_{j=0}^{\ell-1}\binom{\ell-1}{j}x_{i,j} = 1.$$

- Condition (1) from Lemma B.1 must be satisfied. Due to the symmetry that is baked in to the solutions we're considering, we only need to check the constraint for pairs of sets where s^* is contained in one set but not the other. This constraint is as follows: for all $t' \in [h]$,

$$\sum_{i=0}^{1}\sum_{j=t'}^{\ell-1}\binom{\ell-1-t'}{j-t'}x_{i,j} = \sum_{j=t'-1}^{\ell-1}\binom{\ell-t'}{j-t'+1}x_{1,j}.$$

- Condition (2) from Lemma B.1 must be satisfied:

$$x_{1,0} \geq \frac{1}{k_0}.$$

Thus, there exists a probability distribution satisfying the hypotheses of Lemma B.1 if and only if $P(h, k_0, \ell)$ is nonempty. The description of this polyhedron is only of polynomial-size, so we can solve it efficiently using linear programming. However, many of the coefficients are extremely large, and we eventually ran into numerical difficulties. Table 1 lists the tightest lower bounds we were able to obtain in terms of how large k_0 had to be for a given value of h, and Table 2 provides one point in $P(10, 72, 73)$ as an example.

h	1	2	3	4	5	6	7	8	9	10
k_0	2	6	10	16	21	30	38	49	59	72

Table 1: For each positive integer h, smallest value of k_0 for which $P(h, k_0, k_0+1)$ is nonempty, i.e., the smallest committee size for which Lemma B.1 implies that guaranteeing JR requires $\Omega(m^{h+1})$ exact queries of size $t > h$. The constraints are tight only for $h \in \{1, 2\}$.

$x_{1,0}$	$x_{0,2}$	$x_{1,5}$	$x_{0,13}$	$x_{1,20}$	$x_{0,31}$	$x_{1,41}$	$x_{0,52}$	$x_{1,59}$	$x_{0,67}$	$x_{1,70}$
0.01398	3.204e-4	1.184e-9	1.012e-15	4.781e-20	7.926e-23	5.799e-23	1.875e-20	2.058e-16	8.968e-11	4.577e-6

Table 2: The point in $P(10, 72, 73)$ maximizing $x_{1,0}$. All variables not shown in the table have value zero.

Theorem 3.3. *For any $\varepsilon > 0$, there exists a target committee size k with $k = \Theta(\log 1/\varepsilon)$ such that for all t, any non-adaptive (k, t)-committee selection algorithm with exact queries that makes fewer than $\Omega(m^{11})$ queries satisfies JR with probability at most ε.*

Proof. Let $\varepsilon > 0$ be given. Then let

$$k := 72\left(\frac{\log(2/\varepsilon)}{\log(73/72)} + 1\right)$$

and $\delta := \varepsilon/2$. As Tables 1 and 2 show, Lemma B.1 holds for $h = 10$, $k_0 = 72$, and $\ell = 73$. Thus, for any t, any non-adaptive (k, t)-committee selection algorithm that makes fewer than $\Omega(m^{11})$ queries satisfies JR with probability at most

$$\left(\frac{72}{73}\right)^{\lfloor k/72 \rfloor} + \delta \leq \left(\frac{72}{73}\right)^{(k/72)-1} + \delta = \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

□

We note that there is a gap between these results and Theorem 3.4. An intriguing direction for future work is to obtain matching upper and lower bounds for the query complexity of guaranteeing JR using a non-adaptive algorithm with exact queries. For $k = 1$ we need $\Theta(m)$ queries, and for $k \in \{2, 3\}$, we need $\Theta(m^2)$ queries. However, the complexity is unknown for all larger k, and we conjecture that the exponent of m grows as a polynomial function of k.

C Family of examples for noisy vs exact queries

Fix some $k \geq 4$, t, and m. We will construct a family of instances on m candidates where there exists a non-adaptive exact-query algorithm which can guarantee JR using $\lceil m/t \rceil$ queries while a non-adaptive noisy-query algorithm necessarily needs $\Omega(m \log(m)/t)$ to guarantee it with any fixed probability δ. We describe the approval profile by the distribution over approval sets by sampling a voter uniformly at random. There is one special candidate a^*. This candidate a^* is approved by a $2/k$ fraction of the electorate while all other candidates b are approved by $1/(2k)$. Further, these approvals are independent in the sense that when we sample a voter, the joint distribution over approvals is as if each of these approvals were selected independently. For example, given a set $S \subseteq C$ of candidates such that S contains a^* and ℓ other candidates, the proportion of voters who approve exactly the set S is $(2/k) \cdot (1/(2k))^{\ell} \cdot (1 - 1/(2k))^{m-1-\ell}$.

The first observation we make is that the committees that satisfy JR are exactly those that include a^*. Notice that if a committee includes a^*, no other candidate has enough approval support $(1/k)$ to have a blocking coalition to violate JR. On the other hand, if there is a committee W of size k that does not include a^*, we can compute the proportion of voters that approve a^* that do not approve of any candidates in W. This is

$$\frac{2}{k} \cdot \left(1 - \frac{1}{2k}\right)^k > \frac{2}{k}\left(1 - \frac{k}{2k}\right) = \frac{1}{k}.$$

Hence, for such a W, there would exist a sufficiently large blocking coalition for a^*.

Next, we show that there is a non-adaptive exact-query algorithm that can guarantee JR for any instance of this form (i.e., regardless of which candidate is a^*). Indeed, it simply makes $\lceil m/t \rceil$ queries that cover all candidates. From this, it can deduce candidate approval scores and ensure that the committee it chooses contains the candidate with approval score $2/k$.

Finally, let us consider a non-adaptive algorithm that makes ℓ queries and guarantees JR with probability $1 - \delta$ regardless of which candidate is a^*. Notice that such an algorithm should guarantee JR with this same probability against a distribution of instances where a^* is selected uniformly at random. Let us consider an algorithm A that maximizes the probability of selecting a JR committee against this distribution. Notice that it is without loss of generality that A is deterministic by Yao's minimax principle. We show that this can only be done if $\ell \geq f(m)$ where $f(m) \in \Omega(m \log m)$ (treating k, t, and δ as constants) is a function to be defined later.

