Learning Lotteries with Minimal Stakeholder Violations from Membership Queries

Nicholas Teh
University of Oxford
nicholas.teh@cs.ox.ac.uk

Paul W. Goldberg
University of Oxford
paul.goldberg@cs.ox.ac.uk

Abstract

An AI developer often has several candidate versions of a system and can deploy a mixture of them, sending each request to a randomly chosen version, as in A/B testing. Each stakeholder (e.g., a safety team, a regulator, or a customer with private test data) accepts a mixture exactly when its expected performance (e.g., its rate of harmful replies) meets the stakeholder's own standard, which the developer can learn only by asking whether a proposed mixture is acceptable. We model this as choosing a lottery over m alternatives for n agents, each of whom accepts exactly the lotteries in an unknown halfspace and answers such binary membership queries. While Choo et al. (ICML 2026) sought a lottery that every agent accepts, none may exist when, say, safety and helpfulness requirements conflict, so we minimize the number of rejecting agents. We show that this problem is APX-hard even when all halfspaces are known and all utilities are 0 or 1. We then give exact algorithms that run in polynomial time when m is constant, and in quasi-polynomial time when, for a constant η, some optimal lottery has no accepting agent within η of its threshold. When only some agents can be queried, optimizing over a uniform sample of agents gives additive and relative guarantees for the whole population, with a sample size that does not depend on n, and a linear program gives a polynomial-time method for any m. On the other hand, for $m \geq 3$, a deterministic algorithm that queries only K distinct agents cannot guarantee fewer than $n - K$ extra rejections, and this is achievable; in other words, it can guarantee nothing for the agents it does not ask. Small synthetic experiments indicate that our worst-case bounds are conservative on random instances.

1 Introduction

Training a language model to refuse harmful requests also makes it refuse some harmless ones [Bai et al., 2022, Röttger et al., 2024], and the parties affected by a deployment weigh these two errors differently. Consider a chatbot with a *strict* version, which gives a harmful reply to 0.2% of risky prompts but refuses 12% of harmless requests, and a *lenient* version, with rates of 1% and 2%. The developer need not commit to one version: it can route each conversation to a randomly chosen version, which is routine in A/B testing and staged releases [Kohavi et al., 2020]. If two thirds of conversations go to the strict version, linearity of expectation gives rates of about 0.47% and 8.7%. This mixture meets both a safety team's limit of 0.5% on harmful replies and a hospital's limit of 9% on refusals, although neither version meets both limits on its own. Suppose that a regulator also caps harmful replies at 0.3%. Then at least 87.5% of conversations must go to the strict version, whereas the hospital accepts at most 70%; no mixture meets all three requirements, and the developer has to choose whose requirement goes unmet.

We model such decisions with a standard framework from probabilistic social choice and random assignment [Brandt, 2017, Hylland and Zeckhauser, 1979]. A decision maker chooses a *lottery* over a finite set of alternatives (here, model versions), and each agent (here, a stakeholder) accepts it if

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

and only if the expected score of the lottery is at least the agent's threshold; the lotteries that an agent accepts therefore form a halfspace of the probability simplex. For requirements on average rates—of harmful replies, of refusals, or of errors on a customer's own test set—this linear model is exact, because a mixture's rate equals the sum, over versions, of the routing probability times that version's rate. The same model also describes applications ranging from content moderation [Gillespie, 2018, Gorwa et al., 2020] to school or housing allocation [Abdulkadiroğlu and Sönmez, 2003, Budish, 2011] to the division of a budget among public projects [Suksompong and Teh, 2026].

The decision maker, however, knows neither the scores nor the thresholds. A customer will not reveal its private test data, and a regulator's standard may be available only as legal text [European Union, 2024]. What a stakeholder can do is look at a concrete proposal and answer yes or no; following Angluin [1988], we call such a question ("is this lottery acceptable to you?") a *membership query*. Answering it demands far less of a stakeholder than reporting full utilities [Boutilier et al., 2006, Braziunas and Boutilier, 2008], yet a single answer can cost an evaluation run or a human review, so a decision maker facing thousands of stakeholders can consult only some of them.

This query model was introduced by Choo et al. [2026b], who sought *unanimity*: a lottery that every agent accepts, or a report that there is none. Unanimity fits settings in which every requirement is mandatory, but it offers little guidance when requirements conflict. As the example shows, a single demanding agent can rule out every lottery, and learning that no lottery is unanimously acceptable does not tell the developer what to deploy (keeping the old version is one of the alternatives, not a way out of the choice). This paper addresses the following question: *when the requirements cannot all be met, how can a decision maker who asks few membership queries find a lottery that as few agents as possible reject?* We call the problem of minimizing the number of rejecting agents MIN-VIOL.

One could instead measure how severely each agent is violated, as Teh and Goldberg [2026] do in the same query model by minimizing a power mean of the distances from the lottery to the violated agents' acceptance boundaries (Appendix A compares the two objectives). We count rejecting agents because, for safety, a small aggregate violation can hide many agents who each fall just below their thresholds. This choice has a computational cost. Unanimity is a linear feasibility problem, whereas MIN-VIOL asks for a point of the simplex satisfying as many linear inequalities as possible, which is hard in general (Section 1.2).

1.1 Our Results

Our main results concern what can be promised to the stakeholders who are not asked.

Even with full information, the problem is hard. MIN-VIOL is APX-hard even when all utilities are 0 or 1 (Theorem 3.1), so it has no polynomial-time approximation scheme unless $\mathrm{P} = \mathrm{NP}$; this strengthens the NP-hardness shown by Choo et al. [2026b]. In other words, even a developer who knows every requirement cannot, in general, get arbitrarily close to the best lottery efficiently.

Exact algorithms under additional structure. When the number m of alternatives is constant (say, a handful of model versions), MIN-VIOL can be solved exactly in polynomial time (Theorem 4.1), by an algorithm that is XP rather than FPT in m. For larger m, if some optimal lottery leaves no accepting agent within a margin η of its threshold, then some optimal lottery uses only $\mathcal{O}(\log n/\eta^2)$ alternatives (Theorem 4.2, via a sampling argument of Castiglioni et al. [2023]), and searching over such lotteries finds one in quasi-polynomial time for constant η (Corollary 4.4). Finally, when few violations are allowed, Helly's theorem gives an algorithm that is FPT in the numbers of alternatives and violations together, whereas the number of violations alone gives W[2]-hardness (Section 4.3).

Learning from a random sample of agents. If $k \geq m$ agents are sampled uniformly at random and their halfspaces are recovered, then, with probability at least $1 - \delta$, any lottery that minimizes the number of rejecting sampled agents has at most $\mathrm{OPT} + \mathcal{O}(n\sqrt{(m\log(ek/m) + \log(1/\delta))/k})$ violations, where OPT is the minimum number of violations (Theorem 5.1). Put another way, for an error that is a given fraction of n, the number of agents to ask does not depend on n: a developer with a million customers need not ask more of them than one with a thousand. If $\mathrm{OPT} \geq \rho n$, then $\mathcal{O}((m\log(1/\rho) + \log(1/\delta))/(\zeta^2\rho))$ samples give a $(1+\zeta)$-approximation (Theorem 5.2).

A polynomial-time linear program for any number of alternatives. For large m, we solve a linear program that penalizes agents who reject the lottery or accept it with margin less than η. With high probability, for every lottery $\mathbf{x}$, its output has at most $(1 + 1/\eta)$ times as many violations as $\mathbf{x}$, plus

the number of agents who accept $\mathbf{x}$ with margin less than η, plus $\mathcal{O}\big(\frac{n}{\eta}\sqrt{(\log m + \log(1/\delta))/k}\big)$ (Theorem 5.3). The guarantee is strong when few agents are barely satisfied.

Without random sampling, unasked agents get no guarantee. Can clever, adaptive questioning of a fixed set of agents replace random sampling? For $m \geq 3$, it cannot: every deterministic algorithm that queries only K distinct agents may leave $n - K$ more agents unsatisfied than the optimum, however many queries it spends on them (Theorem 6.1), and optimizing over a fixed set of K queried agents matches this bound (Theorem 6.2). In other words, a deterministic decision maker can promise nothing to the agents it does not ask. For AI deployment, this means that a developer who always consults the same internal teams or large customers has, in the worst case, no guarantee for anyone else; choosing at random whom to ask is what makes the sampling guarantees possible.

Synthetic experiments. On random instances, our bounds are conservative: with 500 agents, the lottery learned from a random sample of 120 had, on average, fewer than 10 more rejecting agents than the optimum, and the $n - K$ worst case arose only for our lower-bound construction (Section 7).

