

Finding Common Ground in a Sea of Alternatives

JAY CHOOI, Harvard University, USA

PAUL GÖLZ, Cornell University, USA

ARIEL D. PROCACCIA, Harvard University, USA

BEN SCHIFFER, Harvard University, USA

SHIRLEY ZHANG, Harvard University, USA

We study the problem of selecting a statement that finds common ground across diverse population preferences. Generative AI is uniquely suited for this task because it can access a practically infinite set of statements, but AI systems like the Habermas machine [Tessler et al., 2024] leave the choice of generated statement to a voting rule. What it means for this rule to find common ground, however, is not well-defined.

In this work, we propose a formal model for finding common ground in the infinite alternative setting based on the *proportional veto core* from social choice. To provide guarantees relative to these infinitely many alternatives and a large population, we wish to satisfy a notion of proportional veto core using only query access to the unknown distribution of alternatives and voters.

We design an efficient sampling-based algorithm that returns an alternative in the (approximate) proportional veto core with high probability and prove matching lower bounds, which show that no algorithm can do the same using fewer queries. On a synthetic dataset of preferences over text, we confirm the effectiveness of our sampling-based algorithm and compare other social choice methods as well as LLM-based methods in terms of how reliably they produce statements in the proportional veto core.

CONTENTS

Abstract	0
Contents	0
1 Introduction	1
2 Model	4
3 Basic Properties	5
4 Algorithmic Results	7
5 Lower Bounds	10
6 Interpretation of \mathcal{D}	12
7 Experimental Results	13
8 Discussion	17
References	19
A Reduction of ϵ -PVC to PVC	22
B Deferred Proofs	22
C Deferred Proofs for Computing ϵ -PVC and Critical Epsilons	26
D Additional Experiment Details	28

1 Introduction

In many countries, polarization has reached unhealthy levels [Silver, 2022], dividing politics into “us” versus “them”. This division exacts a heavy toll: it reduces trust between people, entails political gridlock and democratic erosion, and increases political violence [Lee, 2022, McCoy et al., 2018, Orhan, 2022, Piazza, 2023]. To stem the tide of polarization, recent initiatives encourage people to discuss politics across divides and find points of agreement. The nonprofit *Braver Angels*¹, for example, brings groups of Democrats and Republicans face to face, and *citizens’ assemblies* [Dryzek et al., 2019, OECD, 2020] convene a representative sample of constituents to design common policy proposals. Major obstacles, however, stand in the way of scaling these efforts to a society-wide level, including the cost of trained facilitators and cognitive limits that prevent high-quality deliberation in large groups [Fishkin, 2009]. The recent success of *large language models (LLMs)* in processing natural language text shows promise in overcoming these obstacles [Dembrane, 2023, Konya et al., 2023, Landemore, 2024, McKinney and Chwalisz, 2025, Small et al., 2023].

Perhaps the most prominent AI tool with this goal is the *Habermas machine*, presented by Tessler et al. [2024] in *Science*. To help a group deliberate on a topic, the Habermas machine transforms opinion statements from the group members into possible group statements, which are pieces of text representing the common ground of opinions. Tessler et al. [2024] test the Habermas machine in the context of a *citizens’ assembly* and find that groups interacting with the Habermas machine converge in their beliefs towards a common position, decreasing polarization.

Besides LLM components (for the generation of text and prediction of agreement), the Habermas machine also crucially relies on *voting* to select among generated proposals. Voting happens on two levels: first, to generate alternative statements, 16 statements are sampled from a fine-tuned LLM, ranked by a proxy model for each group member, and only the winner according to the Schulze voting rule [Schulze, 2011] is kept; second, four alternative statements generated in this fashion are ranked by the actual group members, and the winning statement is again selected by Schulze.

For finding common ground, Schulze may not be the right tool for selecting among statements because this voting rule is inherently majoritarian: it readily overrides the preferences of a minority to satisfy even a slight majority, as evidenced by axioms such as Condorcet consistency and the mutual majority criterion. While Tessler et al. [2024] find that minority opinions have an influence on the winning statement,² this is an empirical finding and only holds on average across their experiments. To robustly find common ground, we should aim for guarantees on the influence of minorities that hold for every possible group deliberation.

What, then, can social choice offer as an axiom for finding common ground and as a voting rule to be embedded inside the Habermas machine? We argue that the *proportional veto core (PVC)* of Moulin [1981, 1982] is the right notion since it gives explicit outcome guarantees to cohesive voter coalitions of all sizes. Consider a coalition of, say, 30% of the participants and some alternative statement c . If the coalition can find a set with $100\% - 30\% = 70\%$ of the alternatives that they all rank above c , the PVC guarantees that c will not be selected. In this way, minority groups are guaranteed an influence on the outcome, and they can avoid the statements they disagree with the most, which is important for a claim of common ground. To pursue this guarantee, the Habermas machine could use a voting rule whose output always lies in the PVC (like Vote by Veto [Moulin,

¹<https://braverangels.org/>

²Specifically, Tessler et al. [2024] embed the opinions by all group members using a text embedding model, and run a regression to predict the embedding of the winning statement as a mixture of these opinion statements. Group members are partitioned into a “majority” and “minority” group based on whether they answered “somewhat agree/agree/strongly agree” or “somewhat disagree/disagree/strongly disagree” in an initial question on the topic. Tessler et al. [2024] find that minority agents tend to have non-negligible weight in this regression.

[1981](#)]), one could constrain any voting rule to alternatives in the PVC, or one could evaluate voting rules (classical or LLM-based) in terms of how reliably their output lies in the PVC.

One more challenge of using social choice in settings like the Habermas machine is that the space of alternatives is tremendously large and often infinite. This leads to the natural question: Are the 16 statements sampled i.i.d. from the fine-tuned LLM enough to find common ground, and are the four statements resulting from independent runs of sampling and simulated voting enough? How can we capture what “enough” is? In this paper,

we define proportional veto core guarantees with respect to a very large set of statements and design practical algorithms that satisfy them.

Our pursuit of guarantees with respect to a large set of textual alternatives is closely aligned with the *generative social choice* framework by [Fish et al. \[2026\]](#). Like them, we can only access this space of alternatives indirectly through queries, which in our case means sampling from an unknown distribution over alternatives.

1.1 Our Contributions

We start by giving a formal mathematical model of what it means for a statement to find common ground when there are an infinite number of alternatives. We propose using the ϵ -proportional veto core (ϵ -PVC), which extends the standard PVC to the setting with a distribution over alternatives. We define the *critical ϵ* of an alternative as the smallest ϵ for which that alternative is in the ϵ -PVC. Therefore, a smaller critical ϵ indicates that a statement better represents common ground.

To model information elicitation when there are infinite alternatives, we study a query-based information model with two types of queries. First, we explore *generative* queries, which return alternatives sampled from an unknown distribution of interest over the alternative space, such as the output of an LLM with respect to a given prompt. Second, we explore *discriminative* queries, which return user preferences over alternatives. We focus on two types of discriminative queries: *min-queries* that return a voter’s least favorite alternative among a given set and *pairwise queries* that return a voter’s preference between two given alternatives.

Theoretical Results. Our theoretical work focuses on providing upper and lower bounds for how many queries are necessary to find an element in the ϵ -PVC. We begin by showing that the ϵ -PVC always has size at least ϵ . We then present a query-based algorithm that with high probability finds an element of the ϵ -PVC using $\tilde{O}(1/\epsilon^2)$ generative queries and min-queries. We prove that no algorithm can always find an element of the ϵ -PVC using fewer than $\Omega(1/\epsilon^2)$ generative queries and min-queries, which shows that our algorithm is worst-case optimal for both types of queries.

Pairwise queries are perhaps the easiest information elicitation method from the voters’ perspective. Therefore, we show that our algorithm can be adapted to use pairwise queries while giving the same theoretical guarantees. The key modification is showing that with $O(nm)$ pairwise queries, we can find an element of the PVC for finite sets of n voters and m alternatives. We provide an efficient method for doing this and further show that there is a matching lower bound for this step; i.e., no algorithm can find an element of the PVC with fewer than $\Omega(nm)$ pairwise queries for n voters and m alternatives.

Experimental Results. We conclude by empirically evaluating the critical ϵ achieved by different methods for selecting alternatives on simulated preference data. We obtain diverse preference data sets for a variety of topics by using ChatGPT to generate both alternatives and preference data for different personas. To model a query-based setting, the different voting rules are only given access to a small subset of voters and alternatives and must select an alternative from this subset.

Our first empirical result confirms that the sampling-based approach from the theory section finds alternatives in the ϵ -PVC for small ϵ . We then show that different social choice methods such as Schulze and plurality can select alternatives with large critical ϵ , indicating that these methods may not necessarily choose a good common-ground statement. Finally, we conclude our experiments by studying generative voting rules that use LLMs to generate common-ground statements. We show that these generative voting rules do generate statements with small critical ϵ , which indicates that the notion of PVC aligns well with how an LLM tries to find common ground. We also show that when the LLM is not properly tuned to the population, the generative voting rules give statements with significantly higher critical ϵ compared to our algorithm.

1.2 Related Work

Many recent works have studied how to use AI for finding consensus on a large scale [Bakker et al., 2022, Gudiño et al., 2024, Kahng et al., 2019, Konya et al., 2023, Landemore, 2022, Tessler et al., 2024]. The most similar of these methods is the Habermas machine [Tessler et al., 2024], which we already discussed at length. At the end of Section 2, we further discuss the clone-proofness axiom satisfied by the Schulze rule, and why our approach to common ground does not require it.

Finding common-ground statements is an important goal of online deliberation platforms like *Polis* [Small et al., 2021] and *Remesh*³, on which users submit statements and rate their agreement with the statements written by others. Both platforms [Computational Democracy Project, 2025, Konya et al., 2023] present user-submitted statements based on a “bridging-based ranking” [Ovadya and Thorburn, 2023], which measures if a statement has high approval rates in all clusters of users. Our approach is much more general because we give guarantees to all subsets of voters rather than only some prespecified groups.

The idea of finding common-ground and bridging has also been popular in the context of annotating social media posts [De et al., 2025, Wojcik et al., 2022]. For example, Wojcik et al. [2022] study a bridging-based algorithm for finding common-ground and selecting notes to append to tweets. Furthermore, Slaughter et al. [2025] show that this approach to finding common ground successfully reduces engagement with and proliferation of misinformation on social media platforms. While these methods for finding common ground have had practical success, they lack formal theoretical guarantees.

The PVC was first studied by Moulin [1981, 1982] in the traditional social choice setting of perfect information about a finite number of voters and alternatives. In the same setting, Ianovski and Kondratev [2023] give a polynomial time algorithm for computing the PVC and a neutral and anonymous algorithm for finding an element from the PVC. Ideas from these algorithms inspire some of our algorithmic procedures, as discussed in later sections. More recently, the PVC has been studied in a variety of contexts including approval ballots [Halpern et al., 2025], federated learning [Chaudhury et al., 2024], and policy aggregation [Alamdari et al., 2024]. The settings studied by Chaudhury et al. [2024] and [Alamdari et al., 2024] both can have an infinite number of alternatives, which lead these works to notions similar to the ϵ -PVC.

Like us, the work on *generative social choice* [Boehmer et al., 2025, Fish et al., 2026] uses generative models for social choice over an infinite space of textual alternatives. Similar to our setting, they study generative queries that generate alternatives and discriminative queries that access voter preferences. A major difference, however, is that their goal is to find a set of *several* representative statements, which enables them to give guarantees to minority groups without finding common ground within each statement. A second major difference is that their space of alternatives lacks our structure of a probability distribution. As a result, their generative queries (finding a statement

³<https://www.remesh.ai/>

maximizing a function of voter preferences) are much more complex and much harder to implement than our query of sampling from a fixed distribution.

Beyond the generative social choice model, there has also been significant work on applying social choice techniques to settings with many or infinite alternatives and limited information in the setting of AI alignment [Conitzer et al., 2024, Ge et al., 2024, Götz et al., 2025, Procaccia et al., 2025a] and AI evaluation [Procaccia et al., 2025b, Zhang and Hardt, 2024]. Due to the size of the alternative space in these settings, these works also focus on choosing subsets of alternatives and voters to give approximate or probabilistic generalization guarantees.

2 Model

Let \mathcal{M} be a (possibly infinite) set of alternatives and let \mathcal{N} be a finite (but potentially very large) set of voters. Each voter $i \in \mathcal{N}$ has a ranking over all alternatives in \mathcal{M} . For alternatives $a, b \in \mathcal{M}$ and voter $i \in \mathcal{N}$, we use $a \succ_i b$ to mean that voter i prefers alternative a to alternative b . Importantly, we do not assume perfect information about the preferences of the voters.

Instead, we assume that we can access information about the alternatives and voters using two types of queries: **generative queries** and **discriminative queries**. Generative queries allow us to access the set of alternatives \mathcal{M} . Specifically, let \mathcal{D} be a fixed distribution over \mathcal{M} . For a set of alternatives $M \subseteq \mathcal{M}$, define $\mu_{\mathcal{D}}(M)$ as the measure of the set M for distribution \mathcal{D} . A generative query returns a random sample from the distribution \mathcal{D} . Note that no information about the distribution \mathcal{D} or the rankings of the voters is known a priori.

Discriminative queries allow us to gather information about voter preferences. We study two types of discriminative queries. The first type of discriminative query we study is a *min-query*, which takes as input a voter $i \in \mathcal{N}$ and a set of alternatives $X \subseteq \mathcal{M}$ and returns voter i 's least favorite alternative in X . The second type of discriminative query we study is a *pairwise query*, which takes as input a voter $i \in \mathcal{N}$ and two alternatives $a, b \in \mathcal{M}$, and returns which alternative in $\{a, b\}$ is ranked higher by voter i . Our main goal is to use a combination of generative and discriminative queries to efficiently find an alternative in \mathcal{M} that finds common ground for the entire population \mathcal{N} .

We argue that the ϵ -Proportional Veto Core (ϵ -PVC) is a mathematically principled way of finding common ground. Informally, an alternative a is blocked by a coalition of voters if all voters in the coalition prefer a sufficiently large set of alternatives to a . The ϵ -PVC is simply the set of alternatives that is not blocked by any coalition of voters. By ruling out alternatives for which some sizable minority ranks a large block of options higher, the ϵ -PVC excludes polarizing alternatives.

Definition 2.1. An alternative $a \in \mathcal{M}$ is **ϵ -blocked** by a coalition $T \subseteq \mathcal{N}$ of voters if there exists a blocking set of alternatives $S \subseteq \mathcal{M}$ such that

- Each voter $i \in T$ prefers each alternative $b \in S$ to a (i.e. $b \succ_i a$), and
- $\mu_{\mathcal{D}}(S) > 1 - \frac{|T|}{n} + \epsilon$.

The **ϵ -Proportional Veto Core** (ϵ -PVC) is the set of all alternatives that are not ϵ -blocked by any coalition of voters.

When \mathcal{N} and \mathcal{M} are finite, the Proportional Veto Core (PVC) [Moulin, 1981] refers to the special case when $\epsilon = 0$ and \mathcal{D} is the uniform distribution over all alternatives in \mathcal{M} . See Appendix A for more details.

A higher $\epsilon > 0$ makes it harder for coalitions to block alternatives and therefore makes the ϵ -PVC a less demanding notion. Hence, the ϵ -PVC naturally leads to a metric for how well an alternative $a \in \mathcal{M}$ does at finding common ground for the voters \mathcal{N} . Intuitively, the critical ϵ for an alternative is the smallest fraction of voters that must be ignored in each coalition so that no coalition will veto

that alternative. Therefore, a critical ϵ closer to 0 implies that an alternative is better for finding common ground while a larger critical ϵ implies that an alternative is worse at it.

Definition 2.2. For any alternative $a \in \mathcal{M}$, the **critical** ϵ for a is the smallest $\epsilon \geq 0$ such that a is in the ϵ -PVC.

Role of the Statement Distribution. \mathcal{D} plays a crucial role in our notion of common ground since it determines how many alternatives a coalition T must agree on to block an alternative a . For example, let \mathcal{D} be defined by an LLM, which produces a neutral summary with 99% probability, but sometimes insults a minority instead. The ϵ -PVC allows the small minority to block the insulting statements, whose small measure in the distribution reflects its fringe nature. This marks a major difference in perspective to [Tessler et al. \[2024\]](#), who use the Schulze voting rule due to its clone-proofness. If the Habermas machine samples 99 neutral statements and one insulting statement from the LLM and voters see the neutral statements as interchangeable, clone-proofness actually ensures that a slight majority preference for the insulting statement makes it the winner. In addition, clone-proofness is a brittle axiom that only applies to exact clones, so similar but not identically ranked alternatives can still have a large effect on the outcome.⁴

As illustrated in the above example, having a distribution over alternatives enables us to guarantee rights to coalitions by forming a yardstick for how much influence each of them deserves. This does mean, of course, that the distribution must be carefully chosen to be broadly acceptable to the voters. In many cases, the statement distribution generated by an LLM might be a sufficiently neutral ground — say, from a base model, which seem to be more politically moderate than post-trained models [\[Rozado, 2024\]](#), or a model specifically fine-tuned for this purpose. Another option could be to define \mathcal{D} in terms of the distribution of voters as we do in our experiments: we draw characteristics of a random voter and ask the LLM to answer from their perspective. We continue our discussion on the choice of distribution \mathcal{D} in Section 6.

3 Basic Properties

To begin our investigation of the ϵ -PVC, we start by bounding its size and developing a polynomial-time algorithm that can compute the critical ϵ and the ϵ -PVC if \mathcal{D} is the uniform distribution over a given set of alternatives.

3.1 Size of ϵ -PVC

As a warm-up, we study the size of the ϵ -PVC, which will tell us how many generative queries are necessary to even see an element of the ϵ -PVC. We begin by showing that, for any $\epsilon > 0$, the ϵ -PVC is not only non-empty but always has a size (with respect to the measure $\mu_{\mathcal{D}}$) of at least ϵ . Previous papers that have looked at variants of the ϵ -PVC [\[Alamdari et al., 2024, Chaudhury et al., 2024\]](#) give similar proofs of existence, but do not explicitly study the size of the ϵ -PVC.