Suppose for a contradiction $\ell < f(m)$. Let H be the set of candidates that appear in strictly more than $q := \frac{2tf(m)}{m\delta}$ and let L be the remaining candidates. Notice that $|H| \leq \delta/2 \cdot m$, as otherwise $\ell \geq f(m)$. We show that conditioned on $a^* \in L$, the probability A chooses a committee containing a^* is at most $\delta/2$. This implies that A's probability of success is at most $(1 - \delta/2) \cdot \delta/2 + \delta/2 < \delta$.

To that end, consider an algorithm that receives extra queries such that all candidates in L are in exactly q queries. Notice that conditioned on a^* being in L, since all candidates in L are in the same number of queries, the optimal strategy to maximize the probability a^* is a committee-member is to take the k candidates in L with highest empirical approval score. Indeed, this dominates any other strategy as conditioned on any empirical approval scores, this choice of committee covers the maximum likelihood estimates of the underlying distribution.

What we finally show is that with probability at least $1 - \delta/2$, conditioned on $a^* \in L$, a^* will *not* be among the k highest approval scores. Intuitively, with reasonably high probability a^* will have empirical not too much more than it's true approval, say at most $3/k$, while, by choosing $q \in O(\log m)$, due to the noise in estimating empirical approval scores, at least k of the remaining candidates in L will have approval score this large. Indeed, ensuring $q > \frac{k^2 \log(4/\delta)}{2}$ ensures the empirical estimate of a^* is less than $3/k$ with probability at least $1 - \delta/4$ using standard Hoeffding's inequality. For the other emprical means, using tail bounds on the Binomial distribution (Ash 1990), the probability they are at least $3/k$ is at least $\frac{1}{\sqrt{2q}} \exp(-q\Theta(k))$. Notice we can choose $q \in \Theta(\log m)$ such that this value is at least $2(k + \log(4/\delta))/m$. For sufficiently large m, this choice of q will be above $\frac{k^2 \log(4/\delta)}{2}$, leading to a valid $\Theta(m \log m)$ function of f. Further, again applying standard Hoeffding's inequality on the remaining at least $m/2$ candidates in L shows that at least k will satisfy this, as needed.

D Proofs for the Adaptive Exact-Query Setting

In the following, we prove Lemma 3.5, Lemma 3.6, and Theorem 3.4.

D.1 Proof of Lemma 3.5

Lemma 3.5 and its proof are based on the lower bound from Lemma 1 in (Skowron 2021). Our result is more general in two ways: (1) our statement holds for any committee W, no matter what algorithm computed it, and (2) we introduce an approximation parameter α. We begin with the following intermediate lemma:

Lemma D.1. *For any committee $W \subseteq C$ and group of voters $V \subseteq N$, we have*

$$avs_W(V) \geq \min\left\{\left|\bigcap_{i\in V} A_i\right|, \frac{1}{n}\cdot\frac{|V|}{\Delta^*(W)} - 1\right\}.$$

Proof. As mentioned, the following proof is closely related to the proof of Lemma 1 of Skowron (2021). Suppose there exist V and W such that both

$$\frac{1}{|V|}\sum_{i\in V}|W\cap A_i| < \left|\bigcap_{i\in V} A_i\right| \text{ and } \frac{1}{|V|}\sum_{i\in V}|W\cap A_i| < \frac{1}{n}\cdot\frac{|V|}{\Delta^*(W)} - 1.$$

We then have

$$\begin{aligned}\left|\bigcap_{i\in V} A_i\right| &> \frac{1}{|V|}\sum_{i\in V}|W\cap A_i| \\ &\geq \frac{1}{|V|}\sum_{i\in V}\left|W\cap\left(\bigcap_{j\in V} A_j\right)\right| \\ &= \left|W\cap\left(\bigcap_{j\in V} A_j\right)\right|.\end{aligned}$$

This implies $W\cap\left(\bigcap_{i\in V} A_i\right) \subsetneq \bigcap_{i\in V} A_i$, so $\overline{W}\cap\left(\bigcap_{i\in V} A_i\right) \neq \emptyset$. Hence, there is a candidate $c \in \overline{W}\cap\left(\bigcap_{i\in V} A_i\right)$ that is not on the committee W, but is approved by all voters in V. For such a candidate c, we have

$$\begin{aligned}\Delta(W,c) &= \frac{1}{n}\sum_{i\in N: c\in A_i\setminus W}\frac{1}{|A_i\cap W|+1} \\ &= \frac{1}{n}\sum_{i\in N: c\in A_i}\frac{1}{|A_i\cap W|+1} && (c\notin W) \\ &\geq \frac{1}{n}\sum_{i\in V}\frac{1}{|A_i\cap W|+1} && (c\in\textstyle\bigcap_{i\in V} A_i) \\ &\geq \frac{1}{n}|V|\frac{1}{\frac{1}{|V|}\sum_{i\in V}(|W\cap A_i|+1)} && \text{(convexity of } 1/x) \\ &> \frac{1}{n}|V|\frac{1}{\frac{1}{n}\frac{|V|}{\Delta^*(W)}-1+1} \\ &= \Delta^*(W),\end{aligned}$$

a contradiction, as $\Delta(W,x)\leq\Delta^*(W)$ for all candidates x. □

We are now ready to prove Lemma 3.5, which we restate here:

Lemma 3.5. *If a committee W satisfies $\Delta^*(W) < \frac{1}{\alpha k}$, then W satisfies α-EJR and α-proportionality.*

Proof. Fix a committee W satisfying $\Delta^*(W) < \frac{1}{\alpha k}$. We begin with α-proportionality. Fix a $\lambda \in [0,k]$, and a $\frac{\lambda+1}{\alpha}$-large, λ-cohesive group of voters V. By definition of λ-cohesive, $\left|\bigcap_{i\in V} A_i\right| \geq \lambda$. Further, we have

$$\begin{aligned}\frac{1}{n}\cdot\frac{|V|}{\Delta^*(W)} - 1 &\geq \frac{1}{n}\cdot\frac{1}{\Delta^*(W)}\cdot\frac{\lambda+1}{\alpha}\cdot\frac{n}{k} - 1 && (V \text{ is } \tfrac{\lambda+1}{\alpha}\text{-large}) \\ &= \frac{1}{\Delta^*(W)}\cdot\frac{\lambda+1}{\alpha}\cdot\frac{1}{k} - 1 \\ &> \lambda + 1 - 1 && (\Delta^*(W) < \tfrac{1}{\alpha k}) \\ &= \lambda\end{aligned}$$

Together, these imply that

$$\min\left\{\left|\bigcap_{i\in V} A_i\right|, \frac{1}{n}\cdot\frac{|V|}{\Delta^*(W)} - 1\right\} \geq \lambda.$$

Invoking Lemma D.1, we have $\text{avs}_W(V) \geq \lambda$, as needed.