1.2 Related Work

We discuss the closest work here; Appendix A covers further areas (see Table 2).

Learning lotteries from membership queries. Choo et al. [2026b] bounded the number of membership queries needed to find a lottery that every agent accepts, or to conclude that there is none, and showed that maximizing the number of accepting agents is NP-hard even with binary utilities [Choo et al., 2026b, Theorem B.3 of the extended version]. We use their routine for recovering one agent's halfspace as a black box, but our objective is different.

Maximum feasible subsystem. With full information, MIN-VIOL asks for a point of the simplex satisfying as many of the agents' linear inequalities as possible. It is therefore a restricted case of the *maximum feasible subsystem* (MAX-FS) problem, or of its complement, the *minimum unsatisfied linear relation* (MIN-ULR) problem. For general systems of linear inequalities without mandatory constraints, MAX-FS can be approximated within a factor 2 but has no PTAS unless $\mathrm{P} = \mathrm{NP}$ [Amaldi and Kann, 1995], while MIN-ULR is much harder: it cannot be approximated within $2^{\log^{1-c} n}$ for any constant $c > 0$ unless $\mathrm{NP} \subseteq \mathrm{QP}$ [Arora et al., 1997, Amaldi and Kann, 1998]; see also Amaldi et al. [2003] and, for $0/1$ coefficients, Elbassioni et al. [2009]. Hardness for general systems does not by itself give hardness for lottery instances, in which the variables form a probability vector, the simplex equation must hold, and, with binary utilities, each agent requires enough probability on a fixed subset of alternatives. Theorem 3.1 gives a reduction that stays within these instances. Conversely, a full-information method that treats the simplex constraints as mandatory can be used once all n agents have been recovered, at a cost of n times the per-agent recovery cost; our limited-query results ask what can be guaranteed with fewer queries. Castiglioni et al. [2023] study a relaxation of MAX-FS over the simplex in which each constraint may be violated by a small additive slack. Theorem 4.2 uses their sampling argument, and Section 4.2 compares the two.

AI alignment with diverse stakeholders. Conitzer et al. [2024] and Sorensen et al. [2024] argued that aligning AI systems with people whose values differ calls for tools from social choice; this line of work mostly concerns training, whereas we study which mixture of trained versions to deploy. In a similar vein, Neoh et al. [2026b] selected a small committee of AI systems from binary or pairwise feedback using few queries. Randomized classifiers in fair classification meet known constraints on group-level rates [Agarwal et al., 2018]; our constraints are hidden and need not all be met.

Probabilistic social choice and portioning. Randomization is widely used to aggregate preferences, break ties, allocate resources, or satisfy fairness and efficiency criteria [Brandt, 2017, Hylland and Zeckhauser, 1979, Bogomolnaia and Moulin, 2001, Abdulkadiroğlu and Sönmez, 2003, Budish, 2011]. A lottery can also be viewed as a division of a divisible resource, such as money or time, among the alternatives; aggregating preferences into such a division is known as *portioning* or budget aggregation [Bogomolnaia et al., 2005, Aziz et al., 2019, Freeman et al., 2021, Airiau et al., 2023, Freeman and Schmidt-Kraepelin, 2024, Elkind et al., 2026], and we refer to the survey by Suksompong and Teh [2026] for an overview of voting in such divisible settings. This literature takes preferences, proposed divisions, priorities, or constraints as input, whereas our decision maker observes only accept/reject answers about proposed lotteries.

Learning-augmented algorithms. Learning-augmented algorithms use possibly erroneous predictions to circumvent worst-case limits such as Theorem 6.1 [Mitzenmacher and Vassilvitskii, 2021], for instance in mechanism design [Agrawal et al., 2024], online fair division [Banerjee et al., 2023, Neoh et al., 2026a], and voting [Berger et al., 2024] (Appendix A); Choo et al. [2026b] used such advice to find a unanimously acceptable lottery with fewer queries.

2 Preliminaries

Let $m \geq 2$ be the number of alternatives and $n \geq 1$ the number of agents. For an integer $z \geq 1$, write $[z] := \{1, \dots, z\}$. Let $N := [n]$ be the set of agents and $S := \{s_1, \dots, s_m\}$ the set of alternatives. A *lottery* is a probability vector $\mathbf{x} = (x_1, \dots, x_m) \in \Delta(S) := \{\mathbf{x} \in \mathbb{R}^m_{\geq 0} : \sum_{j=1}^m x_j = 1\}$, where x_j is the probability of s_j; the *pure* lottery on s_j is the standard basis vector $\mathbf{e}_j$. We write $\langle \cdot, \cdot \rangle$ for the standard inner product on $\mathbb{R}^m$, and $\log$ for the natural logarithm.

Agents. Agent $i \in N$ accepts exactly the lotteries in its *acceptable set* $A_i = \{\mathbf{x} \in \Delta(S) : \langle u_i, \mathbf{x} \rangle \geq \tau_i\}$, where $u_i \in [0,1]^m$ gives agent i's utility $u_i(s_j)$ for each alternative and τ_i is its threshold. We write $U_i(\mathbf{x}) := \langle u_i, \mathbf{x} \rangle$ for the expected utility. Thus A_i is the part of the simplex on one side of a hyperplane (Figure 1a). We exclude degenerate agents: every agent accepts at least one lottery and rejects at least one, i.e., $\varnothing \neq A_i \subsetneq \Delta(S)$. Membership queries reveal A_i but not the pair (u_i, τ_i): since $\langle \mathbf{1}, \mathbf{x} \rangle = 1$ on $\Delta(S)$, the pair $(\lambda u_i + \mu \mathbf{1}, \lambda \tau_i + \mu)$ gives the same set A_i for every $\lambda > 0$ and $\mu \in \mathbb{R}$. Margins $U_i(\mathbf{x}) - \tau_i$ are therefore meaningful only after fixing a normalization. If A_i has nonempty relative interior in $\Delta(S)$, we use the unique pair with $\min_j u_i(s_j) = 0$ and $\max_j u_i(s_j) = 1$. Otherwise, A_i consists of the lotteries that put all probability on some nonempty proper subset of alternatives, and we let u_i be the indicator vector of that subset and $\tau_i = 1$ (Lemma B.1). From now on, (u_i, τ_i) denotes this *normalized pair*; thus $u_i \in [0,1]^m$ and $\tau_i \in (0,1]$.

Membership queries. A query $(i, \mathbf{x}) \in N \times \Delta(S)$ returns $\texttt{Query}(i, \mathbf{x}) = 1$ if agent i accepts $\mathbf{x}$ and 0 otherwise; no utility or threshold is observed. We write Q for the total number of queries and K for the number of distinct agents queried.

Finite-grid convention. All geometric and statistical results allow arbitrary real-valued normalized pairs. Only when we state an explicit query budget, or a running time that depends on recovered agents, do we assume that the normalized pair lies on a known grid: $u_i(s_j) \in \{0, \varepsilon, 2\varepsilon, \dots, 1\}$ and $\tau_i \in \{\varepsilon, 2\varepsilon, \dots, 1\}$, where $\varepsilon \in (0, 1/2]$ and $1/\varepsilon$ is an integer. The lottery is not discretized: the decision maker may propose any point of $\Delta(S)$. Under this convention, the halfspace recovery routine of Choo et al. [2026b] finds an agent's normalized pair with $T_\varepsilon = \mathcal{O}(m \log(1/\varepsilon))$ queries; we call this *recovering* the agent. Appendix B.2 explains why some such assumption is needed for exact recovery.

Objective. The *violation count* of a lottery $\mathbf{x}$ is $\mathrm{viol}(\mathbf{x}) := |\{i \in N : U_i(\mathbf{x}) < \tau_i\}|$, the number of agents who reject $\mathbf{x}$. The problem MIN-VIOL asks for a lottery minimizing viol, and $\mathrm{OPT} := \min_{\mathbf{x} \in \Delta(S)} \mathrm{viol}(\mathbf{x})$. The minimum is attained: the set of lotteries with at most a given number of violations is a finite union of intersections of the closed sets A_i, hence closed in the compact set $\Delta(S)$. The same holds for any finite set or multiset of agents. The following example shows that mixed lotteries can be strictly better than pure ones.