PROPOSITION 3.1. *For any instance of the problem and any ϵ , the size of the ϵ -PVC (with respect to the measure $\mu_{\mathcal{D}}$) is at least ϵ .*

PROOF SKETCH. The key idea of this proof relies on an algorithm we call *vote by γ -veto* (Algorithm 3 in Appendix B.1), which is guaranteed to return a set of alternatives C with mass at least ϵ such that all alternatives in C are also in the ϵ -PVC. In this algorithm, the voters are ordered randomly, and then in this order each voter vetoes their least favorite $\frac{1-\epsilon}{n}$ mass of alternatives that has not

⁴While the winner of Schulze on samples from \mathcal{D} is independent of the multiplicities of identical samples, we want to caution that the precondition of clone-proofness, requiring alternatives to be treated the same by *every* voter, makes it a brittle axiom. For example, [Pierczynski and Szufa \[2024\]](#) give an election where Schulze does not satisfy clone-proofness for alternatives that are almost clones. In this way, multiplicities can still have unpredictable effects under Schulze.

yet been vetoed. Clearly, once all n voters have vetoed $\frac{1-\epsilon}{n}$ mass, the remaining alternatives must have mass at least ϵ . The proof that the remaining alternatives are in the ϵ -PVC follows similarly to the proof for the Vote by Veto rule of [Moulin \[1982\]](#). Intuitively, any alternative a not in the ϵ -PVC must get removed before the algorithm ends, as the last voter to veto from the blocking set for this alternative a is guaranteed to veto a if a has not yet been vetoed. For the full proof, see [Appendix B.1](#). \square

A direct corollary of [Proposition 3.1](#) is that, among sufficiently many generative queries, there will with high probability be an alternative in the ϵ -PVC:

COROLLARY 3.2. *For any $\epsilon, \delta > 0$, a set of $m = \frac{\log(1/\delta)}{\epsilon}$ generative queries will with probability $1 - \delta$ contain an element of the ϵ -PVC.*

PROOF. Let M be the result of $m = \frac{\log(1/\delta)}{\epsilon}$ generative queries and let C_ϵ be the ϵ -PVC. Then

$$\Pr(M \cap C_\epsilon = \emptyset) = (1 - \mu_{\mathcal{D}}(C_\epsilon))^m \leq (1 - \epsilon)^m \leq e^{-\epsilon m} = \delta,$$

where the first inequality is by [Proposition 3.1](#). Therefore, the probability that at least one element of M is in C_ϵ is at least $1 - \delta$. \square

While $O(1/\epsilon)$ generative queries contain at least one element of the ϵ -PVC with high probability, we show in [Section 5](#) that a larger number, specifically $\Omega(1/\epsilon^2)$, of generative queries is needed to *identify* an element of the ϵ -PVC. Therefore, identifying an element of the ϵ -PVC with generative queries is statistically harder than simply generating an element of the ϵ -PVC.

In contrast to the traditional PVC, the 0-PVC can, in general, be empty:

PROPOSITION 3.3. *For an infinite number of alternatives, the 0-PVC can be empty.*

PROOF. Let $\mathcal{M} = \mathbb{N}_{\geq 1}$, and let each alternative $i \geq 1$ have mass $\frac{1}{2^i}$. Suppose that all voters prefer alternatives with larger numbers. We prove, by contradiction, that no alternative i is in the 0-PVC. Indeed, any such i is blocked by the grand coalition $T = \mathcal{N}$ and the alternative set $S = \{i+1\}$, since all voters prefer $i+1$ over i and since $\mu_{\mathcal{D}}(S) = \frac{1}{2^{i+1}} > 0 = 1 - \frac{|T|}{n}$. \square

3.2 Computing Critical ϵ

As mentioned in the previous section, the critical ϵ of an alternative measures how well a given alternative does at finding common ground. In this section, we give an algorithm for finding the critical ϵ of a given alternative in polynomial time when the number of alternatives is finite and \mathcal{D} is the uniform distribution over \mathcal{M} . Our algorithm generalizes the algorithm of [Ianovski and Kondratev \[2023\]](#) for deciding if an alternative is in the traditional PVC. By running this algorithm for all alternatives, we can construct the entire ϵ -PVC in polynomial time in the same setting.

ALGORITHM 1: Computing the critical ϵ

Input: Alternative a , $|\mathcal{N}| = n$, $|\mathcal{M}| = m$

- 1 Construct a flow graph G as follows:
 - 2 Connect a source node to n voter nodes, each edge with weight m
 - 3 Connect $m - 1$ nodes (for all alternatives except a) to a sink node, each with edge weight n
 - 4 Connect voter node i to alternative node j with infinite weight if voter i prefers a to alternative j
 - 5 Let K be the size of the min-cut for graph G
 - 6 **return** $\frac{2nm - n - K}{nm} - 1$

THEOREM 3.4. *When \mathcal{M} is finite, Algorithm 1 computes the critical ϵ of a given alternative in polynomial time.*

PROOF SKETCH. Our proof generalizes the ideas of [Iakovski and Kondratev \[2023\]](#) for computing the PVC. Their algorithm checks whether an alternative a is in the PVC by constructing a graph G_a such that there exists a complete biclique in G_a if and only if a is not in the PVC. We show that for the same graph G_a , the size of the maximum biclique is larger than $(1 + \epsilon)nm$ if and only if that alternative is not in the ϵ -PVC. In order to efficiently compute the size of the maximum biclique, we follow the same method as [Iakovski and Kondratev \[2023\]](#) of constructing the flow graph in Algorithm 1 such that the weight of the max flow K in this graph plus the size of the maximum biclique in G_a equals $2nm - n$. In the last line of Algorithm 1, we use this fact to find the smallest ϵ for which the maximum biclique of G_a is larger than $(1 + \epsilon)nm$. The runtime of the algorithm is polynomial because the min-cut can be found in polynomial time (Appendix C.2.2). For the detailed proof, see Appendix C. \square

4 Algorithmic Results

In this section, we present our main algorithmic results for efficiently finding an element of the ϵ -PVC. We give a simple algorithm using generative queries and min-queries that finds an alternative in the ϵ -PVC, with high probability, using $\tilde{O}(1/\epsilon^2)$ queries of each type. A key takeaway from this result is that the number of generative and discriminative queries does *not* depend on the size of \mathcal{N} (the number of voters) or the distribution over alternatives. This is important because the population may be large and the generative distribution altogether unknown to the algorithm. Then, we discuss how the same algorithm can be modified to use pairwise discriminative queries instead of min-queries.

4.1 Min-Queries

We now present and analyze Algorithm 2, which finds an element of the ϵ -PVC using generative queries and min-queries.

ALGORITHM 2: Finding an element of the ϵ -PVC

```

1 Input:  $\mathcal{N}, \mathcal{M}, \delta, \epsilon$ 
2 Output: An alternative that is with probability  $1 - \delta$  an element of the  $\epsilon$ -PVC
3 Set  $\tau = \frac{8}{\epsilon^2} \log \left( \frac{32}{\epsilon^2 \delta} \log \left( \frac{32}{\epsilon^2 \delta} \right) \right)$ .
4 Let  $M$  be a set of alternatives that is the result of  $\tau$  generative queries.
5 Initialize  $X = M$ .
6 while  $|X| > 1$  do
7   Select a random voter  $i \in \mathcal{N}$ .
8   Do one min-query to find voter  $i$ 's least favorite alternative among  $X$ .
9   Remove that alternative from  $X$ .
10 end
11 return the remaining element in  $X$ 

```

THEOREM 4.1. *Using $O(\epsilon^{-2} \log(\frac{1}{\epsilon \delta}))$ generative and min-queries, Algorithm 2 returns an element of the ϵ -PVC with probability at least $1 - \delta$.*

PROOF. By construction, Algorithm 2 uses τ generative queries and $\tau - 1$ min-queries, which gives the desired bound on the number of queries, observing that, for small enough ϵ, δ ,

$$\tau = \frac{8}{\epsilon^2} \log \left(\frac{32}{\epsilon^2 \delta} \log \left(\frac{32}{\epsilon^2 \delta} \right) \right) \leq \frac{8}{\epsilon^2} \log \left(\frac{32^2}{\epsilon^4 \delta^2} \right) \leq \frac{8}{\epsilon^2} \log \left(\frac{32^4}{\epsilon^4 \delta^4} \right) = \frac{32}{\epsilon^2} \log \frac{32}{\epsilon \delta} = O \left(\epsilon^{-2} \log \left(\frac{1}{\epsilon \delta} \right) \right).$$

Let N be the set of voters that are selected in Lines 6–9 of Algorithm 2. Note that a voter $i \in N$ could be chosen in two different iterations of the while loop, in which case we view these as distinct copies of the voter in the set N . This means we must have that $|N| = \tau - 1$.

We first show that any alternative in the PVC for voters N and alternatives M will, with probability $1 - \delta$, be in the ϵ -PVC for N, \mathcal{D} . We defer the proof of this result to Lemma 4.2 below. With this result, all we need to show is that the alternative we return is in the PVC for N and M .

In Lines 6–9, we are functionally applying the Vote by Veto rule from [Moulin, 1982] to find an element of the PVC for N and M using one min-query for each voter in N . Let a be the alternative returned by Algorithm 2, and assume for contradiction that a is not in the PVC for N and M , i.e., not in the 0-PVC for the uniform distribution over M . Hence, there exists a coalition of voters T and a set of alternatives S such that $b \succ_i a$ for each $i \in T$ and $b \in S$ and such that

$$\frac{|S|}{\tau} = \frac{|S|}{|M|} > 1 - \frac{|T|}{|N|} \quad (1)$$

Each voter $i \in N$ removed one alternative $b_i \in M$ from the set X . Since a was available in X at the time and the voter chose to remove b_i , it must hold that $a \succ_i b_i$. This means that the set $W := \{b_i \mid i \in T\}$ of alternatives removed by the voters in T is disjoint from S . Moreover, a is clearly not part of either set (because it was not removed and cannot be strictly preferred over itself). This means that $S \subseteq M \setminus W \setminus \{a\}$ and hence $|S| \leq \tau - |T| - 1$. It follows that

$$\frac{|S|}{\tau} \leq \frac{\tau - |T| - 1}{\tau} = 1 - \frac{|T| + 1}{|N| + 1} \leq 1 - \frac{|T|}{|N|},$$

where the last inequality follows since $|T| \leq |N|$. This contradicts Eq. (1), showing that a is indeed in the PVC for N and M . Together with the lemma, we conclude that the returned statement is in the ϵ -PVC with probability at least $1 - \delta$. \square

The key to concluding the proof of the theorem is Lemma 4.2, a quite technical result that connects the ϵ -PVC of the full instance with the PVC for the sampled voters and alternatives. At a high level, the proof must argue that the randomly selected set of voters and alternatives are sufficiently representative of the population and alternative distribution. But it is not immediately obvious how to do so, given the exponential number of coalitions, and infinite collection of alternative sets that could show that an alternative is ϵ -blocked.

Instead, we apply concentration inequalities in a more subtle manner so that the number of generative queries needed depends only on ϵ and is independent of $|N|$ (see Line 3 of Algorithm 2). Specifically, we show that we do not need enough samples to sufficiently approximate every subset of voters and alternatives, but rather only need enough samples to sufficiently approximate the voters and alternatives for the blocking coalitions of the generated alternatives.

LEMMA 4.2. *Let N be the set of (not necessarily distinct voters) selected in Lines 6–9 of Algorithm 2. For any $\delta, \epsilon > 0$, every element of the PVC for N and M from Algorithm 2 will be in the ϵ -PVC for N and \mathcal{D} .*

PROOF. The key observation is that every alternative $x \in M$ not in the ϵ -PVC for M, N, \mathcal{D} must have some blocking coalition of voters $T_x \subseteq N$ and alternatives $S_x \subseteq M$ satisfying Definition 2.1. Denote the i th alternative sampled on Line 4 as x_i . If x_i is not in the ϵ -PVC for M, N, \mathcal{D} , then we will show that with high probability, the corresponding blocking sets T_{x_i} and S_{x_i} are sufficiently

accurately proportionally represented in the sampled sets M and N up to a factor of ϵ . If this is the case, then Definition 2.1 implies that x_i will also not be in the PVC for M and N . For each x_i , define the event E_i as the event that the above holds if x_i is not in the ϵ -PVC. If we show that E_i holds with high probability for every alternative $x_i \in M$, then we can conclude that every alternative in $a \in M$ falls in one of two cases: Either a is in the ϵ -PVC for $\mathcal{M}, \mathcal{N}, \mathcal{D}$ or a is not in the PVC for N and M . This implies the desired result that any element of the PVC for N and M is in the ϵ -PVC for \mathcal{N}, \mathcal{D} .

We now proceed with the formal proof:

Let X be a set of alternatives sampled as generative queries from \mathcal{D} with size $|X| = \frac{8 \log\left(\frac{32}{\epsilon^2 \delta} \log\left(\frac{32}{\epsilon^2 \delta}\right)\right)}{\epsilon^2}$

and let $Y \subseteq \mathcal{N}$ be a random subset of \mathcal{N} with size $|Y| = \frac{8 \log\left(\frac{32}{\epsilon^2 \delta} \log\left(\frac{32}{\epsilon^2 \delta}\right)\right)}{\epsilon^2}$. We will show that with probability $1 - \delta$, any alternative that is not in the ϵ -PVC for the full set of alternatives will not be in the PVC for alternatives X and voters Y . This implies that any element of the PVC for alternatives X and voters Y will with probability $1 - \delta$ be in the ϵ -PVC for \mathcal{N}, \mathcal{D} . Define $PVC(X, Y)$ as the PVC for alternatives X and voters Y . Define $PVC_\epsilon(\mathcal{N}, \mathcal{D})$ as the ϵ -PVC for \mathcal{N}, \mathcal{D} . For any element $x \in \mathcal{M}$ such that $x \notin PVC_\epsilon(\mathcal{N}, \mathcal{D})$, there exists some $T_x \subseteq \mathcal{N}$ and $S_x \subseteq \mathcal{M}$ such that for all $i \in T_x$ and $a \in S_x$, $a \succ_i x$ and $\mu_{\mathcal{D}}(S_x) + \frac{|T_x|}{|\mathcal{N}|} > 1 + \epsilon$.

Let $X = \{x_1, \dots, x_{|X|}\}$ be the ordered set of generative queries. Now we define the event $E_i := \left\{x_i \in PVC_\epsilon(\mathcal{N}, \mathcal{D})\right\} \cup \left\{x_i \notin PVC_\epsilon(\mathcal{N}, \mathcal{D}) \text{ and } \frac{|X \cap S_{x_i}|}{|X|} \geq \mu_{\mathcal{D}}(S_{x_i}) - \epsilon/4 \text{ and } \frac{|Y \cap T_{x_i}|}{|Y|} \geq \frac{|T_{x_i}|}{|\mathcal{N}|} - \epsilon/4\right\}$. First, note that if $\bigcap_{i=1}^{|X|} E_i$ holds, then every element of X is either in $PVC_\epsilon(\mathcal{N}, \mathcal{D})$ or is not in $PVC(X, Y)$. This implies that every element not in $PVC_\epsilon(\mathcal{N}, \mathcal{D})$ is not in $PVC(X, Y)$.

All that remains is to show that $\Pr\left(\bigcap_{i=1}^{|X|} E_i\right) \geq 1 - \delta$. First, we will start with lower bounding $\Pr(E_i)$ (which by symmetry is the same for all i). We have:

$$\Pr(E_i) = \Pr(x_i \in PVC_\epsilon(\mathcal{N}, \mathcal{D})) + \Pr\left(\frac{|X \cap S_{x_i}|}{|X|} \geq \mu_{\mathcal{D}}(S_{x_i}) - \frac{\epsilon}{4} \text{ and } \frac{|Y \cap T_{x_i}|}{|Y|} \geq \frac{|T_{x_i}|}{|\mathcal{N}|} - \frac{\epsilon}{4} \middle| x_i \notin PVC_\epsilon(\mathcal{N}, \mathcal{D})\right) \Pr(x_i \notin PVC_\epsilon(\mathcal{N}, \mathcal{D})).$$

Focusing on the middle probability and using Hoeffding's inequality, we have that

$$\begin{aligned} & \Pr\left(\frac{|X \cap S_{x_i}|}{|X|} \geq \mu_{\mathcal{D}}(S_{x_i}) - \frac{\epsilon}{4} \text{ and } \frac{|Y \cap T_{x_i}|}{|Y|} \geq \frac{|T_{x_i}|}{|\mathcal{N}|} - \frac{\epsilon}{4} \middle| x_i \notin PVC_\epsilon(\mathcal{N}, \mathcal{D})\right) \\ & \geq 1 - \Pr\left(\frac{|X \cap S_{x_i}|}{|X|} < \mu_{\mathcal{D}}(S_{x_i}) - \frac{\epsilon}{4} \middle| x_i \notin PVC_\epsilon(\mathcal{N}, \mathcal{D})\right) - \Pr\left(\frac{|Y \cap T_{x_i}|}{|Y|} < \frac{|T_{x_i}|}{|\mathcal{N}|} - \frac{\epsilon}{4} \middle| x_i \notin PVC_\epsilon(\mathcal{N}, \mathcal{D})\right) \\ & \geq 1 - e^{-2|X|\epsilon^2/16} - e^{-2|Y|\epsilon^2/16} & \text{[Hoeffding's Ineq.]} \\ & = 1 - 2e^{-2|X|\epsilon^2/16} & [|X| = |Y|] \\ & = 1 - 2e^{-2 \frac{8 \log\left(\frac{32}{\epsilon^2 \delta} \log\left(\frac{32}{\epsilon^2 \delta}\right)\right)}{\epsilon^2} \epsilon^2/16} \\ & = 1 - \frac{\delta}{\frac{16 \log\left(\frac{32}{\epsilon^2 \delta}\right)}{\epsilon^2}} \\ & \geq 1 - \frac{\delta}{\frac{8 \log\left(\frac{32}{\epsilon^2 \delta} \log\left(\frac{32}{\epsilon^2 \delta}\right)\right)}{\epsilon^2}} \\ & = 1 - \frac{\delta}{|X|}. \end{aligned}$$

Returning to bounding $\Pr(E_i)$, we have that

$$\Pr(E_i) \geq \Pr(x_i \in PVC_\epsilon(\mathcal{N}, \mathcal{D})) + \left(1 - \frac{\delta}{|X|}\right) \Pr(x_i \notin PVC_\epsilon(\mathcal{N}, \mathcal{D})) \geq 1 - \frac{\delta}{|X|}.$$

Finally, we can conclude that $\Pr\left(\bigcap_{i=1}^{|X|} E_i\right) \geq 1 - \sum_{i=1}^{|X|} \Pr(\neg E_i) \geq 1 - \delta$ as desired. \square

4.2 Pairwise Discriminative Queries

While Algorithm 2 uses min-queries, it would be much more practically useful to have an algorithm based on pairwise discriminative queries, which are easier for voters to respond to. Fortunately, we can achieve this with one simple change to Algorithm 2.