Next, we show α-EJR. Fix an $\ell \in [k]$, and an $\frac{\ell}{\alpha}$-large, ℓ-cohesive group of voters V. As before, by the definition of ℓ-cohesive, we have $\left|\bigcap_{i \in V} A_i\right| \geq \ell$. Further, by the same argument as above with $\ell = \lambda + 1$,

$$\frac{1}{n} \cdot \frac{|V|}{\Delta^*(W)} - 1 > \ell - 1.$$

Together, these imply that

$$\min\left\{\left|\bigcap_{i \in V} A_i\right|, \frac{1}{n} \cdot \frac{|V|}{\Delta^*(W)} - 1\right\} > \ell - 1.$$

Invoking Lemma D.1, we have $\text{avs}_W(V) > \ell - 1$, and since utilities are integers, this implies that $|A_i \cap W| \geq \lceil \text{avs}_W(V) \rceil \geq \ell$ for at least one voter $i \in V$, as needed. □

D.2 Proof of Lemma 3.6

Lemma 3.6 here:

Lemma 3.6. *For any committee W and $c \notin W$, we have that $\max_{x \in W} \Delta(W, c, x) \geq \frac{(k+1)\Delta(W,c)-1}{k}$. In particular, if $\Delta(W, c) \geq \frac{1}{\alpha k}$, then $\max_{x \in W} \Delta(W, c, x) \geq \frac{(1-\alpha)k+1}{\alpha k^2}$.*

Proof. Fix W and $c \notin W$. We will use the notation $W^+ := W \cup \{c\}$. First, we show that

$$\min_{x \in W} \Delta(W^+ \setminus \{x\}, x) \leq \frac{1 - \Delta(W, c)}{k}. \tag{1}$$

To that end, let us consider $\Delta(W^+ \setminus \{x\}, x)$ for an arbitrary $x \in W^+$. We have

$$\begin{aligned}\Delta(W^+ \setminus \{x\}, x) &= \text{PAV-SC}(W^+) - \text{PAV-SC}(W^+ \setminus \{x\}) \\ &= \frac{1}{n} \sum_{i \in N : x \in A_i} \frac{1}{|W^+ \cap A_i|}.\end{aligned}$$

Adding up over all $x \in W^+$, we have

$$\begin{aligned}\sum_{x \in W^+} \Delta(W^+ \setminus \{x\}, x) &= \frac{1}{n} \sum_{x \in W^+} \sum_{i \in N : x \in A_i} \frac{1}{|W^+ \cap A_i|} \\ &= \frac{1}{n} \sum_{i : W^+ \cap A_i \neq \emptyset} \frac{|W^+ \cap A_i|}{|W^+ \cap A_i|} \\ &\leq 1.\end{aligned}$$

On the other hand, we have

$$\begin{aligned}\sum_{x \in W^+} \Delta(W^+ \setminus \{x\}, x) &= \Delta(W, c) + \sum_{x \in W} \Delta(W^+ \setminus \{x\}, x) \\ &\geq \Delta(W, c) + k \cdot \min_{x \in W} \Delta(W^+ \setminus \{x\}, x).\end{aligned}$$

Combining these two inequalities, we get that

$$\Delta(W, c) + k \cdot \min_{x \in W} \Delta(W^+ \setminus \{x\}, x) \leq 1.$$

Rearranging yields (1). Finally, notice that

$$\max_{x \in W} \Delta(W, c, x) = \Delta(W, c) - \min_{x \in W} \Delta(W^+ \setminus \{x\}, x).$$

Hence, by (1),

$$\max_{x \in W} \Delta(W, c, x) \geq \Delta(W, c) - \left(\frac{1 - \Delta(W, c)}{k}\right) = \frac{(k+1)\Delta(W, c) - 1}{k},$$

as needed. □

D.3 Proof of Theorem 3.4

Theorem 3.4. *For any $m \geq t > k$, Algorithm 1 yields a committee satisfying α-proportionality and α-EJR while making at most*

$$\left\lceil \frac{m-k}{t-k} \right\rceil \frac{\alpha k^2}{(1-\alpha)k+1} H_k$$

queries, where H_k is the k^{th} harmonic number. For $\alpha = 1$, this leads to a query complexity of $\mathcal{O}\left(mk^2 \log k\right)$ while for any fixed $\alpha < 1$, this leads to a query complexity of $\mathcal{O}\left(mk \log k\right)$.

Proof. Clearly, if Algorithm 1 terminates, the resulting committee W satisfies

$$\Delta^*(W) = \Delta(W, c') < \frac{1}{\alpha k}$$

and hence α-EJR and α-proportionality by Lemma 3.5. What remains is to bound how many steps the algorithm takes to terminate. To do this, we use the PAV score of the current committee as a potential function. In every iteration of the loop for which the algorithm does not terminate, we have

$$\Delta(W, c') \geq \frac{1}{\alpha k}$$

and hence, by Lemma 3.6, the increase in PAV score at each step will be

$$\max_x \Delta(W, c', x) \geq \frac{(1-\alpha)k+1}{\alpha k^2}.$$

Notice that the minimum and maximum PAV score that can be possibly attained by any committee of size k are 0 (when nobody approves of any candidate) and H_k (the harmonic number which is attained when everyone approves of every candidate) respectively. Hence, there can be at most

$$\frac{\alpha k^2}{(1-\alpha)k+1} H_k$$

steps. Since at each step we make $\left\lceil \frac{m-k}{t-k} \right\rceil$ queries, the result follows. □

E PAV and LS-PAV Yield a Committee That Satisfies $\Delta^*(W) < 1/k$

As mentioned previously, it is known that PAV and LS-PAV satisfy both EJR and proportionality. Here, we show that they yield committees that satisfy $\Delta^*(W) < 1/k$, which implies EJR and proportionality through Lemma 3.5. This is noteworthy because 1) it has a much simpler proof, 2) it implies that PAV and LS-PAV committees can be certified in a computationally efficient manner, by verifying that $\Delta^*(W) < 1/k$.