Example 2.1. Let $m \geq 4$ and $n = 4$, and let agents 1, 2, 3, and 4 accept $\mathbf{x}$ if and only if $x_1 + x_2 \geq 3/5$, $x_1 + x_3 \geq 3/5$, $x_2 + x_3 \geq 3/5$, and $x_4 \geq 3/10$, respectively; all utilities are 0 or 1. If agents 1, 2, and 3 accept $\mathbf{x}$, adding their conditions gives $x_1 + x_2 + x_3 \geq 9/10$, so $x_4 \leq 1/10$ and agent 4 rejects $\mathbf{x}$. Hence no lottery is accepted by every agent. The lottery $\mathbf{x}^* = (1/3, 1/3, 1/3, 0, \dots, 0)$ is rejected only by agent 4, so $\mathrm{OPT} = 1$, whereas every pure lottery is rejected by at least two agents. The three agents who accept $\mathbf{x}^*$ do so with margin $2/3 - 3/5 = 1/15$.

Interpreting linear acceptability. $U_i(\mathbf{x})$ is the expected value, under the lottery, of a quantity attached to each alternative: an expected von Neumann–Morgenstern utility [von Neumann and Morgenstern, 1944, Fishburn, 1982], or an average outcome such as access probability, service level, safety, or the negative of a loss. By the law of total expectation, such averages combine linearly when an alternative is drawn at random. This linear threshold model is the standard cardinal baseline in random assignment and probabilistic social choice [Hylland and Zeckhauser, 1979, Brandt, 2017], and, as with linear classifiers in statistical learning [Vapnik, 1998, Anthony and Bartlett, 1999]

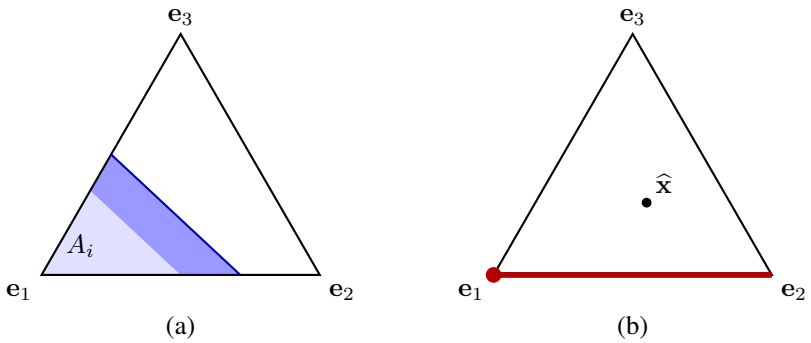


Figure 1: Lotteries over three alternatives drawn as points of a triangle with corners $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$. (a) An agent's acceptable set A_i (shaded) is the part of the triangle on one side of a line; the darker strip contains the lotteries that the agent accepts with margin less than η. (b) The profile in the proof of Theorem 6.1 when $\widehat{\mathbf{x}} \neq \mathbf{e}_1$: every queried agent accepts exactly the lotteries on the bottom edge, and every unqueried agent accepts only $\mathbf{e}_1$. All unqueried agents reject $\widehat{\mathbf{x}}$, but every agent accepts $\mathbf{e}_1$.

[Shalev-Shwartz and Ben-David, 2014], it gives the hidden acceptable sets a finite-dimensional structure that can be learned. Appendix B.1 and Section 8 discuss its limits.

3 Hardness of Approximating MIN-VIOL

Before asking what can be learned from limited queries, we study the full-information problem, in which all normalized pairs (u_i, τ_i) are given.

Theorem 3.1. *Given all normalized pairs $(u_i, \tau_i)_{i \in N}$,* MIN-VIOL *is APX-hard. In particular, unless* $\mathrm{P} = \mathrm{NP}$*, it admits no polynomial-time approximation scheme (PTAS). This remains true even when all utilities are binary, i.e., $u_i(s_j) \in \{0, 1\}$ for each $i \in N$ and $j \in [m]$.*

Proof idea. We give an approximation-preserving reduction from MINIMUM VERTEX COVER on cubic graphs, which is APX-hard [Alimonti and Kann, 2000]. For a cubic graph on q vertices, there is one alternative for each vertex and one extra alternative that receives the probability not placed on vertices. The *vertex agent* of vertex j has utility 1 only on alternative j and threshold $1/q$, so it accepts if and only if $x_j \geq 1/q$. For each edge $\{j, j'\}$, there are q^2 identical *edge agents* with utility 0 on alternatives j and j', utility 1 elsewhere, and threshold $1 - 1/q$, so they accept if and only if $x_j + x_{j'} \leq 1/q$. A lottery that violates an edge agent violates all q^2 copies, more than the q violations of the pure lottery on the extra alternative. From any lottery $\mathbf{x}$ we obtain a vertex cover of size at most $\mathrm{viol}(\mathbf{x})$: all vertices if some edge agent rejects $\mathbf{x}$, and otherwise the vertices j with $x_j < 1/q$, which cover every edge because no edge has both endpoints with probability at least $1/q$ (that would give $x_j + x_{j'} \geq 2/q$). Conversely, a vertex cover C gives a lottery with exactly $|C|$ violations (probability $1/q$ on each vertex outside C). Hence an α-approximation for MIN-VIOL gives an α-approximation for MINIMUM VERTEX COVER on cubic graphs. The full proof is in Appendix C. □

Comparison with hardness proofs for general systems. The hardness results of Amaldi and Kann [1995] concern linear systems with arbitrary coefficients and unconstrained variables. In our reduction, every graph constraint must instead be an acceptance condition with $0/1$ utilities over lotteries: the edge condition $x_j + x_{j'} \leq 1/q$ is written as $1 - x_j - x_{j'} \geq 1 - 1/q$, the probability of all alternatives other than j and j'. The extra alternative absorbs the probability not placed on vertices, so that every vertex cover gives a valid lottery and the pure lottery on the extra alternative violates only the q vertex agents; this is what makes violating an edge agent never worthwhile.

What the theorem does not show. Theorem 3.1 rules out a PTAS but not every constant-factor approximation. Much stronger inapproximability is known for MIN-ULR on general linear systems [Arora et al., 1997, Amaldi and Kann, 1998]; we do not know whether it holds for lottery instances. The theorem also concerns multiplicative approximation with full information, so it is not a lower bound matching the additive sampling guarantees of Section 5.

4 Exact Algorithms Under Full Information

Theorem 3.1 shows that exact optimization needs additional structure. We consider three kinds: a constant number of alternatives (Section 4.1), a margin condition under which few alternatives suffice (Section 4.2), and a small number of violations (Section 4.3).

4.1 A Constant Number of Alternatives

Fix the set of agents that a lottery must satisfy. The lotteries satisfying them form a polytope, and an optimal lottery can be taken to be a vertex of it. Each vertex is the unique solution of the simplex equation together with $m-1$ tight constraints, each of the form $x_j = 0$ or $U_i(\mathbf{x}) = \tau_i$. Enumerating all such linear systems and keeping the best solution gives the following result (proof in Appendix D.1). It is closely related to exact algorithms for linear programming with a given number of violated constraints in fixed dimension [Matoušek, 1995, Chan, 2005], which can also be applied here; we include it because it is simple and has an explicit running time.

Theorem 4.1. *Given all normalized pairs $(u_i, \tau_i)_{i \in N}$, there is a deterministic algorithm that returns a lottery minimizing* viol *in time $\mathcal{O}(\binom{n+m}{m-1}(m^3 + nm))$. This is polynomial for every constant number m of alternatives. Since the exponent of n grows with m, this is an XP algorithm in the parameter m, not an FPT one; in particular, the decision version (is* OPT *at most a given number?) is in* XP *when parameterized by m.*

4.2 Few Alternatives Under a Margin Condition

A lottery *uses* alternative s_j if $x_j > 0$. We now show that a lottery using many alternatives can be replaced by one using few. For $\eta > 0$ and a lottery $\mathbf{x}$, let $B_\eta(\mathbf{x}) := \{i \in N : \tau_i \le U_i(\mathbf{x}) < \tau_i + \eta\}$ be the set of agents who accept $\mathbf{x}$ with margin less than η (in the normalization of Section 2; see Figure 1a), and let $s_\eta := \lceil \log(2n)/(2\eta^2) \rceil$.