Specifically, we can simulate each min-query in Line 8 with a simple iterative algorithm to find the least-preferred alternative for a given voter in a linear number of pairwise comparisons: First, choose an arbitrary ordering of the candidates in X and do a single pairwise query asking voter i to compare the first two alternatives in the ordering. Then, ask voter i to compare the least-preferred alternative from that first comparison to the next alternative in the ordering. Repeat this process, always keeping the least-preferred alternative from each comparison. After iterating through all alternatives in X , this is guaranteed to only use $|X| - 1$ pairwise queries to find voter i 's least favorite alternative among X .

5 Lower Bounds

In this section, we show that Algorithm 2 is tight in terms of both generative and discriminative queries. To prove this for each of generative and min-queries, we give a specific hard instance of the problem that requires $\tilde{\Omega}(1/\epsilon^2)$ of that type of query in order to find an element of the ϵ -PVC. We then show that the number of pairwise queries in the modification of Section 4.2 is also tight.

5.1 Generative and Min-Queries

In Theorem 5.1, we give a lower bound on the number of generative queries needed in order to identify an element of the ϵ -PVC. Importantly, this result combined with Corollary 3.2 implies a fundamental gap between the number of queries needed to generate versus identify an element of the ϵ -PVC. Specifically, Corollary 3.2 says that $\tilde{O}(1/\epsilon)$ generative queries suffice to *generate* an element of the ϵ -PVC with high probability, while Lemma 5.1 says that we need at least $\tilde{\Omega}(1/\epsilon^2)$ generative queries to *identify* an element of ϵ -PVC. This highlights a surprising subtlety about generative queries, which is that it can be easier to stumble across a statement in the ϵ -PVC than to certify which statement is in the ϵ -PVC.

THEOREM 5.1. *For any $\delta, \epsilon > 0$, no algorithm can use fewer than $\Omega(\log(1/\delta)/\epsilon^2)$ generative queries and for every distribution \mathcal{D} identify an element of the ϵ -PVC with probability greater than $1 - \delta$.*

PROOF. Consider the following two discrete distributions with support $\{a_1, a_2\}$. Distribution \mathcal{D}_1 puts $0.5 + \epsilon$ probability mass on a_1 and $0.5 - \epsilon$ probability mass on a_2 . Distribution \mathcal{D}_2 puts $0.5 - \epsilon$ probability mass on a_1 and $0.5 + \epsilon$ probability mass on a_2 . Suppose that half of the voters in \mathcal{N} have preferences $a_1 \succ a_2$ and the other half of the voters have preference $a_2 \succ a_1$. With this voter population, the ϵ -PVC under distribution \mathcal{D}_1 is $\{a_1\}$ while the ϵ -PVC under distribution \mathcal{D}_2 is $\{a_2\}$. Therefore, in order to find the ϵ -PVC, we must have sufficient information to distinguish between distributions \mathcal{D}_1 and \mathcal{D}_2 . Next, note that the KL distance between these distributions satisfies $KL(\mathcal{D}_1 || \mathcal{D}_2) \leq O(\epsilon^2)$. Therefore, a standard information theory result (stated and proven in Lemma 5.3 for completeness) gives that with fewer than $\Omega(\log(1/\delta)/\epsilon^2)$ generative queries, it is impossible to distinguish between these two distributions with probability greater than $1 - \delta$, which implies the desired result. \square

A similar construction shows that in the worst case, we also need at least $\tilde{O}(1/\epsilon^2)$ min-queries to distinguish between two populations that are both close to ambivalent between two alternatives. The main difference between this construction and the one in Theorem 5.1 is that it is difficult to distinguish between two similar populations rather than two similar distributions.

THEOREM 5.2. *For any $\delta, \epsilon > 0$, no algorithm can use fewer than $\Omega(\log(1/\delta)/\epsilon^2)$ discriminative queries and for every distribution \mathcal{D} identify an element of the ϵ -PVC with probability greater than $1 - \delta$.*

PROOF. Consider the uniform distribution \mathcal{D} with support $\{a_1, a_2\}$. Consider the following two sets of voters. In \mathcal{N}_1 , a $(1/2 + \epsilon)$ fraction of voters have preference $a_1 \succ a_2$ and a $(1/2 - \epsilon)$ fraction of voters have preference $a_2 \succ a_1$. In \mathcal{N}_2 , a $(1/2 + \epsilon)$ fraction of voters have preference $a_2 \succ a_1$ and a $(1/2 - \epsilon)$ fraction of voters have preference $a_1 \succ a_2$. The ϵ -PVC for voters \mathcal{N}_1 is just $\{a_1\}$ while the ϵ -PVC for the voters \mathcal{N}_2 is $\{a_2\}$. Distinguishing between these two voter populations with discriminative queries is equivalent to distinguishing between a $Bernoulli(1/2 + \epsilon)$ distribution and a $Bernoulli(1/2 - \epsilon)$ distribution. We can again appeal to Lemma 5.3 to get the desired result that no algorithm can use fewer than $\Omega(\log(1/\delta)/\epsilon^2)$ queries to identify an element of the ϵ -PVC with probability higher than $1 - \delta$ for both of these settings. \square

The proofs of Theorem 5.1 and Theorem 5.2 both rely on the following standard result from information theory, which we state here formally and reprove for completeness.

LEMMA 5.3. *If two distributions \mathcal{D}_1 and \mathcal{D}_2 have Kullback-Leibler distance satisfying $KL(\mathcal{D}_1 \parallel \mathcal{D}_2) \leq c\epsilon^2$ for some constant c , then no algorithm can achieve minimax misclassification error of less than δ for these two distributions using fewer than $O(\log(1/\delta)/\epsilon^2)$ samples.*

PROOF. Let \mathcal{D}_1^n and \mathcal{D}_2^n be the product distributions for n samples from \mathcal{D}_1 and \mathcal{D}_2 respectively. Then by definition, $KL(\mathcal{D}_1^n, \mathcal{D}_2^n) = n\epsilon^2$. By the Bretagnolle-Huber Inequality, this implies that the $TV(\mathcal{D}_1^n, \mathcal{D}_2^n) \leq 1 - \frac{1}{2}e^{-n\epsilon^2}$. By Le Cam's method, the minimax misclassification error for any classifier distinguishing between $\mathcal{D}_1^n, \mathcal{D}_2^n$ must be lower bounded by $\frac{1}{2}(1 - TV(\mathcal{D}_1^n, \mathcal{D}_2^n)) \geq \frac{1}{4}e^{-n\epsilon^2}$. Plugging in $n = \log(1/(4\delta))/(c\epsilon^2)$ gives a misclassification error of at least δ , which is the desired result. \square

5.2 Pairwise Discriminative Queries

Theorems 5.1 and 5.2 together imply that Algorithm 2 is tight in terms of the number of generative queries and min-queries needed. As discussed in Section 4, Algorithm 2 works by finding an element of the PVC for a set of voters N and alternatives M . We also showed that this algorithm can be modified to use $O(nm)$ pairwise queries, where $n = |N|$ and $m = |M|$. In Theorem 5.4, we show that finding the PVC from a set of alternatives N and alternatives M (with sizes n and m respectively) cannot be done in fewer than $\Omega(nm)$ pairwise discriminative queries, which implies that there is no better subroutine than this modification in the worst case. Theorem 5.4 and its proof may be of independent interest to those studying the elicitation of voter preferences using pairwise comparisons. The proof below establishes a lower bound of $nm/32$; we present a longer proof of the same result that achieves a tighter lower bound of $nm/2$ in Appendix B.2.

THEOREM 5.4. *When the sets of alternatives M and voters N are finite, no algorithm can always find an element in the PVC using fewer than $\Omega(mn)$ pairwise discriminative queries.*

PROOF. Let $m = kn + 1$ for any $k \geq 1$. By definition, any alternative a in the PVC must not be ranked in the last k alternatives by any voter. We will show that any algorithm ALG needs at least

$nm/32$ queries to find any alternative not ranked in the last k by any voter, which implies we need at least $nm/32$ queries to find any alternative in the PVC.

Consider the following adversary. The adversary answers pairwise queries by *ALG* arbitrarily (and consistently with previous queries), and for every alternative a keeps track of a counter c_a which counts the number of pairwise queries involving alternative a up to this point. The adversary also keeps track of a set X of “active” voters which is initialized to be \mathcal{N} . As soon as the counter c_a for an alternative a hits $n/4$, the adversary arbitrarily chooses one of the remaining “active” voters i for which a can be in the bottom k , and assigns a to be one of voter i ’s k worst ranked alternatives. The adversary then truthfully tells *ALG* that a is ranked in one of the last k spots for voter i . Similarly, if the queries answered so far force an alternative a to be ranked in the last k of a voter i , then a is also assigned to voter i . If voter i has k distinct alternatives assigned to voter i ’s last k spots, then voter i is no longer active and is removed from the set of active voters.

Now suppose that *ALG* has asked fewer than $nm/32$ discriminative queries. Partition the full set of alternatives M into M_1 and M_2 , such that M_1 contains all alternatives that have been involved in at least $n/4$ discriminative queries up until this point, and M_2 contains all alternatives that have been involved in strictly less than $n/4$ discriminative queries. Note that for any alternative in M_1 , the adversary has already placed that alternative in the last k spots of at least one voter, and therefore no alternative in M_1 is in the PVC.

To conclude the proof, we must show that *ALG* cannot determine whether or not any alternative in M_2 is in the PVC. Because *ALG* has asked fewer than $nm/32$ queries so far and each query involves exactly two alternatives, the number of alternatives involved in at least $n/4$ discriminative queries is at most $m/4$. This implies that $|M_1| \leq m/4$. An alternative a is only assigned to a voter if either it is in M_1 or if the queries for a specific voter i force a to be in i ’s bottom k spots. For the latter case, there must have been at least $\frac{m-1}{k} = n$ queries per such assigned a . Therefore, there are at least $nk - m/4 - \frac{nm/32}{n} \geq \frac{23kn-9}{32}$ remaining spots left in the bottom k positions of the active voters, which implies there are at least $23n/32 - 1$ active voters remaining. For any alternative $a \in M_2$, we know that strictly less than $n/4$ voters have been queried about a . Therefore, since there are at least $23n/32 - 1$ active voters remaining, there must be at least $23n/32 - 1 - n/4 > 0$ active voters (for sufficiently large n) who have not yet been queried about alternative a . Importantly, this means that a may be in one of the last k spots of these voters’ rankings, which further implies that *ALG* is unable to determine whether a is in the PVC. This is true for every alternative $a \in M_2$, so we can conclude that *ALG* cannot find any element of the PVC with fewer than $nm/32$ queries. \square

6 Interpretation of \mathcal{D}

In this section, we further discuss the significance of the distribution \mathcal{D} . One way to think about \mathcal{D} is to view it as a weighting scheme that puts more weight on more “important” alternatives. Traditional voting theory does not explicitly assume such a weighting scheme, but does commonly assume neutrality. A voting rule satisfies neutrality if it is indifferent to the identity of the alternatives, in other words the outcome does not change if the alternatives are permuted. Therefore, any voting rule that satisfies neutrality must implicitly weight the alternatives equally, i.e. use a uniform weighting. Similarly, when there is a compact space of infinite alternatives, any voting rule satisfying neutrality must implicitly use a distribution \mathcal{D} that is the uniform distribution over all alternatives.

All of our theoretical results (both positive and negative) apply when \mathcal{D} is the uniform distribution. However, when there are infinite alternatives, the uniform distribution may not always be the correct choice. As an example, consider the alternative space $\mathcal{M} = [0, 1]$ with 100 voters. 99 of these voters only like alternative 0 and one voter only likes alternative 1. In this example, despite infinite alternatives existing in the interval $[0, 1]$, the only two alternatives that should be relevant are $\{0, 1\}$,

which have measure 0 under the uniform distribution. Unfortunately, the natural generalization of many voting rules (e.g., Borda count) ignore sets of size 0, and therefore may even select an alternative not from $\{0, 1\}$. Similarly, when computing the ϵ -PVC, both of the alternatives in $\{0, 1\}$ with measure 0 could be vetoed depending on tie-breaking. This is clearly an undesirable outcome, and can be easily avoided by choosing a more appropriate distribution \mathcal{D} .

Motivated by the example above, one way to choose the distribution \mathcal{D} is to put more weight on alternatives that are more “important”. This means that the distribution \mathcal{D} is implicitly capturing the cardinal preferences of the voters by putting higher weight on alternatives for which voters have higher utilities. A common choice for such a distribution is the Bradley-Terry-Luce (BTL) model, where the probability of selecting an alternative is proportional to a score for that alternative. Suppose each player i has a utility function $u_i : \mathcal{M} \rightarrow [0, 1]$ such that $\int_{\mathcal{M}} u_i(m) dm = 1$. Then we could let \mathcal{D} be the mixture of BTL models over all voters, i.e. $\mu_{\mathcal{D}}(m) = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} u_i(m)$. This is equivalent to a BTL model with scores equal to the sum of voter utilities.

Happily, choosing \mathcal{D} to be this BTL model also gives us an intuitive interpretation for the ϵ -PVC. If \mathcal{D} is this BTL model, then the size of a coalition necessary to veto an alternative is proportional to the utilitarian social welfare (total population utility) for that alternative. This means that a larger coalition of voters is needed to veto an alternative that has higher utility among the voters. We formalize this idea below in the extreme case .

THEOREM 6.1. *If \mathcal{D} satisfies $\mu_{\mathcal{D}}(m) = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} u_i(m)$ then the following holds for any subset $S \in \mathcal{M}$. If at least an $x + \epsilon$ fraction of voters rank every alternative in S above every alternative not in S and the total utility of S is greater than $1 - x$ fraction of the total utility of all alternatives, then the ϵ -PVC must be a subset of S .*

PROOF. Let T be the coalition of voters that prefers S to every alternative not in S and satisfies $\frac{|T|}{n} \geq x + \epsilon$. We will show that the coalition T with alternatives S can veto every alternative $a \in \mathcal{M} \setminus S$. This follows directly from the fact that $\mu_{\mathcal{D}}(S) = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} u_i(m) > 1 - x \geq 1 - (x + \epsilon) + \epsilon \geq 1 - \frac{|T|}{n} + \epsilon$ which satisfies Definition 2.1. \square

In the discrete setting, we have the following corollary:

COROLLARY 6.2. *If \mathcal{D} satisfies $\mu_{\mathcal{D}}(m) = \frac{1}{|\mathcal{N}|} \sum_{i \in \mathcal{N}} u_i(m)$ then the following holds. If at least an $x + \epsilon$ fraction of the population ranks alternative $a \in \mathcal{M}$ first and the utilitarian social welfare of a is greater than x fraction of the total utility of all alternatives, then a is the only alternative in the ϵ -PVC.*

For intuition, suppose $x = 1/2$ and \mathcal{D} is the BTL distribution (as above). In this case, Corollary 6.2 says that if there is any alternative a such that a majority of voters rank a first and a has more than half of the total population utility, then a is the only alternative in the ϵ -PVC. Similarly, if at least a $1/2 + \epsilon$ fraction of voters have utility 1 for a and utility 0 for every other alternative, then a is the only alternative in the ϵ -PVC. Examples such as these demonstrate how the ϵ -PVC combines both ordinal and cardinal preferences when \mathcal{D} is the utility-based BTL model.

7 Experimental Results

We now analyze the ϵ -PVC of synthetically generated preference data, modeling the preferences of individuals over opinion statements on various policy issues. Our first experimental goal is to verify that our algorithm performs well in practical settings. This complements our asymptotic analysis in Section 4, demonstrating that our algorithm also performs well at small sample sizes. Our second goal is to measure how often popular voting rules choose alternatives that are in the ϵ -PVC. Our third goal is to understand how well LLMs can generate statements in the ϵ -PVC given different generative capabilities and types of information.

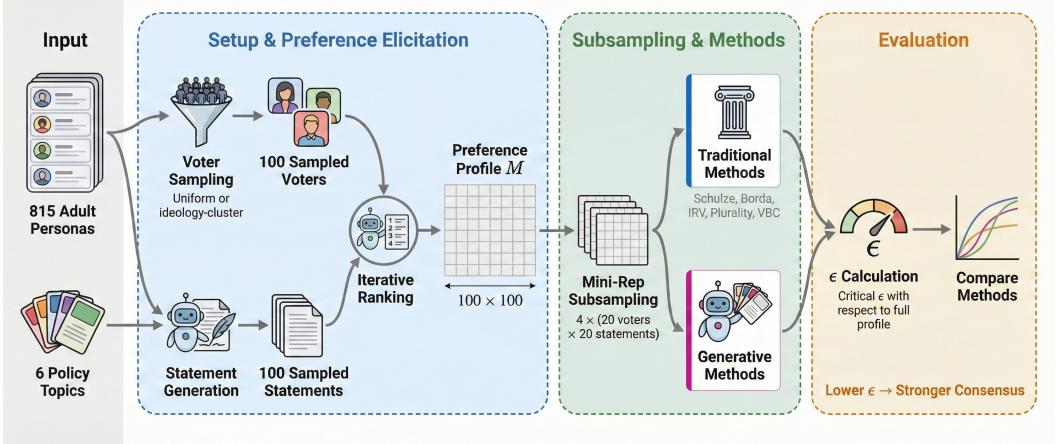


Fig. 1. Our experiment setup. We generate statements conditioned on synthetic persona, and build preference profiles from voters conditioned on synthetic persona. We then evaluate various voting rules, including those powered by LLMs, and compare their critical ϵ s.

For each experiment, we first use the persona dataset of [Castricato et al. \[2024\]](#) to select the 100 distinct voters in \mathcal{N} . We then generate a different alternative for each voter for a total of 100 alternatives. We let \mathcal{D} be the uniform distribution over these alternatives, to capture a distribution defined by first choosing a random voter, and then generating a statement based on that user’s preferences, as discussed in Section 2.