Lemma E.1. *For both PAV and LS-PAV, the returned committee W satisfies $\Delta^*(W) < 1/k$.*

Proof. For the committee W computed by PAV or LS-PAV, we have that for any candidate $c \notin W$,

$$\max_{x \in W} \Delta(W, c, x) < \frac{1}{k^2}$$

since otherwise the PAV score of W could be improved by at least $1/k^2$ by adding c and removing the worst candidate. By Lemma 3.6, this implies that

$$\frac{(k+1)\Delta^*(W) - 1}{k} < \frac{1}{k^2}.$$

Rearranging yields $\Delta^*(W) < \frac{1}{k}$, so the result follows from Lemma 3.5. □

F Noisy Query Model Discussion

Note that a voter may be queried more than once during the run of the algorithm. One could instead assume that voters arrive in a random order, but we prefer our model for two reasons. First, it can be justified by assuming that voter preferences are drawn i.i.d. from an underlying population distribution, and we want our committee to satisfy properties with respect to this distribution. The downside of a model that considers a distribution over approval sets rather than a finite set of voters is that it would require nonstandard definitions of committee properties, i.e., with respect to a distribution rather than a profile. Hence, our definition essentially treats the profile as the population distribution while still allowing the use of standard definitions. Second, our model allows us to decouple the query complexity from the number of voters in the profile. Finally, we believe that a different choice of model would lead to qualitatively similar results. In particular, if the total number of voters is substantially larger than the number of queried voters, sampling with replacement approaches sampling without replacement.

G Proof of Theorem 4.1

Theorem 4.1. *For any $m \geq t > k$, with probability at least $1-\delta$, Algorithm 2 returns a committee that satisfies α-EJR and α-proportionality after querying no more than*

$$578 H_k \left\lceil \frac{m-k}{t-k} \right\rceil \left(\frac{\alpha k^2}{(1-\alpha)k+1} \right)^3 \log\left(\frac{4mk^4}{\delta} \right)$$

voters. For any fixed $\delta > 0$, if $\alpha = 1$, this leads to a query complexity of $\mathcal{O}\left(mk^6 \log k \log m\right)$ and if $\alpha < 1$, this leads to a query complextiy of $\mathcal{O}\left(mk^3 \log k \log m\right)$.

Proof. We first show that with probability at least $1-\delta$, all $\hat{\Delta}$ estimates in the first $H_k \cdot \frac{2\alpha k^2}{(1-\alpha)k+1}$ iterations of the loop are within $\pm\varepsilon := \frac{(1-\alpha)k+1}{12\alpha k^2}$ of the corresponding true Δ values. Then, we show that conditioned on these accurate estimates, the algorithm satisfies the theorem properties.

To show the error bounds, we will use a straightforward application of Hoeffding's inequality. Indeed, when the corresponding ℓ voters are sampled, notice that $\hat{\Delta}$ is simply the sample mean of independent samples with expectation of the corresponding Δ. Further, these samples are always proportions falling in $[-1, 1]$. Hence, any specific estimate will not be within $\pm\varepsilon$ with probability at most $2\exp(-\varepsilon^2\ell/2)$. Note that there are $m-k$ choices of x for $\hat{\Delta}(W, x)$ and $(m-k)\cdot k$ choices of (x, y) pairs for $\hat{\Delta}(W, x, y)$. Hence, there are a total of $(m-k)(k+1)$ estimates per iteration. Therefore, there are at most $H_k \cdot \frac{2\alpha k^2}{(1-\alpha)k+1}(m-k)(k+1)$ estimates in the first $H_k \cdot \frac{2\alpha k^2}{(1-\alpha)k+1}$ iterations. A union bound tells us the probability that all estimates in these iterations are within $\pm\varepsilon$ is at least

$$1 - 2\exp(-\varepsilon^2\ell/2) \cdot H_k \cdot \frac{2\alpha k^2}{(1-\alpha)k+1}(m-k)(k+1).$$

We simply need to show that this value is at least $1-\delta$.

To that end, recall that

$$\ell = \left\lceil 288 \left(\frac{\alpha k^2}{(1-\alpha)k+1} \right)^2 \log\left(\frac{8mk^4}{\delta} \right) \right\rceil.$$

Noting that $H_k \leq k$ and $\frac{\alpha k^2}{(1-\alpha)k+1} \leq k^2$, we have that

$$8mk^4 \geq 2k \cdot (2k^2) \cdot m \cdot (2k) \geq 2H_k \cdot \frac{2\alpha k^2}{(1-\alpha)k+1}(m-k)(k+1).$$

Hence,

$$\ell \geq \frac{2}{\varepsilon^2} \log\left(\frac{2H_k \cdot \frac{2\alpha k^2}{(1-\alpha)k+1}(m-k)(k+1)}{\delta} \right),$$

so we have

$$2\exp\left(-\varepsilon^2\ell/2\right) \leq \frac{\delta}{H_k \cdot \frac{2\alpha k^2}{(1-\alpha)k+1}(m-k)(k+1)},$$

as needed.

Next, condition on all of these estimates being accurate. Notice if the algorithm terminates within the first $H_k \cdot \frac{2\alpha k^2}{(1-\alpha)k+1}$ iterations, this means that for the returned committee, $\max_{x \notin W} \hat{\Delta}(W, x) < \frac{1}{\alpha k} - \frac{(1-\alpha)k+1}{8\alpha k^2} = \frac{1}{\alpha k} - \varepsilon$. By our assumption about the accuracy of each $\hat{\Delta}$, we have that $\Delta^*(W) = \max_{x \notin W} \hat{\Delta}(W, x) < \frac{1}{\alpha k}$. Hence, by Lemma 3.5, W satisfies the desired properties. Further, there are $\left\lceil \frac{m-k}{t-k} \right\rceil \cdot \ell$ queries per iteration. Noting that

$$\ell \leq 289 \left(\frac{\alpha k^2}{(1-\alpha)k+1} \right)^2 \log\left(\frac{8mk^4}{\delta} \right)$$

to avoid the ceiling, this means the total query complexity is at most

$$H_k \cdot \frac{2\alpha k^2}{(1-\alpha)k+1} \cdot \left\lceil \frac{m-k}{t-k} \right\rceil \cdot \ell \leq 578 H_k \left\lceil \frac{m-k}{t-k} \right\rceil \left(\frac{\alpha k^2}{(1-\alpha)k+1} \right)^3 \log\left(\frac{8mk^4}{\delta} \right)$$

as needed.