Theorem 4.2. *For every lottery $\mathbf{x}$ and every $\eta > 0$, there is a lottery $\tilde{\mathbf{x}}$ that uses at most s_η alternatives and satisfies $U_i(\tilde{\mathbf{x}}) \ge U_i(\mathbf{x}) - \eta$ for all $i \in N$. Consequently,* $\mathrm{viol}(\tilde{\mathbf{x}}) \le \mathrm{viol}(\mathbf{x}) + |B_\eta(\mathbf{x})|$. *In particular, if some optimal lottery $\mathbf{x}^*$ has $B_\eta(\mathbf{x}^*) = \varnothing$, then some optimal lottery uses at most s_η alternatives.*

Proof idea and relation to Castiglioni et al. [2023]. Draw s_η alternatives independently from $\mathbf{x}$ and let $\tilde{\mathbf{x}}$ be their empirical distribution. For each agent, $U_i(\tilde{\mathbf{x}})$ is an average of s_η independent values in $[0, 1]$ with mean $U_i(\mathbf{x})$, so Hoeffding's inequality and a union bound over the n agents show that, with probability at least $1/2$, $U_i(\tilde{\mathbf{x}}) \ge U_i(\mathbf{x}) - \eta$ for every agent. This sampling step is the argument that Castiglioni et al. [2023] use for their relaxation of MAX-FS over the simplex, and we do not claim it as new. Their guarantee is for the relaxed problem, which counts a constraint as satisfied if it is violated by at most an additive slack. Theorem 4.2 instead counts violations at the original thresholds: an agent who accepts $\mathbf{x}$ with margin at least η still accepts $\tilde{\mathbf{x}}$, so the only agents who can become violated are those in $B_\eta(\mathbf{x})$. The full proof is in Appendix D.2. □

Proposition 4.3. *For every integer $s \ge 1$, a lottery minimizing* viol *among all lotteries that use at most s alternatives can be computed in time $(m(n+s))^{\mathcal{O}(s)}$.*

For each set of at most s alternatives, the algorithm applies Theorem 4.1 to the lotteries that use only those alternatives (proof in Appendix D.3). Combining it with Theorem 4.2 gives the following (proof in Appendix D.4).

Corollary 4.4. *For every $\eta > 0$, running the algorithm of Proposition 4.3 with $s = s_\eta$ takes time $(m(n+s_\eta))^{\mathcal{O}(s_\eta)}$ and returns a lottery $\widehat{\mathbf{x}}$ with* $\mathrm{viol}(\widehat{\mathbf{x}}) \le \mathrm{viol}(\mathbf{x}) + |B_\eta(\mathbf{x})|$ *for every lottery $\mathbf{x}$. In particular,* $\mathrm{viol}(\widehat{\mathbf{x}}) \le \mathrm{OPT} + |B_\eta(\mathbf{x}^*)|$ *for every optimal lottery $\mathbf{x}^*$, and $\widehat{\mathbf{x}}$ is optimal if $B_\eta(\mathbf{x}^*) = \varnothing$ for some optimal $\mathbf{x}^*$.*

The running time is not polynomial. For constant η, it is $(mn)^{\mathcal{O}(\log n)}$, which is quasi-polynomial, and it grows quickly as η decreases. The lottery $\tilde{\mathbf{x}}$ in the proof of Theorem 4.2 has all probabilities equal to multiples of $1/s_\eta$. Hence the guarantee of Corollary 4.4 is also obtained by the enumeration of Castiglioni et al. [2023], if each of the $\binom{m+s_\eta-1}{s_\eta}$ such lotteries is scored by its exact violation count, in time $\mathcal{O}(\binom{m+s_\eta-1}{s_\eta} nm)$. Proposition 4.3 returns the best lottery among all that use at most

s_η alternatives, which is never worse, at a higher running time. Castiglioni et al. [2023] also show that, under the Exponential Time Hypothesis, their relaxed problem requires quasi-polynomial time.

4.3 Few Violations

The running time of Theorem 4.1 is polynomial for constant m, but its exponent grows with m. When few violations are allowed, the exponential dependence can be limited to m and the number of violations together (proof in Appendix E.2).

Theorem 4.5. *Given all normalized pairs $(u_i, \tau_i)_{i \in N}$ and an integer $v \in \{0, \dots, n\}$, there is a deterministic algorithm that decides whether some lottery has at most v violations and, if so, returns one, using at most $m^v \cdot \mathrm{poly}(n, m)$ LP feasibility checks over $\Delta(S)$. Hence, when the utilities and thresholds are given explicitly, this problem is fixed-parameter tractable when parameterized by (v, m).*

Proof idea. If the acceptable sets of the remaining agents have a common point, an LP finds it. Otherwise, by Helly's theorem [Helly, 1923] in the $(m-1)$-dimensional affine hull of the simplex, at most m of these sets already have no common point. Such a subfamily can be found with at most n LP feasibility checks, by removing agents one at a time as long as the remaining sets still have no common point. A lottery with at most v violations must violate one of these at most m agents, so the algorithm branches on which of them to drop; the recursion has depth at most v and at most m branches per step. This is in the spirit of algorithms for optimization with few violated constraints [Matoušek, 1995].

The parameter cannot be reduced to the number of violations alone (proof in Appendix E.3). □

Theorem 4.6. *Given all normalized pairs $(u_i, \tau_i)_{i \in N}$ and an integer v, deciding whether some lottery has at most v violations is W[2]-hard when parameterized by v alone, even when all utilities are 0 or 1. Hence, unless $\mathrm{FPT} = \mathrm{W}[2]$, there is no algorithm with running time $f(v) \cdot \mathrm{poly}(n, m)$ for any function f.*

Proof idea. We reduce from HITTING SET: given a family of subsets of $[q]$ and an integer v, is there a set of at most v elements that meets every member of the family? As in the proof of Theorem 3.1, there is one alternative for each element and one extra alternative. The agent of element j accepts if and only if $x_j \geq 1/q$. For each set F in the family, $v+1$ identical agents accept if and only if $\sum_{j \in F} x_j \leq (|F| - 1)/q$, which is an acceptance condition with $0/1$ utilities. A lottery with at most v violations cannot violate these $v+1$ copies, so every set F contains an element j with $x_j < 1/q$; these elements form a hitting set whose size is the number of violations. Conversely, a hitting set C gives a lottery with exactly $|C|$ violations (probability $1/q$ on each element outside C).

Finally, when $m = 2$, the simplex is a segment and each acceptable set is an interval, so MIN-VIOL reduces to finding a point contained in the largest number of intervals, which takes $\mathcal{O}(n \log n)$ time by sorting the endpoints (Proposition E.1). □

5 Limited Queries: Learning from Sampled Agents

The previous sections studied the full-information problem, in which every agent's constraint is known. We now ask what can be guaranteed when far fewer than n agents are queried. In this section, k agents $i_1, \dots, i_k$ are sampled independently and uniformly from N (with replacement), and the acceptable sets of the sampled agents are recovered; Section 6 drops the sampling assumption. The results are statistical statements about uniform sampling. Each sampled agent costs at most T_ε queries, so a budget of Q queries allows $\lfloor Q/T_\varepsilon \rfloor$ samples; Appendix G states the resulting guarantees together with running times.

Theorem 5.1. *Fix $\delta \in (0, 1)$ and $k \geq m$. With probability at least $1 - \delta$, every lottery $\widehat{\mathbf{x}}$ that minimizes the number of sampled agents who reject it (counting repeated samples with multiplicity) satisfies $\mathrm{viol}(\widehat{\mathbf{x}}) \leq \mathrm{OPT} + 8n\sqrt{(m \log(2ek/m) + \log(8/\delta))/k}$.*

Proof idea. Agent i rejects $\mathbf{x}$ if and only if $\langle u_i - \tau_i \mathbf{1}, \mathbf{x} \rangle < 0$, so the set of agents rejecting $\mathbf{x}$ is cut out by a halfspace through the origin of $\mathbb{R}^m$. The family of these sets therefore has VC dimension at

most m, and the standard VC inequality gives uniform convergence of the sampled rejection rate to the population rate. The argument is standard; what is specific to lotteries is the bound m on the VC dimension. The proof is in Appendix F.1. □

The same sampling procedure also gives a relative guarantee when OPT is not too small. The proof replaces the VC inequality with the relative-error bound of Li et al. [2001], which bounds the difference between the sampled and true rejection rates by a small multiple of their sum plus ρ (Appendix F.2).

Theorem 5.2. *There is an absolute constant $C > 0$ such that the following holds. Fix $\delta, \rho \in (0, 1/2)$ and $\zeta \in (0, 1]$, and let $k \geq C(m \log(1/\rho) + \log(1/\delta))/(\zeta^2 \rho)$. With probability at least $1 - \delta$, every lottery $\widehat{\mathbf{x}}$ as in Theorem 5.1 satisfies $\mathrm{viol}(\widehat{\mathbf{x}}) \leq (1 + \zeta/2)\mathrm{OPT} + (\zeta/2)\rho n$. In particular, if $\mathrm{OPT} \geq \rho n$, then $\mathrm{viol}(\widehat{\mathbf{x}}) \leq (1 + \zeta)\mathrm{OPT}$.*

Theorems 5.1 and 5.2 concern how well a sample represents the population, assuming exact minimization on the sample, which Theorem 4.1 provides when m is constant. Section 5.1 handles arbitrary m.