As in our theoretical model, the voting rules in our experiments only have access to a random subset of alternatives and voters, specifically 20 of each. Of course, practical situations may have more than 100 voters and alternatives, and one would want to sample as many as possible, but this small-scale experiment allows us to run many experiments with LLMs at reasonable cost. We compute the critical ϵ with respect to all 100 alternatives and voters to measure how well the alternative selected by each voting rule does at finding common ground for the overall population. Recall that the critical ϵ (Definition 2.2) measures the smallest ϵ for which the alternative is in the ϵ -PVC. Therefore, a method that is better at finding common ground will choose alternatives with critical ϵ close to 0 while a method that is worse at finding common ground will choose alternatives with relatively large critical ϵ . Upon publication, we will release our implementation of various ϵ -PVC computations as open source.

7.1 Methodology

Our experiment setup is illustrated in Figure 1. For each of six topics, we perform 10 replications, each with 100 voters and 100 alternatives drawn from the persona dataset. We then obtain a ranking of the generated alternatives for each voter to form the overall preference profile. From there, we subsample to get 20 voters and 20 statements yielding a 20×20 preference profile, which we do 4 times per replication. Finally, various voting rules are run on this 20×20 preference profile, and we compare the critical ϵ s of the selected alternatives (with respect to the original 100×100 preference profile). We elaborate on the details of each step of the experiment below.

Topics. We originally formulated 13 topics covering contemporary political issues (Appendix D.1.1). To study if certain voting rules do better or worse on polarized or less polarized questions, we rate each topic by degree of polarization. For each topic, we generate 100 statements and sample

100 voters independently to assign a Likert score from 0-10 to each statement indicating how much they agree with that statement (Figure 4). We then compute the mean Likert score and take its absolute difference from 5 to measure how strongly voters feel about different topics, i.e. how polarizing topics are. Finally, we selected the top three and bottom three topics by this metric.

Personas. We use the 1,000 synthetic personas generated by Castricato et al. [2024]. Their choice of personas is informed by the US census and intended to be ideologically and demographically diverse. Each persona contains comprehensive personal characteristics, such as race, occupation, and how they spend their personal time. An example of a persona can be found in Appendix D.2. We filter the personas to those above the age of 18, thus obtaining 815 adult personas.

Alternatives. For each persona and topic, we generate a representative statement from that persona for that topic by querying an LLM (specifically, gpt-5-mini). We explore other methods to generate alternatives in Appendix D.3.

Preference Profiles. We construct preference profiles by asking each voter (powered by gpt-5-mini and conditioned on a persona) to sequentially rank their top 10 and bottom 10 statements until all 100 statements are ranked. We term this approach *iterative ranking* and discuss further details in Appendix D.4. Each preference profile consists of 100 voters and 100 statements.

Calculating Critical Epsilons. We compute the critical epsilons for each statement using Algorithm 1 (where the critical ϵ is with respect to the full 100×100 preference profile and the uniform distribution over these alternatives).

7.2 Voting Rules

In this section, we compare the critical epsilon of the statements selected by various voting rules. Veto-By-Consumption (VBC) [Ivanovski and Kondratev, 2023], like Algorithm 2, finds an element of the PVC for the subsampled preference profile. We then compare VBC to other standard voting rules, including Borda count, Schulze [Schulze, 2011], Instant Runoff Voting (IRV), and plurality. We also include a baseline that is the critical ϵ of a uniform randomly chosen statement among the 100. Figure 2 shows the cumulative distribution function (CDF) of critical epsilons across all iterations for the three controversial topics, and Table 1 shows the mean critical epsilons for all six topics. CDF graphs for the other topics and additional figures can be found at Appendix D.7. Across topics there is a consistent pattern: the voting rules ranked in increasing order of critical epsilons are:

$$\text{VBC} < \text{Borda} < \text{Schulze} < \text{IRV} < \text{Plurality}. \quad (2)$$

As expected from our theoretical results, even with access to only a small number of voters and alternatives, finding an element of the PVC for the subset gives an alternative with very low critical ϵ for the full sets. The critical ϵ for VBC is almost always 0, with mean consistently less than 0.001 across topics. This supports using methods for finding elements of the PVC in practical situations even without having access to full information.

Unsurprisingly, the other voting rules select alternatives that have higher critical ϵ . For example, Schulze (used by the Habermas machine) selects alternatives with an order of magnitude higher critical epsilon than VBC. Perhaps more surprisingly, plurality consistently selects alternatives with higher critical ϵ than just choosing a random alternative. The social choice function that consistently does the best after VBC is Borda count. In practice, Borda count could be a good compromise as a scoring rule that is easily computable, interpretable, and also chooses alternatives with small critical ϵ .

In addition to these synthetic experiments for preferences over textual statements, we also tested how well each of these voting rules perform on preference datasets from Preflib [?], without

subsampling, i.e., when the voting rule has full access to voters and alternatives. In these real-world scenarios, we observe the same pattern of Equation (2), with Borda count consistently having the lowest critical ϵ among the standard voting rules and Plurality consistently having the highest. See Appendix D.5 for more details.

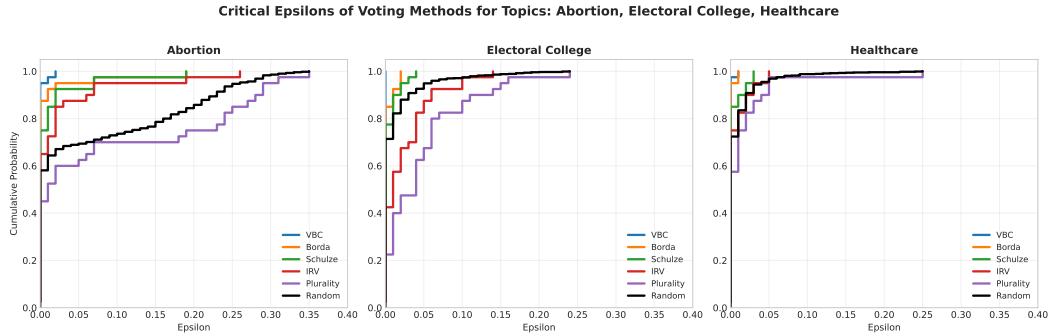


Fig. 2. CDF of critical epsilons for the chosen alternatives of various voting rules for the three polarizing topics. The smaller the epsilons (the closer the line is to the top-left corner), the better at finding common ground the voting method is.

Table 1. Mean critical epsilons for various voting rules across various topics. The smallest epsilon for each topic is bolded.

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	0.0008	0.0000	0.0003	0.0003	0.0000	0.0014
Borda	0.0075	0.0023	0.0005	0.0015	0.0014	0.0025
Schulze	0.0108	0.0040	0.0030	0.0040	0.0033	0.0033
IRV	0.0203	0.0233	0.0063	0.0133	0.0086	0.0050
Plurality	0.0865	0.0460	0.0158	0.0268	0.0142	0.0128
Random	0.0609	0.0108	0.0082	0.0103	0.0108	0.0072

7.3 Generative Voting Rules

In this section we examine *generative voting rules*, which do not have to select among the existing alternatives and instead can use an LLM-generated new alternative. We consider several different generative voting rules that each use different amounts of information about the problem to generate common-ground statements:

- GPT-Blind: the model is only given the topic and no information about voters or alternatives
- GPT-Synthesize: the model is given the topic and the 20 alternatives.
- GPT-Synthesize+Rankings: the model is given the topic, the 20 alternatives, and the 20×20 preference profile.
- GPT-Synthesize+Personas: the model is given the topic and the persona descriptions of the 20 voters.

To keep the values comparable, we do not compute the critical epsilon of a generated statement with respect to the uniform distribution over 101 statements. Instead, we continue to consider the

uniform distribution over the 100 original statements, and treat the generated statement as having zero probability mass in \mathcal{D} .

Table 2 shows the mean critical ϵ for these different generative voting rules. From this table, we can see that the generative voting rules do generate alternatives with smaller critical ϵ than many standard voting rules like Schulze, IRV, and Plurality. VBC and Borda both slightly outperform the generative voting rules despite the fact that the generative methods can produce their own alternatives. Overall, the generative voting rules choose alternatives with relatively low critical ϵ , indicating that what an LLM believes represents common ground is well aligned with the concept of the PVC.

Table 2. Mean critical epsilons for VBC, generative voting rules and random baselines across various topics. Smallest epsilon for each topic is bolded. Under “Random Insertion,” we sample a random statement from the 815 global statements that are not in the 100×100 preference profile and use the same insertion program that we use for the generative methods.

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	0.0008	0.0000	0.0003	0.0003	0.0000	0.0014
GPT-Blind	0.0103	0.0093	0.0080	0.0045	0.0085	0.0200
GPT-Synthesize	0.0055	0.0108	0.0090	0.0128	0.0168	0.0238
GPT-Synth+Rank.	0.0030	0.0053	0.0038	0.0025	0.0080	0.0095
GPT-Synth.+Pers.	0.0030	0.0060	0.0040	0.0040	0.0058	0.0105
Random Insertion	0.0700	0.0100	0.0108	0.0100	0.0073	0.0110
Random	0.0609	0.0108	0.0082	0.0103	0.0108	0.0072

In addition to generative LLM-based methods, we also evaluate LLM-based methods that are still restricted to choosing one of the original 100 alternatives. We discuss these methods and results in more detail in Appendix D.8.

Clustered Voter Populations. In the previous section, we showed that the generative voting rules are able to generate alternatives with very low critical ϵ , and the low numbers make it difficult to distinguish between the generative and non-generative methods. As LLMs are post-trained to be aligned to the general population, we hypothesize that generative voting rules will struggle in elections where the voter base is skewed to a particular ideology. To investigate this, we partition the 815 adult personas into 431 progressives and 255 conservatives by keyword filtering, with 129 unmatched personas (see Appendix D.9 for details). We plot our results for conservative voters in Figure 3 and report the mean critical epsilon in Table 3. We find that generative voting rules degraded significantly compared to Table 2 and sometimes perform worse than the random baseline, while VBC maintains its performance of having most critical epsilons close to 0.

We also find that providing more information to the generative voting rules helps them significantly. Interestingly, providing personas of the voters is the most helpful to the generative voting rules, but with this information the mean critical epsilon is still 10 times larger than that of VBC. Due to space constraints, we report the results for progressive voters in Table 12 in Appendix D.9.3.

8 Discussion

One of the main theoretical contributions of this paper is analyzing the sample complexity of finding alternatives in the ϵ -PVC using different types of queries. While we studied certain forms of generative and discriminative queries, there are many other ways to elicit information about both alternatives and voters. A natural open question is whether there are other reasonable query

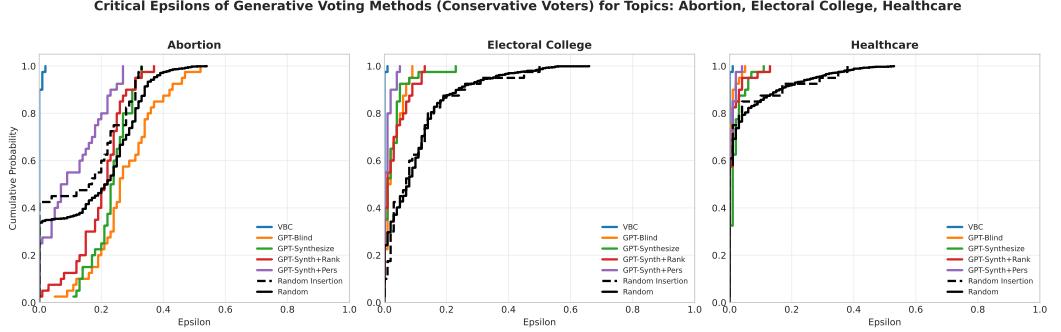


Fig. 3. Critical epsilons for the winners of various generative voting rules on conservative voters for polarizing topics. Generative voting rules consistently do worse than VBC and sometimes even the random baseline.

Table 3. Mean critical epsilons for VBC, generative voting rules and random baselines across various topics for conservative voters. Smallest epsilon for each topic is bolded.

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	0.0013	0.0003	0.0003	0.0000	0.0003	0.0000
GPT-Blind	0.2775	0.0263	0.0050	0.0225	0.0843	0.0220
GPT-Synthesize	0.2313	0.0263	0.0178	0.0393	0.0230	0.0305
GPT-Synth+Rank.	0.1980	0.0295	0.0120	0.0215	0.0280	0.0048
GPT-Synth.+Pers.	0.1065	0.0088	0.0048	0.0033	0.0065	0.0010
Random Insertion	0.1360	0.1013	0.0418	0.0640	0.0153	0.0135
Random	0.1751	0.0989	0.0428	0.0625	0.0382	0.0330

models that are more effective at finding elements of the ϵ -PVC. For example, different forms of discriminative or generative queries may be able to reduce the complexity of identifying an element of the ϵ -PVC and close the $1/\epsilon$ to $1/\epsilon^2$ gap we showed in Section 5.

One of our key assumptions is that the number of alternatives is large or infinite, a setting in which many voting rules are not well-defined. Therefore, another open question is how to generalize other voting rules to the setting of infinite alternatives. For example, despite the PVC technically requiring a number of constraints which is exponential in $|N|$, Theorem 4.1 shows that the number of samples needed to find an element of the ϵ -PVC does not depend on the distribution or the number of voters. One could ask whether other voting rules also have such shortcuts in the infinite alternative setting, and whether similar sample complexity results hold.

Finally, we use the critical ϵ to measure the degree to which a given statement represents common ground for the population. However, there may be many alternatives with critical ϵ equal to 0, in which case it may not be obvious how to choose among them. One practical way to break ties would be to choose the alternative from the ϵ -PVC with the highest score according to some social choice scoring function. For example, we could select the alternative with the highest Borda score among all alternatives in the ϵ -PVC. If we take the Borda score as a proxy for maximizing welfare, then the Borda winner in the ϵ -PVC can be interpreted as the common-ground alternative with the highest welfare. One direction for future work is to further explore the best way to combine the ϵ -PVC with different scoring functions to select alternatives with small critical ϵ and high welfare.

References

Parand A Alamdari, Soroush Ebadian, and Ariel D Procaccia. 2024. Policy aggregation. *Advances in Neural Information Processing Systems* 37 (2024), 68308–68329.

M. Bakker, M. Chadwick, H. Sheahan, M. Tessler, L. Campbell-Gillingham, J. Balaguer, N. McAleese, A. Glaese, J. Aslanides, M. Botvinick, and C. Summerfield. 2022. Fine-Tuning Language Models to Find Agreement Among Humans With Diverse Preferences. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Niclas Boehmer, Robert Bredereck, Piotr Faliszewski, and Rolf Niedermeier. 2022. A Quantitative and Qualitative Analysis of the Robustness of (Real-World) Election Winners. In *Proceedings of the 2nd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO '22)*. ACM, 7:1–7:10. <https://doi.org/10.1145/3551624.3555292>

Niclas Boehmer, Sara Fish, and Ariel D. Procaccia. 2025. Generative Social Choice: The Next Generation. In *Proceedings of the 42nd International Conference on Machine Learning*.

Niclas Böhmer. 2023. *Application-oriented Collective Decision Making: Experimental Toolbox and Dynamic Environments*. Ph. D. Dissertation. Technische Universität Berlin. <https://doi.org/10.14279/depositonce-19123>

Louis Castricato, Nathan Lile, Rafael Rafailov, Jan-Philipp Fränken, and Chelsea Finn. 2024. PERSONA: A Reproducible Testbed for Pluralistic Alignment. arXiv:arXiv:2407.17387 [cs.CL]

Bhaskar Ray Chaudhury, Aniket Murhekar, Zhuowen Yuan, Bo Li, Ruta Mehta, and Ariel D Procaccia. 2024. Fair federated learning via the proportional veto core. In *Forty-first International Conference on Machine Learning*.

Computational Democracy Project. 2025. Group Informed Consensus. <https://compdemocracy.org/group-informed-consensus/> Accessed: 2026-02-08.

Vincent Conitzer, Rachel Freedman, Jobst Heitzig, Wesley H. Holliday, Bob M. Jacobs, Nathan Lambert, Milan Mossé, Eric Pacuit, Stuart Russell, Hailey Schoelkopf, Emanuel Tewolde, and William S. Zwicker. 2024. Position: social choice should guide AI alignment in dealing with diverse human feedback. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Soham De, Michiel A Bakker, Jay Baxter, and Martin Saveski. 2025. Supernotes: Driving consensus in crowd-sourced fact-checking. In *Proceedings of the ACM on Web Conference 2025*. 3751–3761.

Dembrane. 2023. First Report: Democratic Inputs to AI. <https://www.dembrane.com/en-US/blog/report-openai-october-2023> Blog post describing the Common Ground application for hosting deliberations at scale.

E. A. Dinitz. 1970. Algorithm for solution of a problem of maximum flow in networks with power estimation. *Doklady Akademii Nauk SSSR* 194, 4 (1970), 754–757.

John S. Dryzek, André Bächtiger, Simone Chambers, Joshua Cohen, James N. Druckman, Andrea Felicetti, James S. Fishkin, David M. Farrell, Archon Fung, Amy Gutmann, Hélène Landemore, Jane Mansbridge, Sofie Marien, Michael A. Neblo, Simon Niemeyer, Maija Setälä, Rune Slothuus, Jane Suiter, Dennis Thompson, and Mark E. Warren. 2019. The Crisis of Democracy and the Science of Deliberation. *Science* 363, 6432 (March 2019), 1144–1146. <https://doi.org/10.1126/science.aaq0594>

Sara Fish, Paul Gölz, David C. Parkes, Ariel D. Procaccia, Gili Rusak, Itai Shapira, and Manuel Wüthrich. 2026. Generative Social Choice. *J. ACM* (2026). To Appear.

James S. Fishkin. 2009. *When the People Speak: Deliberative Democracy and Public Consultation*. Oxford University Press.

Michael R. Garey and David S. Johnson. 1990. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., USA.

Luise Ge, Daniel Halpern, Evi Micha, Ariel D Procaccia, Itai Shapira, Yevgeniy Vorobeychik, and Junlin Wu. 2024. Axioms for ai alignment from human feedback. *Advances in Neural Information Processing Systems* 37 (2024), 80439–80465.