What remains to be shown is that conditioned on the accurate estimates, the algorithm terminates within $H_k \cdot \frac{2\alpha k^2}{(1-\alpha)k+1}$ iterations. Indeed, we show that each iterations, the PAV score of W increases by at least $\frac{(1-\alpha)k+1}{2\alpha k^2}$. As the minimum and maximum PAV scores of a committee are 0 and H_k respectively, this can occur at most $H_k \cdot \frac{2\alpha k^2}{(1-\alpha)k+1}$ times. Hence, we obtain the desired bound on the number of iterations.

To that end, note that when when we make a swap of c' for c, it must be the case that $\hat{\Delta}(W,c) > \frac{1}{\alpha k} - \varepsilon$. Using our assumptions on $\hat{\Delta}$ errors, this implies that $\Delta(W,c) \geq \frac{1}{\alpha k} - 2\varepsilon$. By Lemma 3.6, we have that

$$\max_{x \in W} \Delta(W,c,x) \geq \frac{(1-\alpha)k+1}{\alpha k^2} - \frac{k+1}{k} 2\varepsilon \geq \frac{(1-\alpha)k+1}{\alpha k^2} - 4\varepsilon.$$

Again, by our assumption on $\hat{\Delta}$ errors,

$$\max_{x \in W} \hat{\Delta}(W,c,x) \geq \max_{x \in W} \Delta(W,c,x) - \varepsilon \geq \frac{(1-\alpha)k+1}{\alpha k^2} - 5\varepsilon.$$

Finally, for the choice c' that maximizes $\hat{\Delta}(W,c,c')$,

$$\Delta(W,c,c') \geq \hat{\Delta}(W,c,c') - \varepsilon \geq \frac{(1-\alpha)k+1}{\alpha k^2} - 6\varepsilon = \frac{(1-\alpha)k+1}{2\alpha k^2},$$

as needed. □

H Description and Analysis of Algorithm 4

In this section, we state Algorithm 4 and analyze its complexity. The worst-case query guarantees are slightly worse than those of Algorithm 2; however, as we discuss below, there are instances where Algorithm 4 performs better, and this can additionally be seen in the experiments of Section 5.

Theorem H.1. *For any $m \geq t > k$, with probability $1-\delta$, Algorithm 4 yields a committee satisfying α-proportionality and α-EJR after querying at most*

$$1152 H_k \left\lceil \frac{m-k}{t-k} \right\rceil \left(\frac{\alpha k^2}{(1-\alpha)k+1} \right)^3 \log\left(\frac{4608 k^8 m^{k+2}}{\delta} \right)$$

voters. For any fixed $\delta > 0$, if $\alpha = 1$, this leads to a query complexity of $\mathcal{O}\left(mk^7 \log k \log m\right)$ and if $\alpha < 1$, this leads to a query complexity of $\mathcal{O}\left(mk^4 \log k \log m\right)$.

We use

$$L := 1152 H_k \left\lceil \frac{m-k}{t-k} \right\rceil \left(\frac{\alpha k^2}{(1-\alpha)k+1} \right)^3 \log\left(\frac{4608 k^8 m^{k+2}}{\delta} \right)$$

to denote this upper bound (notice that it is the same as the L from the algorithm).

Importantly, this theorem states that despite the extensions we introduced in Algorithm 4, it remains theoretically sound: with a sufficient number of samples, it yields a committee satisfying proportionality and EJR.

The proof follows a relatively similar structure to Theorem 4.1: we first show that with probability $1-\delta$, many estimates are sufficiently accurate, and conditioned on this, the algorithm makes progress in terms of PAV score and terminates with a good committee. However, unlike Theorem 4.1, the samples we take are not fresh for each round, so we can not directly apply Hoeffding's inequality in the most straightforward way. Nonetheless, the proof goes through by instead treating the Δ estimates as Martingales in order to use Azuma's inequality. Due to its additional intricacy, we separate this portion into its own lemma.

Lemma H.2. *With probability $1-\delta$, at every step after querying up to L voters,*

$$\Delta(W,x) \leq \widetilde{\Delta}^{+}(W,x) \leq \Delta(W,x) + 2err_k(W,x)$$

and

$$\Delta(W,x,y) - 2err_k(W,x,y) \leq \widetilde{\Delta}^{-}(W,x,y) \leq \Delta(W,x,y)$$

for all committees W, $x \notin W$, and $y \in W$.

Proof. We begin by considering estimates of the form $\widetilde{\Delta}^{-}(W,x,y) \leq \Delta(W,x,y)$; the rest of the estimates will follow similar arguments which we discuss later. Fix an arbitrary committee W, $x \notin W$, $y \in W$, and $s \in [k]$. We define a sequence of random variables $X_0, X_1, \ldots$ where X_j is the unnormalized estimate $|V_s(W,x,y)| \cdot \hat{\Delta}_s^{-}(W,x,y)$ when $|V_s(W,x,y)| = j$, i.e., when j voters have been queried on x, y and at least s candidates of W, and $X_0 = 0$. In other words, when the j^{th} voter of $V_s(W,x,y)$ is queried, X_j is X_{j-1} plus that j^{th} voters estimate for $\Delta(W,x,y)$, $\frac{\mathbb{I}[x \in R_i \text{ and } y \notin R_i]}{|R_i \cap W| + |W \setminus Q_i| + 1} - \frac{\mathbb{I}[x \notin R_i \text{ and } y \in R_i]}{|R_i \cap W|}$.