5.1 A Polynomial-Time LP for Arbitrary m

When m is part of the input, exact minimization on the sample is an instance of MIN-VIOL, which is APX-hard (Theorem 3.1). We therefore replace each violation by a convex penalty. For a margin parameter $\eta \in (0, 1]$, agent i is charged $\max\{0, (\tau_i + \eta - U_i(\mathbf{x}))/\eta\}$: this is 0 if i accepts $\mathbf{x}$ with margin at least η, 1 on the acceptance boundary, and more than 1 if i rejects $\mathbf{x}$. Minimizing the average penalty over the sample is the linear program

$$\begin{aligned} \text{minimize} \quad & \frac{1}{k} \sum_{t=1}^{k} z_t \\ \text{subject to} \quad & z_t \geq \frac{1}{\eta} \left(\tau_{i_t} + \eta - \langle u_{i_t}, \mathbf{x} \rangle\right) \text{ and } z_t \geq 0 \quad \text{for all } t \in [k], \quad \mathbf{x} \in \Delta(S). \end{aligned} \tag{1}$$

Repeated copies of a sampled agent can be merged into one constraint with the corresponding weight in the objective.

Theorem 5.3. *There is an absolute constant $C > 0$ such that the following holds. Fix an integer $k \geq 1$, $\delta \in (0, 1)$, and $\eta \in (0, 1]$, and let $\widehat{\mathbf{x}}_\eta$ be the $\mathbf{x}$-part of any optimal solution of the linear program* (1)*. With probability at least $1 - \delta$, $\mathrm{viol}(\widehat{\mathbf{x}}_\eta) \leq (1 + 1/\eta)\mathrm{viol}(\mathbf{x}) + |B_\eta(\mathbf{x})| + (Cn/\eta)\sqrt{(\log m + \log(1/\delta))/k}$ for every lottery $\mathbf{x}$.*

Proof idea. The penalty is at least the violation indicator. For a fixed lottery $\mathbf{x}$, it is at most $1 + 1/\eta$ for an agent rejecting $\mathbf{x}$, at most 1 for an agent in $B_\eta(\mathbf{x})$, and 0 for every other agent, so the population average penalty of $\mathbf{x}$ is at most $((1 + 1/\eta)\mathrm{viol}(\mathbf{x}) + |B_\eta(\mathbf{x})|)/n$. The penalty is a $(1/\eta)$-Lipschitz function of $U_i(\mathbf{x}) - \tau_i$, which is a convex combination of the m functions $i \mapsto u_i(s_j) - \tau_i$; standard Rademacher complexity bounds [Bartlett and Mendelson, 2002] then give uniform convergence at rate $(1/\eta)\sqrt{\log m/k}$. The proof is in Appendix F.3.

The bound is useful when some lottery has few violations and few agents accepted with margin less than η. If $B_\eta(\mathbf{x}^*) = \varnothing$ for an optimal lottery $\mathbf{x}^*$, the bound is $(1 + 1/\eta)\mathrm{OPT}$ plus the sampling error; if some lottery is accepted by every agent with margin at least η, only the sampling error remains. For instance, in Example 2.1, $\mathrm{OPT} = 1$ and the optimal lottery $\mathbf{x}^*$ given there has $B_{1/15}(\mathbf{x}^*) = \varnothing$.

Full information. If the LP is run on all n agents, each once, instead of on a sample, the sampling term disappears: $\mathrm{viol}(\widehat{\mathbf{x}}_\eta) \leq (1 + 1/\eta)\mathrm{viol}(\mathbf{x}) + |B_\eta(\mathbf{x})|$ for every lottery $\mathbf{x}$. The factor $1 + 1/\eta$ comes from the penalty of violated agents, not from sampling. Capping the penalty at 1 would remove this factor, but the capped penalty is not convex, so minimizing it is no longer a linear program.

Which method to use. Exact minimization on the sample, combined with Theorem 5.1, avoids the factor $1 + 1/\eta$ and the term $|B_\eta(\mathbf{x})|$ of Theorem 5.3, and is preferable when it is affordable: when m is a small constant, when $m = 2$ (Proposition E.1), or when m and the optimal number of violations on the sample are both small (Theorem 4.5). Running Corollary 4.4 on the sample takes quasi-polynomial time and is exact only under its margin condition. The LP runs in polynomial time for every m and every sample, without any margin condition, and its sampling term grows with $\log m$ rather than m.

6 The Deterministic Worst Case

We now drop the sampling assumption. A *profile* $\mathcal{I} = ((u_i, \tau_i))_{i \in N}$ lists all normalized pairs; we write $\mathrm{viol}_{\mathcal{I}}(\mathbf{x})$ and $\mathrm{OPT}(\mathcal{I})$ for the violation count and its minimum under $\mathcal{I}$. For a deterministic algorithm $\mathcal{A}$, let $\mathcal{A}(\mathcal{I})$ be the lottery it returns when its queries are answered according to $\mathcal{I}$; its *additive gap* on $\mathcal{I}$ is $\mathrm{viol}_{\mathcal{I}}(\mathcal{A}(\mathcal{I})) - \mathrm{OPT}(\mathcal{I})$.

An algorithm that recovers a fixed set M of K agents and returns a lottery $\mathbf{x}_M$ minimizing the number of violated agents in M has additive gap at most $n - K$ on every profile. Indeed, if $\mathbf{x}^*$ is optimal for $\mathcal{I}$, then $\mathbf{x}_M$ violates at most as many agents of M as $\mathbf{x}^*$ does, and at most $n - K$ agents outside M, so $\mathrm{viol}_{\mathcal{I}}(\mathbf{x}_M) \le \mathrm{OPT}(\mathcal{I}) + n - K$. The question is whether adaptive queries, possibly many to the same agents, can do better. For $m \ge 3$ they cannot.

Theorem 6.1. *Fix $m \ge 3$ and a deterministic algorithm $\mathcal{A}$. Run $\mathcal{A}$ with every query $(i, \mathbf{x})$ answered "accept" if and only if $x_3 = \cdots = x_m = 0$, and let M be the set of agents it queries and $\widehat{\mathbf{x}}$ its output. Then there is a profile $\mathcal{I}$ with binary utilities and all thresholds equal to 1 such that $\mathcal{A}$, run on $\mathcal{I}$, receives the same answers and therefore returns $\widehat{\mathbf{x}}$, every agent outside M rejects $\widehat{\mathbf{x}}$, and all agents accept the same pure lottery $\mathbf{e}_1$ or $\mathbf{e}_2$. Hence $\mathrm{viol}_{\mathcal{I}}(\widehat{\mathbf{x}}) \ge n - |M|$ and $\mathrm{OPT}(\mathcal{I}) = 0$.*

Proof idea. Give every agent in M the utility vector $(1, 1, 0, \ldots, 0)$ and threshold 1: such an agent accepts exactly the lotteries with $x_3 = \cdots = x_m = 0$, which reproduces every answer. Give every agent outside M the utility vector $\mathbf{e}_1$ if $\widehat{\mathbf{x}} \ne \mathbf{e}_1$ and $\mathbf{e}_2$ if $\widehat{\mathbf{x}} = \mathbf{e}_1$, with threshold 1. Such an agent accepts only that pure lottery, so it rejects $\widehat{\mathbf{x}}$, while that pure lottery is accepted by every agent (Figure 1b). The answer rule needs a third alternative so that the queried agents reject some lotteries, as nondegeneracy requires, while accepting both $\mathbf{e}_1$ and $\mathbf{e}_2$. The proof is in Appendix H.1. □

Theorem 6.2. *Assume that each queried agent can be recovered with finitely many membership queries, as under the finite-grid convention, and let $m \ge 3$. For $K \in \{0, \ldots, n\}$, let $\mathcal{C}_K$ be the class of deterministic algorithms that query at most K distinct agents on every profile. Then $\inf_{\mathcal{A} \in \mathcal{C}_K} \sup_{\mathcal{I}} (\mathrm{viol}_{\mathcal{I}}(\mathcal{A}(\mathcal{I})) - \mathrm{OPT}(\mathcal{I})) = n - K$, where the supremum is over all profiles satisfying the model assumptions.*

The upper bound is the argument above, and the lower bound follows from Theorem 6.1 (proof in Appendix H.2); Teh and Goldberg [2026] prove an analogous lower bound for their distance-based objective. In the sampling model of Section 5, queried agents carry information about unqueried ones. In the worst case they carry none: for $m \ge 3$, the best guarantee depends only on how many distinct agents are queried, not on how many queries are spent on them or how adaptively they are chosen.