Paul Gölz, Nika Haghtalab, and Kunhe Yang. 2025. Distortion of AI Alignment: Does Preference Optimization Optimize for Preferences? arXiv preprint arXiv:2505.23749 (2025).

Jairo F Gudiño, Umberto Grandi, and César Hidalgo. 2024. Large language models (LLMs) as agents for augmented democracy. *Philosophical Transactions A* 382, 2285 (2024), 20240100.

Daniel Halpern, Ariel D. Procaccia, and Warut Suksompong. 2025. The proportional veto principle for approval ballots. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI '25)*. Article 434, 8 pages. <https://doi.org/10.24963/ijcai.2025/434>

Egor Ianovski and Aleksei Y. Kondratev. 2023. Computing the proportional veto core. <https://doi.org/10.48550/arXiv.2003.09153> arXiv:2003.09153 [cs].

A. Kahng, M. K. Lee, R. Noothigattu, A. D. Procaccia, and C.-A. Psomas. 2019. Statistical Foundations of Virtual Democracy. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*. 3173–3182.

Andrew Konya, Lisa Schirch, Colin Irwin, and Aviv Ovadya. 2023. Democratic Policy Development Using Collective Dialogues and AI. arXiv:2311.02242 [cs].

Helene Landemore. 2022. Can AI bring deliberative democracy to the masses. In *HAI Weekly Seminar, Working Paper*. 166–191.

Hélène Landemore. 2024. Can Artificial Intelligence Bring Deliberation to the Masses? In *Conversations in Philosophy, Law, and Politics* (1 ed.), Ruth Chang and Amia Srinivasan (Eds.). Oxford University PressOxford, 39–69. <https://doi.org/10.1093/9780192893215.003.0001>

<https://doi.org/10.1093/oso/9780198864523.003.0003>

Amber Hye-Yon Lee. 2022. Social Trust in Polarized Times: How Perceptions of Political Polarization Affect Americans' Trust in Each Other. *Political Behavior* 44, 3 (Sept. 2022), 1533–1554. <https://doi.org/10.1007/s11109-022-09787-1>

N. Mattei and T. Walsh. 2013. PrefLib: A Library for Preferences. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT)*. 259–270.

Jennifer McCoy, Tahmina Rahman, and Murat Somer. 2018. Polarization and the Global Crisis of Democracy: Common Patterns, Dynamics, and Pernicious Consequences for Democratic Polities. *American Behavioral Scientist* 62, 1 (Jan. 2018), 16–42. <https://doi.org/10.gdfg7m>

Sammy McKinney and Claudia Chwalisz. 2025. *Five Dimensions of Scaling Democratic Deliberation: With and beyond AI*. Technical Report. DemocracyNext.

Herve Moulin. 1981. The Proportional Veto Principle. *The Review of Economic Studies* 48, 3 (July 1981), 407. <https://doi.org/10.2307/2297154>

Hervé Moulin. 1982. Voting with proportional veto power. *Econometrica: Journal of the Econometric Society* (1982), 145–162.

OECD. 2020. *Innovative Citizen Participation and New Democratic Institutions: Catching the Deliberative Wave*. Organisation for Economic Co-operation and Development. <https://doi.org/10.1787/339306da-en>

Yunus Emre Orhan. 2022. The Relationship between Affective Polarization and Democratic Backsliding: Comparative Evidence. *Democratization* 29, 4 (May 2022), 714–735. <https://doi.org/10.1080/13510347.2021.2008912>

Aviv Ovadya and Luke Thorburn. 2023. *Bridging Systems: Open Problems for Countering Destructive Divisiveness across Ranking, Recommenders, and Governance*. Knight First Amendment Institute at Columbia University. <https://knightcolumbia.org/content/bridging-systems> Essays and Scholarship.

James A. Piazza. 2023. Political Polarization and Political Violence. *Security Studies* 32, 3 (May 2023), 476–504. <https://doi.org/10.1080/09636412.2023.2225780>

Grzegorz Pierczyński and Stanislaw Szufa. 2024. Single-Winner Voting with Alliances: Avoiding the Spoiler Effect. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Virginia Dignum (Eds.), 1567–1575. <https://doi.org/10.5555/3635637.3663017>

Florenz Plassmann and T. Nicolaus Tideman. 2014. How Frequently Do Different Voting Rules Encounter Voting Paradoxes in Three-Candidate Elections? *Social Choice and Welfare* 42, 1 (2014), 31–75. <https://doi.org/10.1007/s00355-013-0720-8>

Ariel Procaccia, Benjamin Schiffer, Serena Wang, and Shirley Zhang. 2025b. Metritocracy: Representative Metrics for Lite Benchmarks. *arXiv preprint arXiv:2506.09813* (2025).

Ariel D. Procaccia, Benjamin Schiffer, and Shirley Zhang. 2025a. Clone-Robust AI Alignment. In *Proceedings of the 42nd International Conference on Machine Learning*, Vol. 267. 49903–49926.

David Rozado. 2024. The Political Preferences of LLMs. *PLOS ONE* 19, 7 (July 2024), e0306621. <https://doi.org/10.1371/journal.pone.0306621>

Markus Schulze. 2011. A New Monotonic, Clone-Independent, Reversal Symmetric, and Condorcet-Consistent Single-Winner Election Method. *Social Choice and Welfare* 36, 2 (Feb. 2011), 267–303. <https://doi.org/10.1007/s00355-010-0475-4>

Laura Silver. 2022. Most across 19 countries see strong partisan conflicts in their society, especially in South Korea and the U.S. Pew Research Center. <https://www.pewresearch.org/short-reads/2022/11/16/most-across-19-countries-see-strong-partisan-conflicts-in-their-society-especially-in-south-korea-and-the-u-s/> Accessed: 2026-01-31.

Isaac Slaughter, Axel Peytavin, Johan Ugander, and Martin Saveski. 2025. Community notes reduce engagement with and diffusion of false information online. *Proceedings of the National Academy of Sciences* 122, 38 (2025), e2503413122.

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

Christopher T. Small, Ivan Vendrov, Esin Durmus, Hadjar Homaei, Elizabeth Barry, Julien Cornebise, Ted Suzman, Deep Ganguli, and Colin Megill. 2023. Opportunities and Risks of LLMs for Scalable Deliberation with Polis. *arXiv:2306.11932* [cs]

Michael Henry Tessler, Michiel A. Bakker, Daniel Jarrett, Hannah Sheahan, Martin J. Chadwick, Raphael Koster, Georgina Evans, Lucy Campbell-Gillingham, Tantum Collins, David C. Parkes, Matthew Botvinick, and Christopher Summerfield. 2024. AI Can Help Humans Find Common Ground in Democratic Deliberation. *Science* 386, 6719 (Oct. 2024), eadq2852. <https://doi.org/10.1126/science.adq2852>

T. Nicolaus Tideman and Florenz Plassmann. 2012. Modeling the Outcomes of Vote-Casting in Actual Elections. In *Electoral Systems: Paradoxes, Assumptions, and Procedures*, Dan S. Felsenthal and Moshé Machover (Eds.). Springer, Berlin, 217–251. https://doi.org/10.1007/978-3-642-20441-8_9

T. Nicolaus Tideman and Florenz Plassmann. 2014. Which Voting Rule Is Most Likely to Choose the "Best" Candidate? *Public Choice* 158, 3 (2014), 331–357. <https://doi.org/10.1007/s11127-012-9935-y>

Stefan Wojcik, Sophie Hilgard, Nick Judd, Delia Mocanu, Stephen Ragain, MB Hunzaker, Keith Coleman, and Jay Baxter. 2022. Birdwatch: Crowd wisdom and bridging algorithms can inform understanding and reduce the spread of misinformation. *arXiv preprint arXiv:2210.15723* (2022).

Guanhua Zhang and Moritz Hardt. 2024. Inherent trade-offs between diversity and stability in multi-task benchmarks. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. Article 2435.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. 2025. Verbalized Sampling: How to Mitigate Mode Collapse and Unlock LLM Diversity. arXiv:2510.01171 [cs.CL] <https://arxiv.org/abs/2510.01171>

AI Usage Disclosure

Figure 1 was generated using the PaperBanana framework, where we prompted Nano Banana Pro with our experiment methodology. As the code for PaperBanana was not published, we created customized "Gems" in the Gemini workspace to simulate the framework using the system prompts provided in the PaperBanana paper. We iterated the figure about 40 times until we arrived at our final figure. We post-processed the figure by removing the Gemini watermark. Separately, the code for the project was developed within Cursor, where Claude models were used to assist with planning, code generation, and debugging. AI was also used sparingly in the writing for proofreading and formatting.

Appendix

A Reduction of ϵ -PVC to PVC

The standard definition of PVC (as in [Moulin, 1982]) is the following.

Definition A.1. Let there be a set of voters N with $|N| = n$, and a set of alternatives M with $|M| = m$. Each voter i has a ranking σ_i over all alternatives. Let the veto function $f(x, n, m)$ for some positive integer $x \leq n$ be defined as follows:

$$f(x) = \left\lceil \frac{x}{n} \cdot m - 1 \right\rceil.$$

An alternative a is vetoed if there exists some group of voters $T \in N$ such that there exists a set S of alternatives with $|S| \geq m - f(|T|)$ where every voter $i \in T$ prefers every alternative $b \in S$ to a . The set of alternatives which are not vetoed is called the *proportional veto core*.

Now consider Definition 2.1 with $\mathcal{N} = N$, $\mathcal{M} = M$, $\epsilon = 0$, and \mathcal{D} as the uniform distribution over M . We will show this is equivalent to Definition A.1. Define $n = |N|$ and $m = |M|$. In this choice of parameters for Definition 2.1, an alternative $a \in M$ is blocked if there exists a $T \subseteq N$ and $S \subseteq M$ such that

$$\frac{|S|}{m} > 1 - \frac{|T|}{n}.$$

Defining $y = \frac{m|T|}{n}$, this can be rewritten as

$$|S| > m - y. \tag{3}$$

Note that $|S| > m - y$ if and only if $|S| \geq m - \lceil y - 1 \rceil$. Therefore, Equation (3) is equivalent to

$$|S| \geq m - \lceil y - 1 \rceil = m - f(|T|).$$

This is exactly the same definition of being blocked as in Definition A.1, and therefore we are done.

B Deferred Proofs

B.1 Proof of Proposition 3.1

PROOF OF PROPOSITION 3.1. A key technical tool we use in the proof of Proposition 3.1 is a linear time algorithm that, given full access to \mathcal{D} and the preferences of the voters, can return a set of alternatives that are in the ϵ -PVC. This algorithm is based on the Veto by Voting (VBV) algorithm in [Moulin, 1982] for finding the PVC when M is finite. For any instance of the problem with finite M , VBV always output an element of the PVC. Our modification allows these algorithms to generalize to the setting of ϵ -PVC with infinitely many alternatives.

Proposition 3.1 follows directly from the following lemma analyzing Algorithm 3.

LEMMA B.1. *Every alternative in the set C returned by Algorithm 3 is in the ϵ -PVC. Furthermore, $\mu_{\mathcal{D}}(C) \geq \epsilon$.*

PROOF. We first will show that the set of alternatives returned by Algorithm 3 must all be in the ϵ -PVC. Proof by contradiction. Suppose that a is returned by Algorithm 3 and that a is not in the ϵ -PVC. By definition of ϵ -PVC, there must exist a coalition T and set of alternatives S such that $w \succ_i a$ for all $i \in T$ and $w \in S$ and $\mu_{\mathcal{D}}(S) > 1 - \frac{|T|}{n} + \epsilon$. For $i \in T$, let W_i be the set of alternatives that i ranks lower than a , and let $W = \{a\} \cup (\cup_{i \in T} W_i)$. By construction, we must have that S and W are disjoint, so

$$\mu_{\mathcal{D}}(W) \leq 1 - \mu_{\mathcal{D}}(S) < \frac{|T|}{n} - \epsilon.$$

Because a is in C when the algorithm terminates, every voter in T must have removed or decreased the capacity of alternatives they like less than or equal to a , so every voter in T must have removed

ALGORITHM 3: Vote by γ -Veto

```

1 Input: A set of voters  $\mathcal{N}$  and a set of alternatives  $\mathcal{M}$ 
2 Output: A set of alternatives.
3 Let  $\gamma = \frac{1-\epsilon}{n}$ .
4 Assign each alternatives  $a \in \mathcal{M}$  capacity  $c_a = \mu_{\mathcal{D}}(a)$ .
5 Initialize  $C = \mathcal{M}$ 
6 for voter  $i \in \mathcal{N}$  do
7   Voter  $i$  identifies the smallest set  $X \subseteq C$  with the following properties. First, for every  $x \in X$  and
     $y \in C, y \succ_i x$ . Second, defining  $X_0 = \{x \in X : \mu_{\mathcal{D}}(x) = 0\}$ ,
    
$$\mu_{\mathcal{D}}(X_0) + \sum_{x \in X \setminus X_0} c_x \geq \gamma.$$

8   Remove all alternatives in  $X$  from  $C$  except the one most highly ranked by voter  $i$  and call the most
    highly ranked alternative  $a$ .
9   If  $c_a \leq \gamma - \mu_{\mathcal{D}}(X \setminus \{a\})$ , then remove  $a$  from  $C$ . Otherwise, let  $c_a = c_a - (\gamma - \mu_{\mathcal{D}}(X \setminus \{a\}))$ .
10 end
11 return  $C$ 

```

or decreased the capacity of alternatives in W . Each voter in S consumes γ mass, therefore the total amount of mass that is consumed from W must be at least

$$\gamma|T| = \frac{|T|(1-\epsilon)}{n} = \frac{|T|}{n} - \frac{|T|}{n}\epsilon \geq \frac{|T|}{n} - \epsilon > \mu_{\mathcal{D}}(W).$$

Therefore, all alternatives in W must have been removed from C by the time that the algorithm terminated, and therefore a must have been removed from C . This is a contradiction with the fact that a is returned by the algorithm, which completes the first part of the proof.

To see that $\mu_{\mathcal{D}}(C) \geq \epsilon$, define

$$m(C) := \mu_{\mathcal{D}}(\{x \in C : \mu_{\mathcal{D}}(x) = 0\}) + \sum_{x \in C : \mu_{\mathcal{D}}(x) \neq 0} c_x.$$

By construction, $m(C)$ decreases by exactly γ in each round of the for loop. Since $\gamma = \frac{1-\epsilon}{n}$ and there are n rounds of the for loop, this implies that $m(C) = \epsilon$ when the algorithm terminates. Furthermore, by definition $m(C) \leq \mu_{\mathcal{D}}(C)$, therefore $m(C) = \epsilon$ implies that $\mu_{\mathcal{D}}(C) \geq \epsilon$ as desired. \square

\square

B.2 Alternative Proof of Theorem 5.4

In this section, we present an alternative proof for Theorem 5.4 that achieves a tighter lower bound of $mn/2$ (improvement over $mn/32$ as shown in the proof in Section 5). Note that our algorithm using Median of Medians only uses $\approx 3.33mn$ pairwise discriminative queries, and therefore this tighter lower bound is only a factor of ≈ 6.66 from being fully tight.

SECOND PROOF OF THEOREM 5.4. Let $m = n + 1$.

First, assume that we know we are in one of two cases:

- Alternative a is ranked second to last in every single ranking, and every alternative in $A \setminus \{a\}$ is ranked last in exactly one ranking
- Alternative a is ranked second to last by $n - 1$ voters and is ranked last in exactly one ranking, in which some alternative x is ranked second to last. Every alternative in $A \setminus \{a, x\}$ is ranked last in exactly one ranking.

In the first option above, the PVC is just $\{a\}$. In the second option above, the PVC is just $\{x\}$. Therefore, in order to find the proportional veto core, the algorithm must be able to distinguish between these two cases.

We will show that no algorithm can distinguish between these two cases with fewer than $\frac{nm}{2} = \frac{n(n+1)}{2}$ queries by giving an adversary that requires at least this many queries.

Define $X = A \setminus \{a\}$. We therefore will consider the easier problem of determining if every element of X is ranked last exactly once in N . Note that any queries not involving a will not give any information for this alternate problem, therefore we will only consider queries involving a .

We will prove the desired result using induction. Let Q be a set of queries of the form (x, a, y) for $x \in X$ and $y \in N$ where $x \succ_y a$. For any X, N, Q such that $|X| = |N|$, define $f(X, N, Q)$ as the minimum number of queries in this adversarial setting necessary to determine whether or not every element in X is ranked last in exactly one ranking in N assuming that we start by doing the queries in Q .

Base case: If $|X| = |N| = 1$ then X and N are both singletons (i.e. $X = \{x\}$ and $N = \{y\}$). If $|Q| \geq 1$, then clearly $f(X, N, Q) \geq |Q| \geq 1$. If $Q = \emptyset$, then we then need one query of (x, a, y) to determine whether or not x is ranked last by y . Therefore, for any X, N such that $|X| = |N| = 1$ and any Q , we have that $f(X, N, Q) \geq 1 = \frac{|N|(|N|+1)}{2}$. Inductive Hypothesis: Assume that $f(X, N, Q) \geq \frac{|N|(|N|+1)}{2}$ for all X, N, Q such that $|X| = |N| < n$.

Consider any X, N, Q such that $|X| = |N| = n$.

Case 1: There exists a set $X_1 \subset X$ and $Y_1 \subset N$ such that $|X_1| = |Y_1| < n$ and such that the only possible way to assign every alternative in X to be last in some ranking in N is to assign every alternative in X_1 to be last in some ranking in Y_1 . Define $Y_2 = N \setminus Y_1$ and $X_2 = X \setminus X_1$. Let $Q_1 \subset Q$ be the set of all queries in Q involving alternatives in X_1 and voters in Y_1 and let $Q_2 \subset Q$ be the set of all queries in Q involving alternatives in X_2 and voters in Y_2 . We will show that

$$f(X, N, Q) \geq f(X_1, Y_1, Q_1) + f(X_2, Y_2, Q_2) + |X_1| \cdot (n - |X_1|). \quad (4)$$

To see this result, note that since X_1 alternatives must be ranked last in some ranking in Y_1 and alternatives in X_2 must be ranked last in some ranking in Y_2 , from now on the player only needs to ask queries involving alternatives in X_1 and voters in Y_1 or ask queries involving alternatives in X_2 and voters in Y_2 . This explains the first two recursive terms in the above equation. To show the last additive term, we must show that the number of queries in $Q \setminus Q_1 \setminus Q_2$ (or equivalently the number of queries involving alternatives in X_1 and voters in Y_2 or involving alternatives in X_2 and voters in Y_1) is at least $|X_1| \cdot (n - |X_1|)$.