Algorithm 4: (k,t)-ucb-α-PAV

Choose $W \in \binom{C}{k}$ arbitrarily
$\mathcal{Q} \leftarrow \{\}$ ▷ List to store queries and responses
$\ell \leftarrow 576 \cdot \left(\frac{\alpha k^2}{(1-\alpha)k+1}\right) \log\left(\frac{4608k^8 m^{k+2}}{\delta}\right)$ ▷ Constant to be used later
$L \leftarrow 2H_k \left\lceil \frac{m-k}{t-k} \right\rceil \left(\frac{\alpha k^2}{(1-\alpha)k+1}\right) \cdot \ell$ ▷ Constant to be used later
for $i = 1, 2, \ldots$ **do**
    $V_s(W,x) \leftarrow \{i \mid x \in Q_i \text{ and } |Q_i \cap W| \geq s\}$ ▷ $\forall s \in \{0\} \cup [k], \forall x \notin W,$
    $V_s(W,x,y) \leftarrow \{i \mid \{x,y\} \subseteq Q_i \text{ and } |Q_i \cap W| \geq s\}$ ▷ $\forall s \in [k], \forall x \notin W, \forall y \in W$
    $\hat{\Delta}_s^+(W,x) \leftarrow \frac{1}{|V_s|} \sum_{i \in V_s(W,x)} \frac{\mathbb{I}[x \in R_i]}{|R_i \cap W|+1}$ if $V_s(W,x) \neq \emptyset$ else ∞
        ▷ Upper bound on estimate for $\Delta(W,x)$ using voters queried on at least s candidates of W (along with x)
    $\hat{\Delta}_s^-(W,x,y) \leftarrow \frac{1}{|V_s(W,x,y)|} \sum_{i \in V_s(W,x,y)} \frac{\mathbb{I}[x \in R_i \text{ and } y \notin R_i]}{|R_i \cap W|+|W \setminus Q_i|+1} - \frac{\mathbb{I}[x \notin R_i \text{ and } y \in R_i]}{|R_i \cap W|}$ if $V_s(W,x,y) \neq \emptyset$ else $-\infty$
        ▷ Lower bound on estimate for $\Delta(W,x,y)$ using voters queried on at least s candidates of W (along with x and y)
    $err_s(W,x) \leftarrow \sqrt{\frac{2\log\left(\frac{4L(k+1)m^{k+1}}{\delta}\right)}{|V_s(W,x)|}}$
    $err_s(W,x,y) \leftarrow \sqrt{\frac{2\log\left(\frac{4L(k+1)m^{k+1}}{\delta}\right)}{|V_s(W,x,y)|}}$
    $\widetilde{\Delta}^+(W,x) \leftarrow \min_{s \in [k]} \hat{\Delta}_s^+(W,x) + err_s(W,x)$ ▷ Best UCB-style upper bound on $\Delta(W,x)$ given queries
    $\widetilde{\Delta}^-(W,x,y) \leftarrow \max_{s \in [k]} \hat{\Delta}_s^-(W,x,y) - err_s(W,x,y)$ ▷ Best UCB-style lower bound on $\Delta(W,x,y)$ given queries
    $c' \leftarrow \arg\max_{x \notin W} \Delta^+(W,x)$
    if $\widetilde{\Delta}^+(W,c') < \frac{1}{\alpha k}$ **then**
        return W
    $c \leftarrow \arg\max_{x \in W} \widetilde{\Delta}^-(W,c',x)$
    if $\widetilde{\Delta}^-(W,c',c) \geq \frac{1}{2}\frac{(1-\alpha)k+1}{\alpha k^2}$ **then**
        $W \leftarrow (W \cup \{c'\}) \setminus \{c\}$
    else
        $A \leftarrow \{x \in C \mid |\{i \mid W \cup \{x\} \subseteq Q_i\}| \geq \ell\}$ ▷ Candidates already queried more than ℓ times with W
        $S \leftarrow C \setminus W \setminus A$ ▷ Potential candidates to query along with W
        Make query Q_i on W and $t-k$ candidates $x \notin S$ with highest $\Delta^+(W,x)$, breaking ties arbitrarily
        Receive response R_i and append (i, Q_i, R_i) to $\mathcal{Q}$

Notice that when the j^{th} voter is queried, regardless of the algorithm's choices of when to make such a query, this is simply a random voter from the population chosen independently of everything else. Hence, if their entire approval set was known, the expectation of their estimate of $\Delta(W,x,y)$ would be exactly $\Delta(W,x,y)$. When only part W intersects the query, we choose a bound that would always upper bounds the true estimate. Therefore, $\mathbb{E}[X_j \mid X_{j-1}] \geq X_{j+1} + \Delta(W,x,y)$.

Let $Y_0, Y_1, \ldots$ be the additive errors of X_j from the true $\Delta(W,x,y)$, that is, $Y_j = X_j - j \cdot \Delta(W,x,y)$. The key observation we will make is that the sequence $Y_0, Y_1, Y_2, \ldots$ is, in fact, a submartingale. Indeed, since $Y_j = X_j - j \cdot \Delta(W,x,y)$ and $Y_{j-1} = X_{j-1} - (j-1) \cdot \Delta(W,x,y)$, we have $\mathbb{E}[Y_j \mid Y_{j-1}] \geq 0$.

Additionally, note that an individual voter's Δ estimate is always within $[-1,1]$, so $X_j - X_{j-1} \in [-1,1]$. Using the definition of Y_j, we have that this implies $Y_j - Y_{j-1} \in [-1-\Delta(W,x,y), 1-\Delta(W,x,y)]$. Note that this is a range of size 2, and we can hence use (the asymmetric version of) Azuma's inequality to get that for all $\varepsilon > 0$,

$$\Pr[Y_j \leq -\varepsilon] = \Pr[Y_j - Y_0 \leq -\varepsilon] \leq \exp\left(-\frac{2\varepsilon}{j \cdot 2^2}\right) = \exp\left(-\frac{\varepsilon}{2j}\right).$$