Bounds after a partial query history. The answers observed so far restrict each agent's normalized pair to the pairs consistent with them. For a lottery $\mathbf{x}$, counting the agents who reject $\mathbf{x}$ under every consistent pair, and those who reject it under some consistent pair, gives lower and upper bounds on its violation count that hold for every consistent profile; minimizing both over lotteries gives lower and upper bounds on the unknown optimum. The lottery that minimizes the upper count has additive gap at most the difference between the two bounds on the optimum. This difference can only shrink as more answers arrive, and every unqueried agent adds one to it, which is the effect behind Theorem 6.1. Appendix I proves these facts and gives the exact worst-case additive gap of any fixed lottery after the observed answers.

7 Synthetic Experiments

We ran small synthetic experiments to see how conservative the worst-case bounds are on random instances; Appendix J gives the setup in full. We generated 20 populations with $n = 500$ agents and $m = 3$ alternatives in each of two settings. In both, 15% of agents reject a planted lottery and 85% accept it, with margin at least 0.05 in the *separated* setting and less than 0.05 in the *near-threshold* setting. Since $m = 3$, we computed OPT exactly with the vertex enumeration of Theorem 4.1, and checked it against a mixed-integer program on 280 instances, which agreed in every case. For each population and sample size, we drew 6 independent samples. Table 1 reports the results.

Findings. (i) Exact minimization on 120 sampled agents is within 2% of the optimum on average, far below the explicit bound of Theorem 5.1, which exceeds n at these sample sizes. Each fourfold increase in k reduced the mean extra rejection rate by a factor between about 3.9 and 5.9, faster than

Table 1: Mean extra rejection rate $(\mathrm{viol}(\widehat{\mathbf{x}}) - \mathrm{OPT})/n$ in percent, with 95th percentiles in parentheses.

Method	Agents used	Separated	Near-threshold
Exact minimization on the sample	$k = 30$	6.81 (19.24)	6.25 (15.81)
Exact minimization on the sample	$k = 120$	1.67 (4.22)	1.36 (3.41)
Exact minimization on the sample	$k = 480$	0.43 (1.40)	0.23 (0.80)
LP (1), $\eta = 0.02$	$k = 120$	1.96 (5.60)	3.60 (6.80)
LP (1), $\eta = 0.05$	$k = 120$	0.53 (1.61)	9.20 (17.82)
LP (1), $\eta = 0.02$	all 500	0.51 (1.67)	1.86 (3.80)
LP (1), $\eta = 0.05$	all 500	0.28 (0.80)	5.22 (8.22)

the factor 2 suggested by the $1/\sqrt{k}$ rate of Theorem 5.1. (ii) As the trade-off in η of Theorem 5.3 suggests, $\eta = 0.05$ was better than $\eta = 0.02$ in the separated setting, and on 120 samples even better than exact minimization, but more than doubled the extra rejection rate in the near-threshold setting, where $B_{0.05}(\mathbf{x})$ contains every agent who accepts the planted lottery $\mathbf{x}$. (iii) Querying $K = 100$ agents chosen uniformly at random and optimizing over them gave mean additive gaps of 8.46 and 7.99 agents, against the worst-case value $n - K = 400$; across the tested values of K, the mean gap was between about 0.6% and 19% of $n - K$. The construction of Theorem 6.1 reached a gap of exactly $n - K$ for every tested $K \in \{10, 25, 50, 100, 200, 400\}$, so the worst case can occur, but in our experiments only for a specially chosen profile. (iv) With $n = 1000$, $m = 500$, and $\eta = 0.15$, Theorem 4.2 gives $s_\eta = 169$. When every agent accepted the starting lottery with margin at least η, sampling 169 alternatives as in its proof never created a new rejection in 480 trials, and sampling only 50 created none in 471 of them. When 20% of agents accepted it with margin less than η, the mean new rejection rate with 169 alternatives was 1.19% (95th percentile 1.70%), whereas the theorem allows up to 20%.

8 Conclusion

We studied how to choose a lottery that as few agents as possible reject, in the membership query model of Choo et al. [2026b]. The problem is APX-hard even with full information, but it can be solved exactly under additional structure; under uniform sampling, a small number of queried agents gives guarantees for the whole population, and without sampling, for $m \geq 3$, the number of distinct queried agents alone determines the worst-case guarantee of deterministic algorithms.

Limitations and extensions. The halfspace model covers thresholds on expected access, safety, service, or loss, but not acceptability that depends on variance, ambiguity, or nonlinear probability weighting (Appendix B.1). Theorem 5.1 needs only a bound on the VC dimension of the rejection sets: if another acceptance model gives VC dimension d, the same proof applies with d in place of m. Theorems 4.1, 4.2, and 5.3 use linearity directly, through linear boundaries, expected utility as an average over alternatives, and a piecewise-linear penalty, respectively; they do not extend automatically to general convex acceptable sets. The lower bound of Theorem 6.1 does extend, since halfspaces are a special case of convex sets.

Our results assume noiseless answers. Suppose instead that each answer is flipped with probability $p < 1/2$, independently across queries, including repeated ones. Repeating a query r times and taking the majority answer, breaking ties arbitrarily, is wrong with probability at most $\exp(-r(1-2p)^2/2)$ by Hoeffding's inequality. If the noiseless procedure makes at most $Q \geq 1$ queries, then, for $\delta \in (0, 1)$, choosing $r = \lceil 2\log(Q/\delta)/(1-2p)^2 \rceil$ and stopping after at most Q simulated queries makes every majority answer correct with probability at least $1 - \delta$. All noiseless guarantees therefore continue to hold, with the probability that they hold reduced by at most δ and with at most rQ queries in total. Persistent or adversarial errors, or preferences that change between queries, require a different model.

Acknowledgments and Disclosure of Funding

This work was funded by the Advanced Research + Invention Agency (ARIA) under the ASPAI project. PWG is also supported by EPSRC Grants EP/X040461/1 and EP/X038548/1.

References

Atila Abdulkadiroğlu and Tayfun Sönmez. School choice: A mechanism design approach. *American Economic Review*, 93(3):729–747, 2003.

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 60–69, 2018.

Priyank Agrawal, Eric Balkanski, Vasilis Gkatzelis, Tingting Ou, and Xizhi Tan. Learning-augmented mechanism design: Leveraging predictions for facility location. *Mathematics of Operations Research*, 49(4):2626–2651, 2024.

Stéphane Airiau, Haris Aziz, Ioannis Caragiannis, Justin Kruger, Jérôme Lang, and Dominik Peters. Portioning using ordinal preferences: Fairness and efficiency. *Artificial Intelligence*, 314:103809, 2023.

Paola Alimonti and Viggo Kann. Some APX-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1–2):123–134, 2000.

Edoardo Amaldi and Viggo Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theoretical Computer Science*, 147(1–2):181–210, 1995.

Edoardo Amaldi and Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1–2):237–260, 1998.

Edoardo Amaldi, Marc E. Pfetsch, and Leslie E. Trotter, Jr. On the maximum feasible subsystem problem, IISs and IIS-hypergraphs. *Mathematical Programming*, 95:533–554, 2003.

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

Francis J. Anscombe and Robert J. Aumann. A definition of subjective probability. *The Annals of Mathematical Statistics*, 34(1):199–205, 1963.

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54 (2):317–331, 1997.

Haris Aziz, Anna Bogomolnaia, and Hervé Moulin. Fair mixing: The case of dichotomous preferences. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, pages 753–781, 2019.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Eric Balkanski, Vasilis Gkatzelis, and Xizhi Tan. Strategyproof scheduling with predictions. In *Proceedings of the 14th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 11:1–11:22, 2023.

Siddhartha Banerjee, Vasilis Gkatzelis, Artur Gorokh, and Billy Jin. Online Nash social welfare maximization with predictions. In *Proceedings of the 33rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1–19, 2022.

Siddhartha Banerjee, Vasilis Gkatzelis, Safwan Hossain, Billy Jin, Evi Micha, and Nisarg Shah. Proportionally fair online allocation of public goods with predictions. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 20–28, 2023.

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Viktor Bengs, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. Preference-based online learning with dueling bandits: A survey. *Journal of Machine Learning Research*, 22:1–108, 2021.

Ben Berger, Michal Feldman, Vasilis Gkatzelis, and Xizhi Tan. Learning-augmented metric distortion via (p, q)-veto core. In *Proceedings of the 25th ACM Conference on Economics and Computation (EC)*, page 984, 2024.