This is because in order for such an X_1 and Y_2 to exist, we must be in one of the following two situations:

- Every alternative $x_1 \in X_1$ has been compared to a in every ranking $y_2 \in Y_2$ and $x_1 \succ_{y_2} a$
- Every alternative $x_2 \in X_2$ has been compared to a in every ranking $y_1 \in Y_1$ and $x_2 \succ_{y_1} a$

Option 1 above requires $|X_1||Y_2|$ comparisons in $Q \setminus Q_1 \setminus Q_2$ and option 2 requires $|X_2||Y_1|$ comparisons in $Q \setminus Q_1 \setminus Q_2$. In both cases, we must have that

$$|Q \setminus Q_1 \setminus Q_2| \geq |X_1| \cdot (n - |X_1|).$$

Therefore, we have shown Equation (4).

Using the inductive hypothesis, Equation (4) implies that

$$\begin{aligned}
 f(X, N, Q) &\geq \frac{|X_1|(|X_1| + 1)}{2} + \frac{|X_2|(|X_2| + 1)}{2} + |X_1|(n - |X_1|) \\
 &= \frac{|X_1|(|X_1| + 1)}{2} + \frac{|X_2|(|X_2| + 1)}{2} + |X_1||X_2| \quad (|X_2| = n - |X_1|) \\
 &= \frac{|X_1|^2 + |X_1| + |X_2|^2 + |X_2| + 2|X_1||X_2|}{2} \\
 &= \frac{(|X_1| + |X_2|)^2 + (|X_1| + |X_2|)}{2} \\
 &= \frac{|N|^2 + |N|}{2} \\
 &= \frac{|N|(|N| + 1)}{2}.
 \end{aligned}$$

Case 2: If we are not in case 1, then for any query of the form $q = (x, a, y)$, suppose that the adversary responds that $x \succ_y a$. In this case, we have that (for $Q' = Q + \{q\}$)

$$f(X, Y, Q) = f(X, Y, Q').$$

There are a finite number of potential queries, so eventually we will reach some Q' where $Q \subseteq Q'$ and such that $f(X, Y, Q) = f(X, Y, Q')$ and such that X, Y, Q' falls in case 1. We already showed that in Case 1 we must have that $f(X, Y, Q') \geq \frac{n(n+1)}{2}$, so we can conclude that

$$f(X, Y, Q) = f(X, Y, Q') \geq \frac{n(n+1)}{2}.$$

In order to show that the above adversary is valid, we must maintain that at any point given the query feedback from the adversary, a could either never ranked last or a could be ranked last exactly once. Because the adversary always responds that $x \succ_y a$ for all queries, at any point it is still possible that a could be ranked last by every voter. Therefore, we must show that for this adversary, we also could have that a is not ranked last by one voter. This will follow from the following lemma. This lemma implies that any time one query response from the adversary would result in a query set Q for which a must be ranked last by every voter in N , we must be in the setting of case 1. This implies that the adversary never needs to answer a query in such a way that forces a to be ranked last by every voter until at least $n(n+1)/2$ queries have already been asked.

LEMMA B.2. *For a given (X, N, Q) , suppose that there exists a matching of alternatives in X to voters in N such that every alternative in X could be ranked last by exactly one voter in N given the results of the queries in Q . Further, assume that there exists a query (a, x, y) such that if $x \succ_y a$, then this is no longer the case. Then there must exist a set $X_1 \subset X$ and $Y_1 \subset Y$ such that $|X_1| = |Y_1| < n$ and such that the only possible way to assign every alternative in X to be last in some ranking in N is to assign every alternative in X_1 to be last in some ranking in Y_1 .*

PROOF. Because there exists a matching of alternatives in X to voters in N , choose one such mapping and call it $M : X \rightarrow N$. Create the following directed graph with vertices corresponding to X . Add a directed edge from $x_1 \in X$ to $x_2 \in X$ if and only if the alternative x_1 could be ranked last by the voter $M(x_2)$. If there exists an (x, y) as assumed in the lemma, then there cannot be a directed cycle containing the vertex x . This is because if there is a directed cycle C containing x , then there still exists a valid matching M' that satisfies $x \succ_y a$, which violates the lemma assumption. To construct M' , we could simply modify M by matching every vertex b in the cycle to $M(C(b))$.

Now we show that the lack of cycles containing x implies that there exists some subset of vertices $X_1 \subset X$ such that there are no edges from $X \setminus X_1$ to X_1 and $X_1 \subset X$. The simplest way to do this is

to reduce the graph into strongly connected components. Because there is no cycle containing x , x must be alone in its component. The strongly connected component graph is a DAG so it must have a source, which is a component with no incoming edges. Furthermore, because x is alone in its component there are at least two components. Therefore, taking X_1 to be the source component gives that there are no edges from $X \setminus X_1$ to X_1 . Taking $Y_1 = M(X_1)$ gives that X_1 and Y_1 satisfy the desired property that the only possible way to assign every alternative in X to be last in some ranking in N is to assign every alternative in X_1 to Y_1 , so we are done.

□

□

C Deferred Proofs for Computing ϵ -PVC and Critical Epsilons

C.1 Computing the ϵ -PVC and Critical Epsilons

In this section, we first discuss computing the ϵ -PVC efficiently when we have access to full information about voters and alternatives when M is finite. [Ianovski and Kondratev \[2023\]](#) showed that for the original definition of PVC, the problem of finding all statements in the PVC can be reduced to a biclique problem, for which a polynomial time algorithm is known. We adapt the proof to show that a similar reasoning applies to ϵ -PVC.

Fix an alternative $a \in M$. As we will use a graph reduction, we first present the definition of a *blocking graph*.

Definition C.1 (Blocking graph). [\[Ianovski and Kondratev \[2023\]\]](#) Given an election instance (N, M, \succ) and an alternative $a \in M$. The *blocking graph* of a is a bipartite graph where there are nm left vertices, where each vertex can be identified as one of the m copies of the n voters, and there are $n(m-1)$ right vertices, where each vertex can be identified as one of the n copies of the $m-1$ alternatives of $M \setminus \{a\}$. An edge exists from a left vertex to a right vertex if and only if the left vertex corresponds to a voter $i \in N$ and the right vertex corresponds to an alternative $c \in M \setminus \{a\}$ such that i prefers c to a .

We now present two statements and show that they are equivalent:

CONDITION C.2 (BLOCKING SET). *There exists a coalition $T \subseteq N$ and a blocking set $S \subseteq M$ such that $|T|/n > 1 - |S|/m + \epsilon$ and every voter in T prefers every alternative in S to a .*

CONDITION C.3 (BLOCKING GRAPH). *The blocking graph of a has a biclique with mk left vertices and nb right vertices such that $mk + nb > (1 + \epsilon)nm$.*

To bridge these two statements, we use $|T| = k$ and $|S| = b$. By showing that Condition C.2 is equivalent to Condition C.3, or in words, that the existence of the blocking set is equivalent to the existence of a sufficiently large biclique in the blocking graph, we can use a known polynomial time algorithm for Condition C.3 to check Condition C.2, which in turn determines membership in ϵ -PVC.

PROPOSITION C.4. *Condition C.2 is equivalent to Condition C.3.*

PROOF. (Condition C.2 \implies Condition C.3) Note that by multiplying the inequality in Condition C.3 by nm , we get $mk + nb > (1 + \epsilon)nm$. By construction of the blocking graph, the mk vertices on the left that correspond to the m copies each of the k voters in T are all connected to the nb vertices on the right that correspond to the n copies each of the b alternatives in S .

(Condition C.3 \implies Condition C.2) Take one left vertex x in the biclique. Say x corresponds to a copy of voter i . Enlarge the biclique by including all left vertices that correspond to copies of voter i . This is possible because these vertices have the same set of neighbours by construction of the

blocking graph. Do the same for every left vertex in the biclique. Similarly, do a corresponding enlargement for the right vertices in the biclique. Now, since there are at least mk left vertices, of which we included all m copies of their corresponding voters, we must have at least k voters included in the biclique. Similarly, as we have at least nb right vertices, of which we included all n copies of their corresponding alternatives, we must have at least b alternatives included in the biclique. Let these k voters be T and these b alternatives be S . Then note that $mk + nb > (1 + \epsilon)nm \implies m|T| + b|S| > (1 + \epsilon)nm \implies |T|/n > 1 - |S|/m + \epsilon$. By construction of the blocking graph, these voters in T all prefers every alternative in S to a . \square

Prior work has established that Condition C.3 can be checked in polynomial time:

PROPOSITION C.5. *[Garey and Johnson [1990], pg. 196, GT24] Let G be a bipartite graph and k an integer. We can, in polynomial time, determine whether there exists a biclique $K \subseteq G$ with x left vertices and y right vertices such that $x + y = k$.*

We can thus determine whether a given alternative is in the ϵ -PVC by using the polynomial time algorithm in Proposition C.5 and iterating k from $(1 + \epsilon)nm$ to $2mn - n$, giving us Corollary C.6.

COROLLARY C.6. *The ϵ -PVC can be computed in polynomial time.*

C.2 A Faster Algorithm through a Correspondence with Min-Cut

With proof of existence of a polynomial time algorithm out of the way, here we proceed to present the proof and runtime analysis for the faster algorithm that is used in our implementation (Algorithm 1). Many ideas for this algorithm, including the max-flow correspondence, are from [Ianovski and Kondratev \[2023\]](#)'s $O(m \max(n^3, m^3))$ algorithm for computing the PVC. The core of the idea is that instead of copies of the same vertex in a blocking graph, we turn these copies into weighted edges in a flow graph where the weight is the number of copies.

C.2.1 Correctness.

PROOF. Consider the flow graph after applying the min cut. The min cut could not cut the edges between the voter nodes and the alternative nodes as the edge weights are infinite, so the only edges possibly cut are those between the source node and the voter nodes, and those between the alternative nodes and the sink node. Note that whenever an edge is cut, that node contributes to the min-cut, while the remaining nodes after the min-cut forms a biclique of the blocking graph, which is in fact maximum. To be precise, all voter nodes and alternative nodes are partitioned into either contributing to the min-cut or the biclique. When an edge between the source and a voter node is cut, that voter node contributes m to the min-cut. When an edge between the sink and an alternative node is cut, that alternative node contributes n to the min-cut. The remaining nodes after the min-cut will form a biclique in the corresponding blocking graph. To see this, note that since the remaining voter nodes are still connected to the source, and the remaining alternative nodes are still connected to the sink, there must not be any edges left between any remaining voter nodes and alternative nodes. Otherwise, a single edge between a voter node and an alternative node, which has infinite edge weight, could transmit positive flow.⁵ Since no edges exist between the remaining voter nodes and alternative nodes, by construction, all of these voters must approve all of these alternatives over the given alternative, which is precisely the blocking set, or equivalently,

⁵The careful reader will find that the infinite edge will not transmit flow if all voter nodes are cut from the source, or all alternative nodes are cut from the sink. Indeed this is the baseline case, which is to cut all alternative nodes from the sink for a cut of $(m - 1)n$ (which is less than cutting voter nodes from the source for mn), and will yield $Q = mn \implies Q \leq (1 + \epsilon)nm \forall \epsilon \in [0, 1]$ (here Q loses its meaning as the maximum biclique, but the algorithm is still valid). By finding smaller cuts than mn , the algorithm will obtain a bigger corresponding blocking set, and raise ϵ .

the maximum biclique in the blocking graph. Now, the size of this biclique in the blocking graph involves the number of copies, which is precisely the weights of the edges leftover from the min-cut. Since the min-cut minimizes the weights cut, the constructed biclique is the biggest possible. To be precise, $Q + K = 2mn - n$, where Q is the size of the biclique, K is the min-cut, and $2mn - n$ is the total weight of the flow graph (excluding infinities), or equivalently, the total number of vertices in the blocking graph. With Q obtained, the correspondence in Proposition C.4 where $Q = mk + nb$ allows us to check ϵ -PVC membership with $Q \leq (1 + \epsilon)nm$ as desired. \square

C.2.2 Runtime. The runtime of the algorithm is chiefly that of the construction of the flow graph, and finding the min-cut. The flow graph construction takes $O(nm)$ time, since the flow graph could be fully connected. For example, consider the case where the given alternative is ranked first by all voters. The min-cut algorithm depends on implementation. By the max-flow min-cut theorem, we could use standard max-flow algorithms. Dinic's algorithm [Dinitz, 1970] requires $O(|V|^2|E|)$, which in our case where $|V| = O(m + n)$ and $|E| = O(mn)$, the algorithm runs in $O(mn(m + n)^2)$.

D Additional Experiment Details

We present additional details for the experiments described in Section 7.

D.1 Details on Topics

Below we present the long list of topics and their degree of polarization.

D.1.1 Long List of Topics.

- How should we increase the general public's trust in US elections?
- What are the best policies to prevent littering in public spaces?
- What are your thoughts on the way university campus administrators should approach the issue of Israel/Gaza demonstrations?
- What should guide laws concerning abortion?
- What balance should exist between gun safety laws and Second Amendment rights?
- What role should the government play in ensuring universal access to healthcare?
- What balance should be struck between environmental protection and economic growth in climate policy?
- What principles should guide immigration policy and the path to citizenship?
- What limits, if any, should exist on free speech regarding hate speech?
- What responsibilities should tech companies have when collecting and monetizing user data?
- What role should artificial intelligence play in society, and how should its risks be governed?
- What reforms, if any, should replace or modify the Electoral College?
- What strategies should guide policing to address bias and use-of-force concerns while maintaining public safety?

The Likert scores for each topic used to calculate degree of polarization can be found in Figure 4.

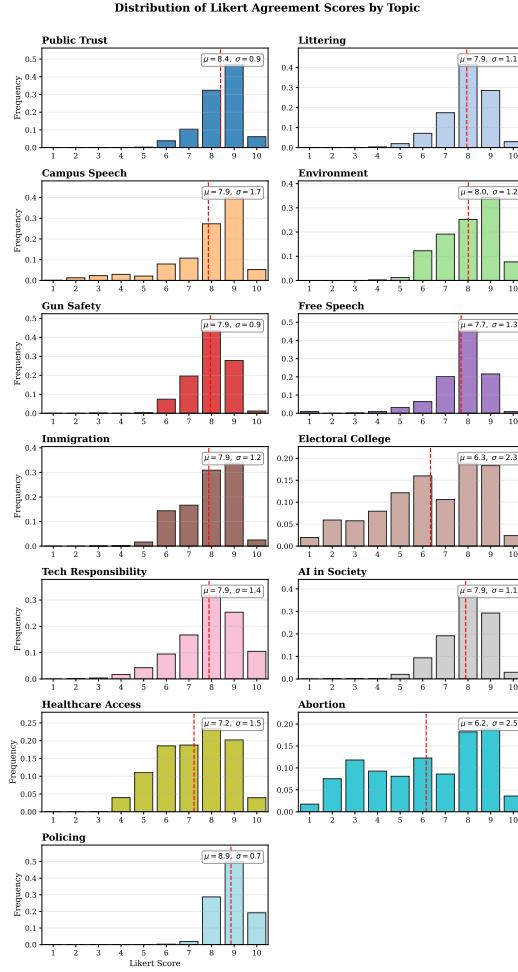


Fig. 4. Likert scores of 100 voters on 100 generated statements. Higher Likert scores indicate higher agreement. The most polarizing topics have average Likert scores close to 5 and a wide spread. The top 3 topics by polarization are abortion, Electoral College, and healthcare. The bottom 3 polarizing topics are environment, policing, and public trust in elections.

D.2 Example Persona

In this section we present an example persona that we use as a voter and for generating statements.

Example Persona

```
age: 72
sex: Female
race: White alone
ancestry: Irish
household language: English only
```

education: Bachelor's degree
 employment status: Civilian employed, at work
 class of worker: Employee of a private for-profit company or business, or of an individual, for wages, salary, or commissions
 industry category: MED-Nursing Care Facilities (Skilled Nursing Facilities)
 occupation category: MED-Registered Nurses
 detailed job description: Provides patient care in a nursing home
 income: 135500.0
 marital status: Widowed
 household type: Cohabiting couple household, no children of the householder less than 18
 family presence and age: No family under 18
 place of birth: Missouri/MO
 citizenship: Born in the United States
 veteran status: Non-Veteran
 disability: None
 health insurance: With health insurance coverage
 big five scores: Openness: Average, Conscientiousness: Average, Extraversion: Average, Agreeableness: Extremely High, Neuroticism: Low
 defining quirks: Collects vintage medical equipment
 mannerisms: Speaks with a soft, soothing voice
 personal time: Spends free time gardening or reading
 lifestyle: Active and community-oriented
 ideology: Conservative
 political views: Republican
 religion: Catholic

D.3 Alternative Methods to Generate Alternatives

D.3.1 Description of Methods to Generate Alternatives. In addition to the experiment set-up presented in the body (listed below as Persona Only), we also explored three different methods of generating statements.

- **Blind:** Given only the topic, the LLM is queried using verbalized sampling [Zhang et al., 2025] to generate diverse statements.
- **Persona Only:** Given the topic and a persona to conditioned on, the LLM is queried to produce a response. This is used for the experiments presented in the body.
- **Deliberation Round Only:** Within a rep, given 100 statements generated using Persona, the LLM is queried using verbalized sampling [Zhang et al., 2025] to generate diverse statements taking into account the 100 statements. This is to simulate a "deliberation round" where a user sees statements written by others and is informed by their stances to write a new statement.
- **Persona+Deliberation Round:** Within a rep, first generate 100 statements using Personas. Then conditioning on each persona, the LLM is queried to produce a single statement that also takes into account the other 99 statements.

In Figures 5 and 6 we show the CDFs for the same experiment as in the body but using different methods of generating alternatives. Using the Persona method gives a more interesting distribution of alternatives and has the added benefits of being relatively simple and motivated by the theory in Section 6.