Using this, we can now analyze the errors. When $|V_s(W,x,y)| = j$ for any such j,

$$\begin{aligned}
\Pr[\hat{\Delta}_s^-(W,x,y) + err_s(W,x,y) \le \Delta(W,x,y)] &= \Pr[\hat{\Delta}_s(W,x,y) - \Delta(W,x,y) \le -err_s(W,x,y)] \\
&= \Pr[j\cdot\hat{\Delta}_s(W,x,y) - j\cdot\Delta(W,x,y) \le -j\cdot err_s(W,x,y)] \\
&= \Pr[X_j - j\cdot\Delta(W,x,y) \le -j\cdot err_s(W,x,y)] \\
&= \Pr[Y_j \le -j\cdot err_s(W,x,y)] \\
&= \Pr\left[Y_j \le -j\cdot\sqrt{\frac{2\log\left(\frac{4L(k+1)m^{k+1}}{\delta}\right)}{j}}\right] \\
&= \Pr\left[Y_j \le -\sqrt{2j\log\left(\frac{4L(k+1)m^{k+1}}{\delta}\right)}\right] \\
&\le \exp\left(-\frac{\left(\sqrt{2j\log\left(\frac{4L(k+1)m^{k+1}}{\delta}\right)}\right)^2}{2j}\right) \\
&= \frac{\delta}{4L(k+1)m^{k+1}}.
\end{aligned}$$

Additionally, note that when $s = k$, this is in fact a martingale (no loose upper bounding is needed), so this inequality continues to hold in other direction for $\hat{\Delta}_k^-(W,x,y) - err_k(W,x,y) \ge \Delta(W,x,y)$. A symmetric argument shows

$$\Pr[\hat{\Delta}_s^-(W,x) - err_s(W,x) \ge \Delta(W,x)] \le \frac{\delta}{4L(k+1)m^{k+1}}$$

and

$$\Pr[\hat{\Delta}_k^-(W,x) + err_s(W,x) \le \Delta(W,x)] \le \frac{\delta}{4L(k+1)m^{k+1}}.$$

for all W, x, and s.

Notice that in the first L queries, the sizes of the V_s sets are trivially upper bounded by L. Hence, we can union bound over all at most L sizes, the two choices of either upper and lower bounds, two choices of either $\Delta(W,x,y)$ or $\Delta(W,x)$ the at most $k+1$ choices of s, and at most m^{k+1} choices of W, x, and y (we are choosing $m+1$ candidates with two being special, so clearly at most choosing a sequence of $k+1$ candidates with repeats). This leads to at most $4L(k+1)m^{k+1}$ possible bad events. Hence, with probability $1-\delta$, none of these bad events happen. Conditioned on this, we have that

$$\widetilde{\Delta}^+(W,x) = \min_{s\in[k]} \hat{\Delta}_s^+(W,x) + err_s(W,x) \ge \Delta(W,x)$$

and

$$\widetilde{\Delta}^-(W,x,y) = \max_{s\in[k]} \hat{\Delta}_s^-(W,x) - err_s(W,x) \ge \Delta(W,x).$$

In addition, using the bounds on $\hat{\Delta}_k$, we have

$$\widetilde{\Delta}^+(W,x) = \min_{s\in\{0\}\cup[k]} \hat{\Delta}_s^+(W,x) + err_s(W,x) \le \hat{\Delta}_k^+(W,x) + err_k(W,x) \le \Delta(W,x) + 2err_k(W,x)$$

and

$$\widetilde{\Delta}^-(W,x,y) = \max_{s\in\cup[k]} \hat{\Delta}_s^-(W,x,y) - err_s(W,x,y) \ge \hat{\Delta}_k^-(W,x,y) - err_k(W,x,y) \le \Delta(W,x,y) - 2err_k(W,x,y).$$

Hence, the desired bounds are satisfied. □

We are now ready to prove the theorem.

Proof of Theorem H.1. We condition on the event that the estimates after at most L voters are all accurate as in Lemma H.2. Let $\ell := 576\cdot\left(\frac{\alpha k^2}{(1-\alpha)k+1}\right)\log\left(\frac{4608k^8m^{k+2}}{\delta}\right)$ as defined in the algorithm. The technical portion of this proof is to show that for any committee W, after at most $\left\lceil\frac{m-k}{t-k}\right\rceil\cdot\ell$ queries, the algorithm either makes a swap or terminates. Notice that when a swap is

made, assuming the estimate is accurate, the PAV score increases by $\frac{(1-\alpha)k+1}{2\alpha k^2}$. Hence, just as in previous proofs, such a swap can only happen $2H_k \frac{\alpha k^2}{(1-\alpha)k}$ times. The choice of L implies termination will occur while estimates are still accurate. Hence, at the point that we terminate, $\Delta^*(W) < \frac{1}{\alpha k}$, so the desired properties are satisfied by Lemma 3.5.

What remains is to show that after $\left\lceil \frac{m-k}{t-k} \right\rceil \cdot \ell$ queries with a committee W, either a swap is made or we terminate. By our query selection strategy, after this many queries, $W \cup \{x\}$ will be contained in at least ℓ queries for all $x \notin W$. This implies that $|V_k(W,x)| \geq \ell$ and $|V_k(W,x,y)| \geq \ell$ for all such x and y. We will later show that when this happens, $err_k(W,x)$ and $err_k(W,x,y)$ are upper bounded by $\varepsilon := \frac{1}{12}\frac{(1-\alpha)k+1}{\alpha k^2}$ for all x and y. Once this upper bound of ε has been shown, the proof is very similar of a swap or termination is similar to Theorem 4.1. If $\widetilde{\Delta}^+(W,c') \geq \frac{1}{\alpha k}$, we will certainly terminate. Otherwise, if $\widetilde{\Delta}^+(W,c') < \frac{1}{\alpha k}$, this means $\Delta(W,c') < \frac{1}{\alpha k} - 2\varepsilon$. Hence, there is a candidate x such that

$$\Delta(W,c',x) \geq \frac{(1-\alpha)k+1}{\alpha k^2} - \frac{k+1}{k} \cdot 2\varepsilon \geq 12\varepsilon - 4\varepsilon = 8\varepsilon.$$

For such an x,

$$\widetilde{\Delta}^-(W,c',x) \geq \Delta(W,c',x) - 2\varepsilon \geq 6\varepsilon = \frac{1}{2}\frac{(1-\alpha)k+1}{\alpha k^2}.$$

Hence, the swap if condition must pass and a swap will be made.