Anna Bogomolnaia and Hervé Moulin. A new solution to the random assignment problem. *Journal of Economic Theory*, 100(2):295–328, 2001.

Anna Bogomolnaia, Hervé Moulin, and Richard Stong. Collective choice under dichotomous preferences. *Journal of Economic Theory*, 122(2):165–184, 2005.

Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence*, 170(8–9): 686–713, 2006.

Felix Brandt. Rolling the dice: Recent results in probabilistic social choice. In Ulle Endriss, editor, *Trends in Computational Social Choice*, chapter 1, pages 3–26. AI Access, 2017.

Darius Braziunas and Craig Boutilier. Elicitation of factored utilities. *AI Magazine*, 29(4):79–92, 2008.

Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

Giuseppe C. Calafiore and Marco C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

Marco C. Campi and Simone Garatti. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, 19(3):1211–1230, 2008.

Matteo Castiglioni, Andrea Celli, and Nicola Gatti. Public Bayesian persuasion: Being almost optimal and almost persuasive. *Algorithmica*, 85:2885–2921, 2023.

Timothy M. Chan. Low-dimensional linear programming with violations. *SIAM Journal on Computing*, 34(4):879–893, 2005.

Davin Choo, Winston Fu, Tzeh Yuan Neoh, Tze-Yang Poon, and Nicholas Teh. Approximate proportionality in online fair division. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, pages 20417–20450, 2026a.

Davin Choo, Paul W. Goldberg, and Nicholas Teh. Learning unanimously acceptable lotteries via queries. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, pages 20451–20493, 2026b. Extended version available as arXiv:2604.17505.

Saar Cohen, Nicholas Teh, Paul W. Goldberg, and Michael J. Wooldridge. Online fair division with budget constraints. arXiv:2607.23310, 2026.

Vincent Conitzer, Rachel Freedman, Jobst Heitzig, Wesley H. Holliday, Bob M. Jacobs, Nathan Lambert, Milan Mosse, Eric Pacuit, Stuart Russell, Hailey Schoelkopf, Emanuel Tewolde, and William S. Zwicker. Position: Social choice should guide AI alignment in dealing with diverse human feedback. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 9346–9360, 2024.

Khaled Elbassioni, Rajiv Raman, Saurabh Ray, and René Sitters. On the approximability of the maximum feasible subsystem problem with 0/1-coefficients. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1210–1219, 2009.

Edith Elkind, Matthias Greger, Patrick Lederer, Warut Suksompong, and Nicholas Teh. Settling the score: Portioning with cardinal preferences. *Artificial Intelligence*, 352:104487, 2026.

European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). *Official Journal of the European Union, L series*, 2024. 12 July 2024.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

Peter C. Fishburn. *The Foundations of Expected Utility*, volume 31 of *Theory and Decision Library*. D. Reidel, Dordrecht, 1982.

Rupert Freeman and Ulrike Schmidt-Kraepelin. Project-fair and truthful mechanisms for budget aggregation. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 9704–9712, 2024.

Rupert Freeman, David M. Pennock, Dominik Peters, and Jennifer Wortman Vaughan. Truthful aggregation of budget proposals. *Journal of Economic Theory*, 193:105234, 2021.

Itzhak Gilboa and David Schmeidler. Maxmin expected utility with non-unique prior. *Journal of Mathematical Economics*, 18(2):141–153, 1989. doi: 10.1016/0304-4068(89)90018-9.

Tarleton Gillespie. *Custodians of the Internet: Platforms, Content Moderation, and the Hidden Decisions That Shape Social Media*. Yale University Press, 2018.

Robert Gorwa, Reuben Binns, and Christian Katzenbach. Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society*, 7(1), 2020.

Steve Hanneke. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 7(2–3):131–309, 2014.

Eduard Helly. Über Mengen konvexer Körper mit gemeinschaftlichen Punkten. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 32:175–176, 1923.

Yichen Huang and Nicholas Teh. Online fair division with upgrade recourse. arXiv:2610.16943, 2026.

Aanund Hylland and Richard Zeckhauser. The efficient allocation of individuals to positions. *Journal of Political Economy*, 87(2):293–314, 1979.

Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, 1979. doi: 10.2307/1914185.

Ron Kohavi, Diane Tang, and Ya Xu. *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing*. Cambridge University Press, 2020.

Yi Li, Philip M. Long, and Aravind Srinivasan. Improved bounds on the sample complexity of learning. *Journal of Computer and System Sciences*, 62(3):516–527, 2001.

James Luedtke and Shabbir Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19(2):674–699, 2008.

Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *Journal of the ACM*, 68(4):24:1–24:25, 2021.

Mark J. Machina. "Expected Utility" Analysis without the Independence Axiom. *Econometrica*, 50 (2):277–323, 1982. doi: 10.2307/1912631.

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. doi: 10.1111/j.1540-6261.1952.tb01525.x.

Jiří Matoušek. On geometric optimization with few violated constraints. *Discrete & Computational Geometry*, 14:365–384, 1995.

Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*, pages 646–662. Cambridge University Press, 2021.

Tzeh Yuan Neoh and Nicholas Teh. Closing gaps in online fair division. arXiv:2609.05310, 2026a.

Tzeh Yuan Neoh and Nicholas Teh. Online fair division with frequency predictions. arXiv:2610.34625, 2026b.

Tzeh Yuan Neoh, Jannik Peters, and Nicholas Teh. Online fair division with additional information. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, pages 92207–92241, 2026a.

Tzeh Yuan Neoh, Nicholas Teh, Je Qin Chooi, Paul W. Goldberg, and Milind Tambe. Efficient ensemble selection from binary and pairwise feedback. arXiv:2605.09588, 2026b.

Bernardo K. Pagnoncelli, Shabbir Ahmed, and Alexander Shapiro. Sample average approximation method for chance constrained programming: Theory and applications. *Journal of Optimization Theory and Applications*, 142:399–416, 2009.

Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ML predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 9661–9670, 2018.

John Quiggin. A theory of anticipated utility. *Journal of Economic Behavior & Organization*, 3(4): 323–343, 1982. doi: 10.1016/0167-2681(82)90008-7.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 5377–5400, 2024.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell L. Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. Position: A roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 46280–46302, 2024.

Warut Suksompong and Nicholas Teh. Voting in divisible settings: A survey. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence (AAAI)*, pages 39789–39796, 2026.

Nicholas Teh and Paul W. Goldberg. Learning acceptable lotteries via queries: Minimizing aggregated violation distances. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026. To appear.

Nicholas Teh, Saar Cohen, Karen Frilya Celine, Benjamin Yu, and Michael J. Wooldridge. Online house allocation with subsidy. arXiv:2609.29691, 2026.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

Chenyang Xu and Pinyan Lu. Mechanism design with predictions. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 571–577, 2022.

A Additional Related Work

Table 2 compares the works closest to ours. The general objective of satisfying as many linear inequalities as possible is classical; what the table separates is the set over which one optimizes, what information about the constraints is available, and what kind of guarantee is proved.

Table 2: Comparison with the closest work. "Explicit" means that all constraints are given as input.

Work	Domain	Information	Objective	Guarantees
MAX-FS and MIN-ULR [Amaldi and Kann, 1995, 1998, Arora et al., 1997, Amaldi et al., 2003]	$\mathbb{R}^d$, general linear systems	Explicit	Maximize satisfied, or minimize unsatisfied, constraints	NP-hardness, approximation algorithms, inapproximability
MAX-FS with $0/1$ coefficients [Elbassioni et al., 2009]	Real variables	Explicit	Maximize satisfied constraints	Approximation algorithms and hardness
LP with violations [Matoušek, 1995, Chan, 2005]	$\mathbb{R}^d$, d fixed	Explicit	Best linear objective when a given number of constraints may be violated	Exact algorithms in fixed dimension
MAX-FS with additive slack [Castiglioni et al., 2023]	Simplex	Explicit	Maximize constraints satisfied up to an additive slack	Quasi-polynomial algorithm; lower bound under ETH
Chance-constrained and scenario optimization [Calafiore and Campi, 2006, Campi and Garatti, 2008, Luedtke and Ahmed, 2008, Pagnoncelli et al., 2009]	Convex or general sets	Samples of random constraints	Optimize a cost subject to a small violation probability	Sample sizes for feasibility of sampled solutions
Unanimous lotteries [Choo et al., 2026b]	Simplex	Membership queries	A lottery accepted by all agents, or report that none exists	Query upper and lower bounds; NP-hardness of maximizing accepting agents
Violation distances [Teh and Goldberg, 2026]	Simplex	Membership queries	Minimize a power mean of the distances to violated agents' acceptance boundaries	When queries determine these distances; exact and sampling-based algorithms; lower bound for deterministic algorithms
This paper	Simplex	Membership queries (explicit in Sections 3–4)	Minimize the number of rejecting agents	APX-hardness; exact and parameterized algorithms; sampling guarantees; LP; exact deterministic value $n - K$