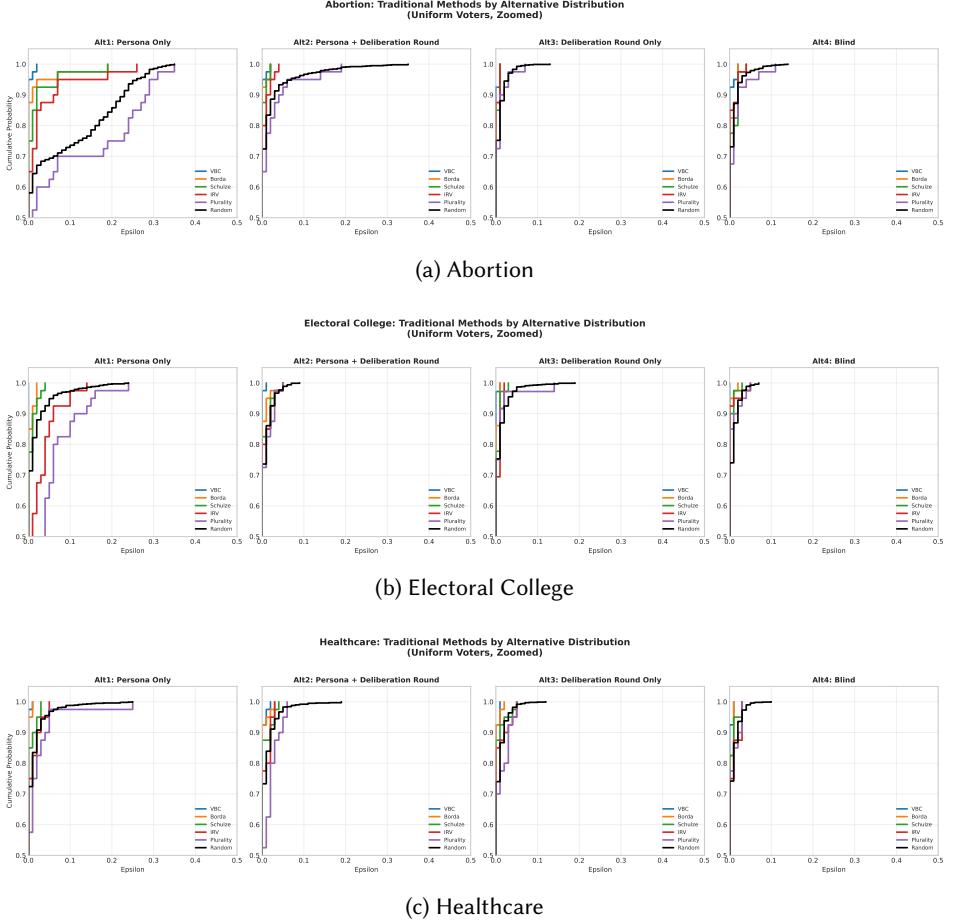


Fig. 5. CDF of critical epsilons across various topics (top three topics by degree of polarization). The leftmost subplot (Person) allows for clearer analysis than other alternative generation methods.

D.4 Building Preference Profiles

In this section, we discuss why we selected iterative ranking as our process for using an LLM to build a preference profile.

D.4.1 The Problem of Preference Degeneracy. Preference degeneracy occurs when the preference rankings produced by generative AI are very similar even for very different profiles. We found that this can happen when the AI is “overwhelmed” with the task complexity of ranking too many alternatives at once, and outputs a default ranking of the alternatives (for example, in the order presented to the AI, or its reverse).

D.4.2 Mitigating Preference Degeneracy. There are two natural ways to mitigate preference degeneracy:

- (1) Increase model capabilities.
- (2) Scaffold the task easier to make it easier for the model to complete.

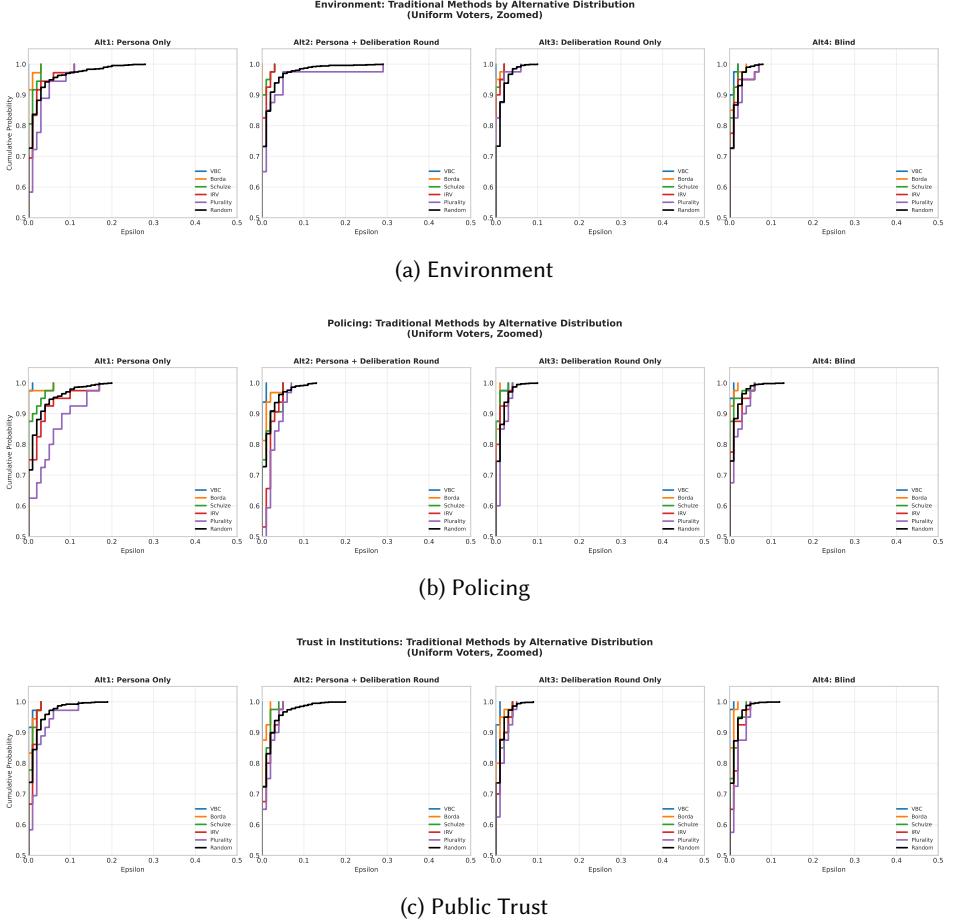


Fig. 6. CDF of critical epsilons across various topics (bottom three topics by degree of polarization). The leftmost subplot (Persona) allows for clearer analysis than other alternative generation methods.

Item 1 has the tradeoff of budget concerns. To illustrate, order of magnitude capability jumps between gpt-5-nano, gpt-5-mini, and gpt-5.2 are accompanied by similar order of magnitude price jumps. We fix the model as gpt-5-mini as a compromise between capability and price.

Another axis of increasing model capabilities is to increase reasoning effort (the length of the model’s chain-of-thought). As the number of tokens get higher, there is also an associated cost increase. Latency also increases as more tokens need to be generated before the tokens related to the extractable ranking is produced. We ran a sweep on the available options for reasoning efforts in Table 4.

There are multiple ways to go about Item 2:

Pairwise comparison. We repeatedly show the model two alternatives, and ask the model which one to pick, until a full ranking is obtained. However, if there are n voters and m alternatives, and an API call takes k tokens, then the number of tokens required (which is usually dominated by input tokens for seeding techniques) is $O(mnk \log m)$. For a reasonable experiment of $m = 100$, $n = 100$, $k = 500$, a single preference profile will take about 33 million tokens, which is about \$8.25.

Using this option, the full experiment will cost: $6 \text{ topics} \times 10 \text{ repetitions} \times 3 \text{ voter distributions} \times \$8.25/\text{profile} = \$1,485$.

Direct Ranking. The other extreme is to simply give the model all the statements and ask it to output its ranking in one API call. This is the cheapest possible approach. Unfortunately, this approach frequently gives degenerate rankings, even for the most capable models currently available.

Top-K/Bottom-K Ranking (Iterative Ranking). Given all remaining alternatives, we ask the model to pick its top K statements and bottom K statements. We then iterate for $m/2k$ rounds with the subsequent set of remaining alternatives until all alternatives have been chosen.

Scoring. Given a set of alternatives, we ask the model to give a score to each alternative. The benefit to this approach is that compared to producing the ranking, the model is not heavily constrained by other alternatives it has processed or will process when processing a new alternative. In other words, the model could ignore the scores it gave to any previous alternatives and focus on giving a score to the current alternative. When the scoring is done, a ranking is induced, and any duplicate scores could be randomly tie-broken. Another approach is to ask the model not to give duplicate scores.

Because pairwise comparisons were too expensive and direct ranking produced degenerate rankings, we focus on the remaining two methods and report results in Table 4. When we receive a ranking, we screen it for invalid rankings (e.g. hallucinated statement IDs, or duplicate statement IDs, etc). If it is invalid, we retry the query, up to 20 times.

Table 4. Degeneracy Mitigation Experiment Results

Mitigation	Reasoning Effort	Unique Rankings	Pres. Order Corr.	API Calls	Avg Time /Call
Top-K/Bottom-K	minimal	79/100	0.029	1307	3.3s
Top-K/Bottom-K	low	98/100	0.043	567	11.1s
Top-K/Bottom-K	medium	100/100	0.033	~ 500	~ 30s
Scoring	minimal	99/100	-0.126	178	8.9s
Scoring	low	100/100	-0.987	100	16.0s
Scoring	medium	98/100	-0.952	~ 100	~ 30s

Note: “Pres. Order Corr.” stands for presentation order correlations. Values near ± 1 indicate degenerate rankings.

As shown in Table 4, the scoring method frequently resulted in degenerate rankings as measured by correlation with presentation order. To minimize preference degeneracy, we use the top-K/bottom-K method for all of our experiments. We found that the low reasoning effort produces a high number of unique rankings with minimal retries (67 compared to minimal’s 807) and have better latency (11.1s vs medium’s 30s). The rankings produced with low reasoning also has a Spearman’s correlation of 0.802 with that of medium reasoning. We thus generate the preference profiles for our full experiment using the top-K/bottom-K approach (iterative ranking) using gpt-5-mini on low reasoning effort.

D.5 Using different methods to find common ground in real election instances

We validate the pattern in Equation (2) on real-world election instances from Preflib [Mattei and Walsh, 2013]. We plot our results in Figure 7.

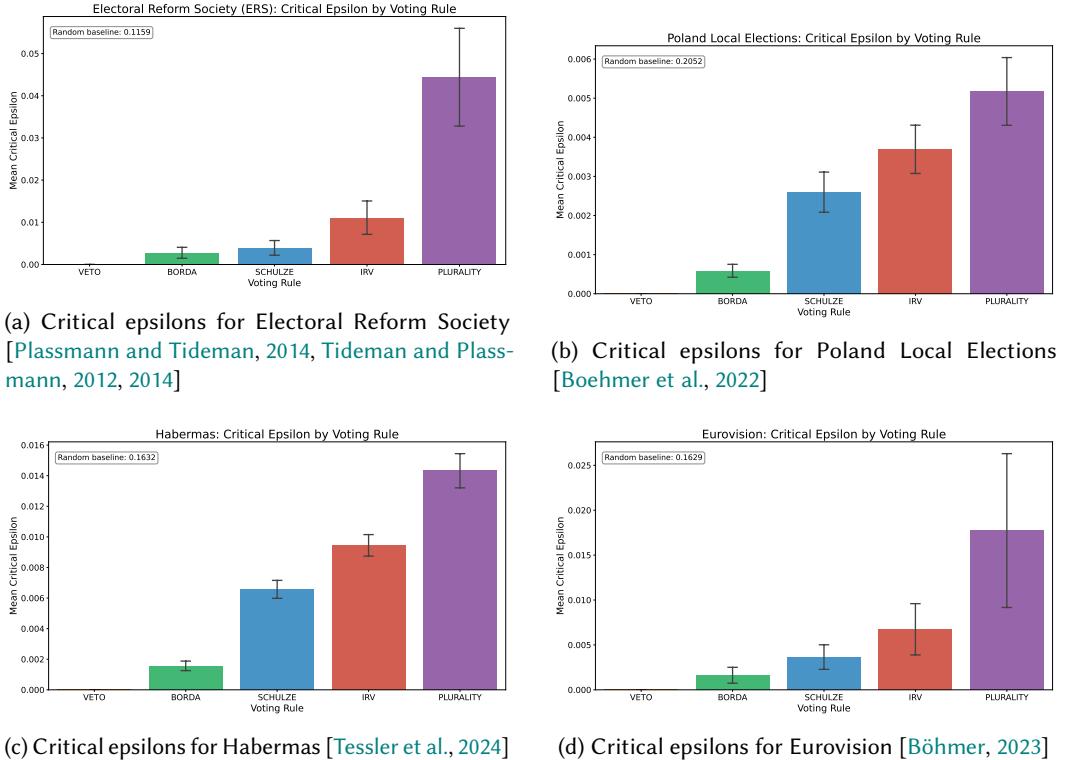


Fig. 7. Comparison of critical epsilons across voting rules on various real-world datasets from PrefLib. Standard errors depicted.

D.6 Inserting a New Statement to a Preference Profile

In this section, we discuss different methods for inserting a single alternative into an already constructed ranking of 100 existing alternatives, which is a necessary step for evaluating generative voting rules.

- Insert into 100: we give the model all 100 statements and ask the model to output the position for the new statement.
- Chunked insertion: we split the 100 statements into 5 contiguous chunks. We ask the model to output the position it will insert the new statement into for each chunk. We then tally the positions and according to its chunk assign the appropriate ranking. For example, if the new statement gets position 0, 3, 15, 8, 9 in the 5 chunks, then the insertion position is $0 + 3 + 15 + 8 + 9 = 35$.
- Pairwise comparison: we ask the model to compare the new statement with every other statement in separate API calls to obtain the insertion position.
- Iterative ranking: we rerun the algorithm for building preference profiles as in Section 7.1 and extract the resulting position of the new statement as its insertion position.

In Figure 8, we evaluate the performance of these insertion algorithms by randomly selecting a statement from a preference profile to be (re-)inserted. We did this for 10 randomly sampled statements and 100 voters for 1,000 insertion tests for each method. We found that the iterative

ranking provided the best fit while still being cost-efficient, and this is the method used in our main experiments.

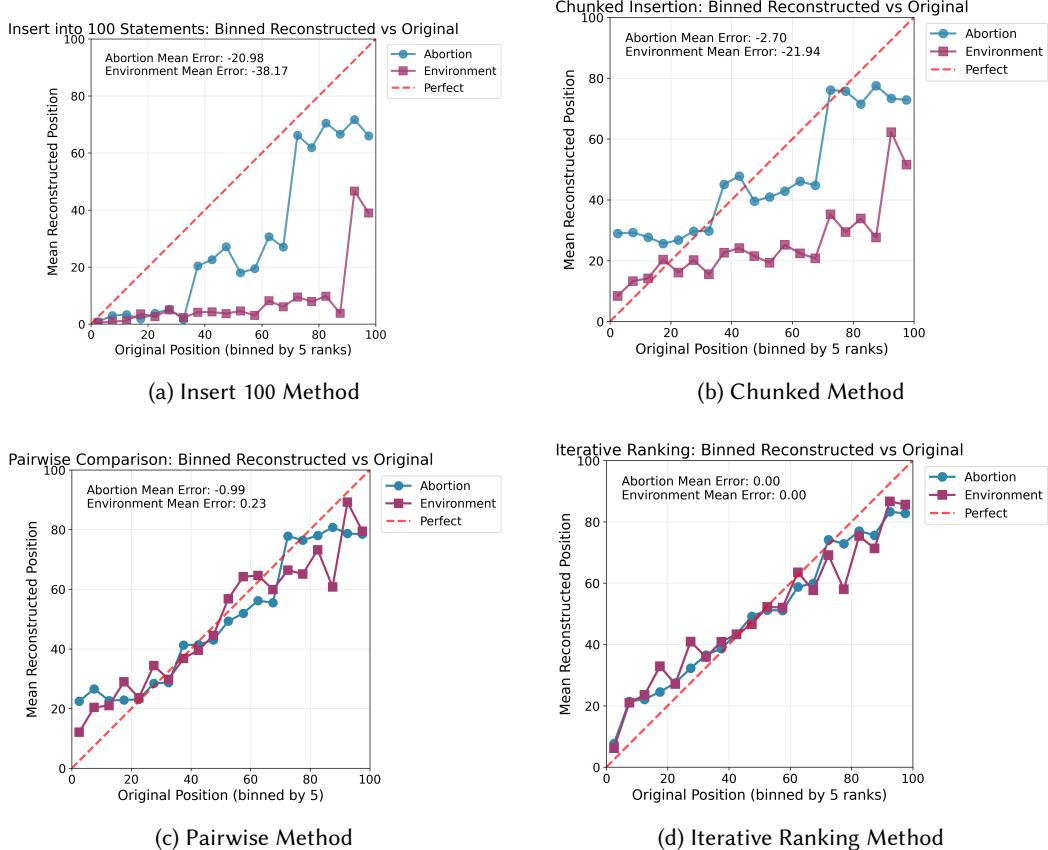


Fig. 8. Binned scatter plots comparing various statement insertion methods by plotting reconstruction position versus original position. Error units are in the ranks, e.g. an error of -20 means that the method ranks the new statement on average 20 positions below its original.

D.7 Detailed Results for Critical Epsilons of Voting Rules

In Tables 5 and 6 and Fig. 9, we present additional detailed statistics of the critical epsilons achieved by various traditional voting rules.