Finally, let us show the necessary bound on err_k. More formally, we must show

$$\sqrt{\frac{2\log\left(\frac{4L(k+1)m^{k+1}}{\delta}\right)}{\ell}} \leq \varepsilon.$$

Observing that $\ell = \frac{2}{\varepsilon^2} \cdot 2\log\left(\frac{4608k^8m^{k+2}}{\delta}\right)$, it is sufficient to show that

$$\log\left(\frac{2L(k+1)m^{k+1}}{\delta}\right) \leq 2\log\left(\frac{4608k^8m^{k+2}}{\delta}\right)$$

To that end, we have

$$\begin{aligned}
\log\left(\frac{4L(k+1)m^{k+1}}{\delta}\right) &\leq \log\left(\frac{8Lkm^{k+1}}{\delta}\right) \\
&= \log\left(\frac{8\left(2H_k\left\lceil\frac{m-k}{t-k}\right\rceil\left(\frac{\alpha k^2}{(1-\alpha)k+1}\right)\cdot\ell\right)km^{k+1}}{\delta}\right) \\
&\leq \log\left(\frac{16\left(k\cdot m\cdot k^2\cdot\ell\right)km^{k+1}}{\delta}\right) \\
&= \log\left(\frac{16k^4m^{k+2}\cdot\ell}{\delta}\right) \\
&\leq \log\left(\frac{16k^4m^{k+2}\cdot\frac{2}{\varepsilon^2}\cdot\left(2\log\left(\frac{4608k^8m^{k+2}}{\delta}\right)\right)}{\delta}\right) && (\tfrac{1}{\varepsilon} \leq 12k^2) \\
&\leq \log\left(\frac{4608k^8m^{k+2}\cdot\left(2\log\left(\frac{4608k^8m^{k+2}}{\delta}\right)\right)}{\delta}\right) \\
&= \log\left(\frac{4608k^8m^{k+2}}{\delta}\right) + \log\left(2\log\left(\frac{4608k^8m^{k+2}}{\delta}\right)\right) \\
&\leq \log\left(\frac{4608k^8m^{k+2}}{\delta}\right) + \log\left(\frac{4608k^8m^{k+2}}{\delta}\right) && (\log(2a) \leq a \text{ for all } a \in \mathbb{R}) \\
&= 2\log\left(\frac{4608k^8m^{k+2}}{\delta}\right),
\end{aligned}$$

as needed. □

Table 3: Polis datasets statistics: Number of queried voters L, number of comments m, comments per query t, and fraction of comments t/m voted on by each voter.

L	m	t	t/m
162	31	20	0.65
1000	1719	20	0.01
87	39	20	0.51
353	231	20	0.09
340	209	20	0.10
94	40	20	0.50
1000	114	20	0.18
230	83	20	0.24
258	98	20	0.20
405	94	20	0.21
278	104	20	0.19
1000	586	20	0.03

Comparing Theorem H.1 with Theorem 4.1, we see that our upper bound on the query complexity Algorithm 4 is k times worse asymptotically. However, even in the worst case, it is unclear whether these bounds are tight; the difference may instead be due to slack in our analysis.

Beyond the worst case, there are problem instances where Algorithm 4 requires fewer queries than Algorithm 2. Consider a setting with k "good" candidates supported by all voters and $m' > k$ "bad" candidates that no one supports. Note that with $\alpha = 1$, to satisfy EJR, all good candidates must be selected. In this instance, Algorithm 2 will perform $\Theta(\frac{k^4 m}{t-k}\log(m))$ queries per swap. Further, since $m' > k$, with probability at least $\frac{1}{2}$, even a randomly-selected initial committee contains no more than $\frac{k}{2}$ good candidates, so $\Omega(k)$ swaps are required. Hence, Algorithm 2 requires $\Omega(\frac{k^5 m}{t-k}\log(m))$ queries.

In contrast, Algorithm 4 does not discard votes after swaps. In particular, consider the estimate $\hat{\Delta}_0^+(W, c)$ for a bad candidate c that uses all voters that voted on c regardless of if they voted on anyone in W. Note that it is always 0 as no voter ever approves of c. Hence, $\widetilde{\Delta}^+(W, c) \leq err_0(W, c)$. On the other hand, for all good candidates c, $\Delta(W, c) \geq 1/k$, so $\widetilde{\Delta}^+(W, c) \geq 1/k$ as well. Hence, once a bad candidate has been queried $\Omega(k^2 \log m)$ times, it will have a worse $\widetilde{\Delta}^+$ when compared to any good candidate. In addition, only $\Omega(k^2 \log m)$ queries are needed for a good candidate's error term to be small enough to ensure a swap (fewer for the earlier swaps). Hence, at most $O(mk^2 \log m)$ queries are needed for Algorithm 4 to terminate.

In summary, despite being slightly worse in terms of worst-case analysis, there is evidence that Algorithm 4 may work better in practice, an intuition confirmed through experimental comparison in Section 5.

I Details on Experiments

The data for each conversation consists of an $L \times m$ matrix, with component $v_{ij} \in \{\texttt{agree}, \texttt{disagree}, \texttt{missing}\}$ representing the vote of participant i on comment j.

Polis Datasets. See Table 3 for the sizes of the Polis dataset.

Reddit Dataset. We preprocessed this dataset in the same way as the Polis datasets (including matrix factorization to infer missing votes). Although the output on Reddit differs from Polis (rankings rather than a subset of the comments), the *input* is similar. We can interpret upvotes as approvals and downvotes as disapprovals, so the data fits well with our experiments. See Section 6 for additional discussion on how our approach applies to social media.

Algorithm Parameters As previously mentioned, in Algorithm 2, we treat ℓ, the number of times we ask voters about each candidate, as a parameter. Similarly, in Algorithm 4, we replace the numerator in the confidence intervals of $\Delta^-(W, c', c)$ with a parameter θ, and we set $\varepsilon = 0$. We assessed $\ell \in \{4, 6, 8, 12, 18, 30\}$ and $\theta \in \{0.03, 0.05, 0.1, 0.2, 0.5, 1.0\}$ on artificial data (e.g., approval profiles generated by sampling each vote independently). We observed that the algorithms are not sensitive to these parameters and picked $\ell = 6$ and $\theta = 0.05$ since they appeared to yield good results based on visual inspection.