Table 5. P99 of critical epsilons for various voting rules across various topics

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	0.0161	0.0000	0.0061	0.0061	0.0000	0.0230
Borda	0.1432	0.0200	0.0100	0.0366	0.0230	0.0265
Schulze	0.1432	0.0361	0.0300	0.0522	0.0300	0.0265
IRV	0.2327	0.1244	0.0500	0.1427	0.0925	0.0265
Random	0.3101	0.1700	0.1200	0.1401	0.1801	0.0801
Plurality	0.3344	0.2088	0.1720	0.1583	0.1030	0.0990

Table 6. Percentage of critical epsilons that are less than 0.01 for various voting rules across various topics

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	95.0000	100.0000	97.5000	97.5000	100.0000	91.6667
Borda	87.5000	85.0000	95.0000	97.5000	91.6667	83.3333
Schulze	75.0000	77.5000	85.0000	87.5000	80.5556	77.7778
IRV	65.0000	42.5000	75.0000	75.0000	69.4444	66.6667
Random	58.1000	71.4000	72.4000	71.7000	72.6667	73.7778
Plurality	45.0000	22.5000	57.5000	62.5000	58.3333	58.3333

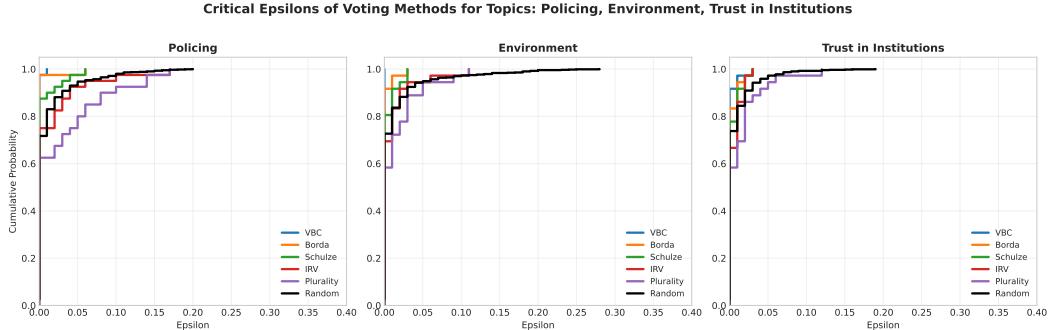


Fig. 9. CDF of critical epsilons for least controversial topics for various voting rules.

D.8 LLM-Powered Voting Rules as a Selector

In addition to testing LLM methods that generate their own statement, we also tested methods that use LLMs to select an existing statement. We tested six different variants that vary in how much information is given to the model.

- GPT-Select: the model is given the 20 sampled statements.
- GPT-Select+Rankings: the model is given the 20 sampled statements, and the 20×20 preference profile.
- GPT-Select+Personas: the model is given the 20 sampled statements and the personas of the 20 voters.
- GPT-Full: the model is given the 100 statements.
- GPT-Full+Rankings: the model is given the 100 statements, and the 20×100 preference profile.
- GPT-Full+Personas: the model is given the 100 statements and the personas of the 20 voters.

We present the results in Tables 7 to 9. VBC still outperforms the LLM methods. Interestingly, that providing more information, like voter personas or the rankings, slightly degrades performance.

Table 7. Mean critical epsilons for VBC, voting rules with GPT as a selector and a random baseline across various topics. Smallest epsilons are bolded.

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	0.0008	0.0000	0.0003	0.0003	0.0000	0.0014
GPT-Select	0.0038	0.0048	0.0048	0.0108	0.0103	0.0094
GPT-Sel+Rank	0.0143	0.0365	0.0163	0.0085	0.0075	0.0056
GPT-Sel+Pers	0.0125	0.0028	0.0095	0.0195	0.0100	0.0067
GPT-Full	0.0013	0.0050	0.0068	0.0055	0.0014	0.0053
GPT-Full+Rank	0.0593	0.0425	0.0055	0.0115	0.0114	0.0075
GPT-Full+Pers	0.0050	0.0023	0.0083	0.0143	0.0086	0.0039
Random	0.0609	0.0108	0.0082	0.0103	0.0108	0.0072

Table 8. P99 critical epsilons for VBC, voting rules with GPT as a selector and a random baseline across various topics. Smallest epsilon for each topic is bolded.

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	0.0161	0.0000	0.0061	0.0061	0.0000	0.0230
GPT-Select	0.0505	0.0422	0.0361	0.1166	0.1190	0.0400
GPT-Sel+Rank	0.1859	0.2088	0.2149	0.0722	0.0820	0.0265
GPT-Sel+Pers	0.1954	0.0422	0.0900	0.0900	0.0865	0.0365
GPT-Full	0.0100	0.0322	0.0666	0.0805	0.0100	0.0400
GPT-Full+Rank	0.3461	0.2205	0.0461	0.0922	0.0925	0.0465
GPT-Full+Pers	0.1032	0.0261	0.1254	0.1000	0.1700	0.0465
Random Insertion	0.3161	0.0861	0.0622	0.1449	0.0300	0.0600
Random	0.3101	0.1700	0.1200	0.1401	0.1801	0.0801

Table 9. Percentage of critical epsilons that are less than 0.01 for VBC, voting rules with GPT as a selector and a random baseline across various topics. Largest percentages for each topic is bolded.

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	0.950	1.000	0.975	0.975	1.000	0.917
GPT-Select	0.850	0.775	0.750	0.725	0.750	0.667
GPT-Sel+Rank	0.700	0.450	0.700	0.825	0.694	0.639
GPT-Sel+Pers	0.900	0.900	0.775	0.550	0.778	0.722
GPT-Full	0.875	0.700	0.675	0.875	0.861	0.778
GPT-Full+Rank	0.500	0.350	0.775	0.700	0.611	0.583
GPT-Full+Pers	0.900	0.875	0.750	0.650	0.889	0.889
Random Insertion	0.500	0.650	0.600	0.725	0.550	0.600
Random	0.581	0.714	0.724	0.717	0.727	0.738

D.9 Details on Clustered Voters

D.9.1 Keywords Used for Classification. To cluster voters, we applied case-insensitive keyword matching on the ideology field in the personas using the following sets of keywords.

Conservative/Traditional.

- conservative
- traditional
- traditionalist
- libertarian
- libertarianism
- fiscal conservatism
- social conservatism
- christian values
- family values
- small government

Progressive/Liberal.

- progressive
- liberal
- social liberal
- social justice
- feminist
- feminism
- egalitarian
- environmentalism
- environmentalist
- environmental conservation
- environmental protection
- social equality
- workers' rights
- social welfare
- humanism
- humanistic
- humanitarian

D.9.2 More Results on Conservative Voters. We report additional statistics in Tables 10 and 11.

Table 10. P99 critical epsilons for for VBC, generative voting rules and random baselines across various topics for conservative voters. Smallest epsilon for each topic is bolded.

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	0.0161	0.0061	0.0061	0.0000	0.0061	0.0000
GPT-Blind	0.5005	0.0900	0.0461	0.1022	0.3588	0.1161
GPT-Synthesize	0.3300	0.1832	0.0944	0.2298	0.1166	0.3461
GPT-Synth+Rank.	0.3544	0.1261	0.1144	0.1022	0.2722	0.0300
GPT-Synth.+Pers.	0.2700	0.0461	0.0322	0.0200	0.0361	0.0100
Random Insertion	0.3261	0.4805	0.3683	0.3600	0.1000	0.2286
Random	0.4601	0.5100	0.4200	0.4201	0.3301	0.3901

D.9.3 Results on Progressive Voters. We present the critical epsilons of generative voting rules on progressive voters in Tables 12 to 14.

Table 11. Percentage of critical epsilons that are less than 0.01 for VBC, generative voting rules and random baselines across various topics for conservative voters. Largest percentages for each topic is bolded.

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	0.900	0.975	0.975	1.000	0.975	1.000
GPT-Blind	0.000	0.225	0.750	0.425	0.100	0.275
GPT-Synthesize	0.000	0.350	0.325	0.175	0.175	0.325
GPT-Synth+Rank.	0.025	0.400	0.575	0.425	0.500	0.675
GPT-Synth.+Pers.	0.250	0.550	0.725	0.750	0.575	0.900
Random Insertion	0.425	0.100	0.575	0.400	0.525	0.750
Random	0.338	0.242	0.609	0.389	0.484	0.539

Table 12. Mean critical epsilons for for VBC, generative voting rules and random baselines across various topics for progressive voters. Smallest epsilon for each topic is bolded.

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	0.0000	0.0018	0.0000	0.0005	0.0000	0.0003
GPT-Blind	0.0185	0.0280	0.0113	0.0048	0.0035	0.0215
GPT-Synthesize	0.0095	0.0103	0.0058	0.0135	0.0108	0.0200
GPT-Synth+Rank.	0.0055	0.0075	0.0005	0.0025	0.0025	0.0093
GPT-Synth.+Pers.	0.0058	0.0050	0.0013	0.0033	0.0018	0.0143
Random Insertion	0.2205	0.0130	0.0220	0.0532	0.0388	0.0120
Random	0.1751	0.0989	0.0428	0.0625	0.0382	0.0330

Table 13. P99 critical epsilons for for VBC, generative voting rules and random baselines across various topics for progressive voters. Smallest epsilon for each topic is bolded.

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	0.0000	0.0405	0.0000	0.0122	0.0000	0.0069
GPT-Blind	0.1622	0.1805	0.0822	0.0261	0.0283	0.0922
GPT-Synthesize	0.0844	0.0600	0.0300	0.1710	0.0927	0.1161
GPT-Synth+Rank.	0.0200	0.0461	0.0100	0.0322	0.0161	0.0661
GPT-Synth.+Pers.	0.0200	0.0300	0.0161	0.0300	0.0100	0.1083
Random Insertion	0.6444	0.1781	0.2544	0.3966	0.3149	0.0761
Random	0.4601	0.5100	0.4200	0.4201	0.3301	0.3901

Table 14. Percentage of critical epsilons that are less than 0.01 for VBC, generative voting rules and random baselines across various topics for progressive voters. Largest percentages for each topic is bolded.

Method	Abortion	Electoral College	Healthcare	Policing	Environment	Trust in Institutions
VBC	1.000	0.950	1.000	0.975	1.000	0.969
GPT-Blind	0.500	0.350	0.600	0.650	0.725	0.350
GPT-Synthesize	0.600	0.500	0.575	0.625	0.525	0.350
GPT-Synth+Rank.	0.650	0.600	0.950	0.850	0.775	0.650
GPT-Synth.+Pers.	0.650	0.650	0.900	0.800	0.825	0.500
Random Insertion	0.275	0.600	0.675	0.625	0.550	0.600
Random	0.338	0.242	0.609	0.389	0.484	0.539

D.10 Prompts

Here we present the prompt templates used to generate statements, build preference profiles, and insert statements.

D.10.1 Generating Alternatives.

With persona, without deliberation round. These are the prompts we used for the main experiments.

System Prompt

You are writing a statement that reflects your perspective on a topic.

User Prompt

You are a person with the following characteristics:

{persona}

Topic: "{topic}"

Write a bridging statement expressing your views on this topic. Your statement should:

- Reflect your background, values, and life experiences
- Aim to find common ground or bridge different viewpoints
- Be 2-4 sentences long
- NOT write in first-person
- NOT explicitly reference your identity or demographics (avoid "As a [X]...")

Write only the statement:

With persona, with deliberation round.

System Prompt

You are writing a statement that reflects your perspective on a topic.

User Prompt

You are a person with the following characteristics:

{persona}

Topic: "{topic}"

Here are 100 statements from people with diverse perspectives on this topic:

{statements_list}

Write a NEW bridging statement expressing your views on this topic. Your statement should:

- Reflect your background, values, and life experiences
- Synthesize key themes you observed across the discussion
- Aim to find common ground or bridge different viewpoints
- Be 2-4 sentences long
- NOT write in first-person
- NOT explicitly reference your identity or demographics (avoid "As a [X]...")
- Be self-contained (do not reference "the statements above" or "other people")

Write only the statement:

Without persona, with deliberation round.

System Prompt

You are a helpful assistant that generates statements. Return only the statement text, no JSON or additional commentary. For each query, please generate a set of five possible responses, each within a separate `<response>` tag. Each `<response>` must include a `<text>` and a numeric `<probability>`.

Please sample at random from the tails of the distribution, such that the probability of each response is less than 0.10.

User Prompt

Topic: "{topic}"

Here are 100 statements from people with diverse perspectives on this topic:

{statements_list}

Write a NEW bridging statement on this topic. Your statement should:

- Synthesize key themes across the different viewpoints
- Aim to find common ground or bridge different viewpoints
- Be 2-4 sentences long
- Be self-contained (do not reference "the statements above" or "other people")

Without persona, without deliberation round.

System Prompt

You are a helpful assistant that generates statements. Return only the statement text, no JSON or additional commentary. For each query, please generate a set of five possible responses, each within a separate `<response>` tag. Each `<response>` must include a `<text>` and a numeric `<probability>`.
 Please sample at random from the tails of the distribution, such that the probability of each response is less than 0.10.

User Prompt

Topic: "{topic}"

Write a bridging statement on this topic. Your statement should:

- Aim to find common ground or bridge different viewpoints
- Be 2-4 sentences long

D.10.2 Building Preference Profiles. Using iterative ranking (Appendix D.4.2), the first few rounds ask the model to pick the top and bottom K statements, and the final round asks the model to rank the remaining statements.

Initial Rounds.

User Prompt

Topic: "{topic}"

Here are $\{n\}$ statements (identified by 4-letter codes):

{stmt_lines}

From these $\{n\}$ statements, identify:

1. Your TOP $\{k\}$ most preferred (in order, most preferred first)
2. Your BOTTOM $\{k\}$ least preferred (in order, least preferred last)

IMPORTANT: Do NOT simply list codes in the order they appear above.
 Your preferences should reflect your persona's values and background.

Return JSON: `{"top_{k}": ["code1", "code2", ...], "bottom_{k}": ["code1", "code2", ...]}`

Where `stmt_lines` is formatted as:

User Prompt

XXXX: "Statement text here..."

YYYY: "Another statement..."

*Final Round.***User Prompt**

Topic: "{topic}"

Here are {n} statements (identified by 4-letter codes):
 {stmt_lines}

Rank ALL of these statements from most to least preferred.

IMPORTANT: Do NOT simply list codes in the order they appear above.
 Your preferences should reflect your persona's values and background.

Return JSON: {"ranking": ["most_preferred", "second", ..., "least_preferred"]}

*D.10.3 Winner Selection.**GPT-Select: Select from P Alternatives.***System Prompt**

You are a helpful assistant that selects consensus statements. Return ONLY valid JSON.

User Prompt

Here are {n} statements from a discussion:

{statements_text}

Which statement would be the best choice as a consensus/bridging statement?
 Consider which one best represents a reasonable middle ground that could satisfy diverse perspectives.

Return your choice as JSON: {"selected_statement_index": <index>}
 Where the value is the index (0-{n-1}) of the statement you select.

Where statements_text is formatted as:

Format

Statement 0: Statement text here...

Statement 1: Another statement...

GPT-Select+Rank: Select with Rankings.

System Prompt

You are a helpful assistant that selects consensus statements. Return ONLY valid JSON.

User Prompt

Here are {n} statements from a discussion:

{statements_text}

Here are preference rankings from {n_voters} voters:

{rankings_text}

Based on both the statements and the preference rankings, which statement would be the best choice as a consensus/bridging statement?

Return your choice as JSON: {"selected_statement_index": <index>}

Where the value is the index (0-{n-1}) of the statement you select.

Where `rankings_text` is formatted as ($K = 20$ sampled voters, full rankings over $P = 20$ alternatives):

Format

Voter 1: 5 > 3 > 1 > 8 > 2 > 0 > 7 > 6 > 4 > 9 > 10 > 11 > 12 > 13 > 14 > 15 > 16 > 17 > 18 > 19

Voter 2: 3 > 5 > 8 > 1 > 2 > 0 > 7 > 6 > 4 > 9 > 10 > 11 > 12 > 13 > 14 > 15 > 16 > 17 > 18 > 19

...

Voter 20: 8 > 3 > 5 > 1 > 2 > 0 > 7 > 6 > 4 > 9 > 10 > 11 > 12 > 13 > 14 > 15 > 16 > 17 > 18 > 19

GPT-Select+Pers: Select with Personas.

System Prompt

You are a helpful assistant that selects consensus statements. Return ONLY valid JSON.

User Prompt

Here are {n} statements from a discussion:

```
{statements_text}
```

Here are the {n_voters} voters who will be voting on these statements:

```
{personas_text}
```

Based on both the statements and the voter personas, which statement would be the best choice as a consensus/bridging statement?

Return your choice as JSON: {"selected_statement_index": <index>}

Where the value is the index (0-{n-1}) of the statement you select.

Where personas_text is formatted as ($K = 20$ sampled voters, filtered to 7 key fields to save token costs):

Format

```
Voter 1: age: 53
sex: Male
race: White alone
education: Master's degree
occupation: MGR-Education And Childcare Administrators
political views: Liberal
religion: Protestant
```

```
Voter 2: age: 54
sex: Male
race: White alone
education: Master's degree
occupation: ENT-News Analysts, Reporters, And Journalists
political views: Democrat
religion: Catholic
```

...

Voter 20: ...

GPT-Full: Select from All 100.

System Prompt

You are a helpful assistant that selects consensus statements. Return ONLY valid JSON.

User Prompt

Topic: {topic}

A group of participants submitted the following {n_all} statements on this topic:

{all_text}

Select the statement that would best serve as a consensus or bridging position - one that:

- Engages substantively with the topic
- Could be acceptable to participants with diverse viewpoints
- Avoids extreme or polarizing framing

Return your choice as JSON: {"selected_statement_index": <index>}

Where the value is the index (0-{n_all-1}) of the statement you select.

The GPT-Full+Rank and GPT-Full+Pers variants add `rankings_text` and `personas_text` respectively, using the same format as GPT-Select.

GPT-Synthesize: Generate New Statement.

System Prompt

You are a helpful assistant that generates consensus statements. Return ONLY valid JSON.

User Prompt

Topic: {topic}

Here are some existing statements from a discussion:

{statements_text}

Generate a NEW statement that could serve as a better consensus/bridging statement.

The statement should:

- Represent a reasonable middle ground that could satisfy diverse perspectives
- Be different from the existing statements but address the same topic
- Be clear and substantive (2-4 sentences)

Return your new statement as JSON: {"new_statement": "<your statement>"}

The GPT-Synthesize+Rank and GPT-Synthesize+Pers variants add `rankings_text` and `personas_text` respectively as in GPT-Select.

GPT-Blind: Blind Generation.
System Prompt

You are a helpful assistant that generates bridging statements. Return ONLY valid JSON.

User Prompt

Given the topic: "{topic}"

Generate a bridging statement that could serve as a consensus position on this topic.

The statement should:

- Represent a reasonable middle ground that could satisfy diverse perspectives
- Acknowledge different viewpoints while finding common ground
- Be clear and substantive (2-4 sentences)

Return your statement as JSON: {"bridging_statement": "<your statement>"}