Aggregated violation distances. Teh and Goldberg [2026] study the same query model with a different objective. Rather than counting the agents who reject a lottery, they measure how severely each agent is violated: after normalizing each agent's acceptance boundary so that distances are comparable across agents, they take the Euclidean distance from the lottery to the boundary of each violated agent, measured in the affine hull of the simplex, and minimize a power mean of these distances. They characterize when membership queries determine these distances, show that the optimum can then be found exactly, give sampling-based guarantees under a limited query budget, and prove that a deterministic algorithm that leaves some agents unqueried can incur a loss that grows with the fraction of such agents, in parallel with Theorem 6.1. The two objectives are complementary. A distance-based objective distinguishes a slight violation from a severe one, but a small aggregate distance can still leave many agents slightly below their thresholds; the violation count treats every rejection as equally serious, which suits requirements that are either met or not. Computationally, the power mean of distances is convex once the relevant boundaries are known, whereas minimizing the violation count is APX-hard even with full information (Theorem 3.1).

Linear programming with violations. In fixed dimension, Matoušek [1995] and Chan [2005] study linear programs in which a given number of constraints may be violated, and give exact algorithms whose running time is polynomial in the number of constraints for fixed dimension. Their constraints are explicit and their objective is a linear function. These algorithms are closest in spirit to our full-information results for small m (Theorem 4.1 and Appendix E), which concern the violation count over the simplex; our main questions, however, concern what can be learned from membership queries.

Robust fitting and maximum consensus. The violation objective also resembles maximum-consensus and robust-fitting formulations, in which one seeks a model consistent with as many constraints or data points as possible [Fischler and Bolles, 1981]. These problems share the view that incompatible constraints should be discarded rather than averaged. However, robust fitting usually concerns parametric models fitted to observed data, with constraints or residuals available numerically. In our setting, the unknown constraints are agents' acceptable halfspaces of the simplex, and the decision maker observes only binary answers to lotteries that it proposes. Our deterministic lower bound is therefore not a standard robust-estimation lower bound.

Preference elicitation, utility elicitation, and active querying. Our membership query model is motivated by preference and utility elicitation, where full utilities are costly, sensitive, or cognitively difficult to obtain [Boutilier et al., 2006, Braziunas and Boutilier, 2008]. Much of this work uses value queries, comparison queries, bound queries, or minimax-regret criteria to choose good decisions under utility uncertainty. We differ in both the query and the objective. A query in our model asks whether a concrete lottery is acceptable, not for a utility value, a pairwise comparison, or a bound on a parameter. The goal is not to optimize expected utility or regret for a single partially known utility function, but to minimize the number of agents below their individual thresholds. Recovering sampled agents is one step of our algorithms, but what they output is a lottery with a guarantee for the whole population, not a utility model for every agent.

Membership query learning, active learning, and preference-based learning. Membership queries are classical in computational learning theory [Angluin, 1988], and active learning studies how adaptively chosen queries can reduce label complexity [Hanneke, 2014]. Preference-based learning and dueling bandits instead infer rewards or rankings from comparisons [Bengs et al., 2021]. Our problem is neither concept learning in the usual sense nor online preference optimization. Each agent has a hidden halfspace of the simplex, but the output is a single lottery that is evaluated on all agents, not a classifier that predicts future labels. This matters for the deterministic lower bound: repeated queries to the same agents may reveal their halfspaces, but they give no information about unqueried agents.

Statistical learning, convex penalties, and chance constraints. The sampling results can be viewed as a generalization problem: each lottery $\mathbf{x}$ defines a binary loss on agents, equal to 1 if the agent rejects $\mathbf{x}$ and 0 otherwise. The linear acceptability assumption then plays the role of a linear threshold hypothesis class in supervised learning: it limits the possible rejection sets enough for uniform convergence from finitely many sampled agents, without claiming that every acceptability rule is linear [Vapnik, 1998, Anthony and Bartlett, 1999, Shalev-Shwartz and Ben-David, 2014]. The LP replaces the discontinuous violation indicator with a one-sided penalty that is an upper bound on it. Convex upper bounds on discontinuous binary losses are classical [Bartlett et al., 2006], and our analysis uses the behavior of Rademacher complexity under convex hulls and Lipschitz maps [Bartlett and Mendelson, 2002]. What is specific to lotteries is that the penalty controls the true violation count through the set $B_\eta(\mathbf{x})$ of accepted agents with small margin. Chance-constrained and scenario optimization seek decisions that violate a random constraint with small probability [Calafiore and Campi, 2006, Campi and Garatti, 2008, Luedtke and Ahmed, 2008, Pagnoncelli et al., 2009]. Our population is finite and accessed through queries rather than given by a distribution, and our deterministic results show that, without a sampling assumption, worst-case guarantees depend only on the number of distinct agents queried.

Learning-augmented algorithms. Theorem 6.1 shows that, without random sampling, nothing can be guaranteed for the agents who are not asked. Learning-augmented algorithms circumvent worst-case limits of this kind in a different way: they receive predictions that may be erroneous, and aim for guarantees that improve with the accuracy of the predictions while remaining reasonable

when the predictions are poor [Lykouris and Vassilvitskii, 2021, Purohit et al., 2018, Mitzenmacher and Vassilvitskii, 2021]. This approach has been applied to mechanism design [Xu and Lu, 2022, Balkanski et al., 2023, Agrawal et al., 2024], online fair division [Banerjee et al., 2022, 2023, Choo et al., 2026a, Cohen et al., 2026, Huang and Teh, 2026, Neoh et al., 2026a, Neoh and Teh, 2026a,b, Teh et al., 2026], and voting [Berger et al., 2024]. In particular, Choo et al. [2026b] used advice, such as a promising lottery or the agents whose requirements are likely to be binding, to find a unanimously acceptable lottery with fewer queries. Our guarantees rely on random sampling rather than on predictions, and combining the two is an interesting direction for future work.

B Omitted Material from Section 2

B.1 On the linear acceptability assumption

Formally, the model assumes that each agent's acceptable set is a halfspace of the simplex: $A_i = \{\mathbf{x} \in \Delta(S) : \langle u_i, \mathbf{x}\rangle \geq \tau_i\}$. This is an assumption about the shape of the hidden acceptable sets, not an assumption that agents report numerical utilities. The decision maker observes neither u_i nor τ_i; it only observes whether proposed lotteries are acceptable. Our information model is therefore weaker than elicitation models that query utility values, comparisons, bounds, or regret-relevant parameters [Boutilier et al., 2006, Braziunas and Boutilier, 2008].

Risk attitudes. The assumption does not require risk neutrality over final outcomes. Suppose alternative s_j induces, for agent i, a distribution P_{ij} over consequences, and the agent evaluates consequences with a von Neumann–Morgenstern utility function v_i [von Neumann and Morgenstern, 1944, Fishburn, 1982]. Under the standard expected utility treatment of compound lotteries [Anscombe and Aumann, 1963], if the decision maker first draws $j \sim \mathbf{x}$ and then deploys s_j, the agent's expected utility is $\sum_{j=1}^{m} x_j \, \mathbb{E}_{c\sim P_{ij}}[v_i(c)]$, so the expected consequence utilities, normalized as in Section 2, give exactly $U_i(\mathbf{x}) = \langle u_i, \mathbf{x}\rangle$. Risk attitudes over consequences are part of v_i; for example, if consequences are monetary, a concave v_i represents risk aversion. The linearity is only in the decision maker's random choice among finitely many alternatives.

Scores. In many platform, allocation, and public policy settings, the alternatives are deterministic rules, and the lottery describes randomized deployment across cohorts, applicants, time periods, or cases. If agent i's requirement concerns an average quantity, oriented so that larger is better, such as expected access probability, service level, safety, compliance, or the negative of an expected loss, then the value of a mixture is exactly the convex combination of the values of its components. Here $u_i(s_j)$ is not a psychological utility but a score of alternative s_j, and τ_i is the minimum acceptable score. For example, if $u_i(s_j) = 1$ for safe alternatives and 0 for unsafe ones, the requirement that an unsafe alternative is chosen with probability at most $1/10$ is $\langle u_i, \mathbf{x}\rangle \geq 9/10$. This expected score model is the standard cardinal baseline for evaluating lotteries in random assignment and probabilistic social choice [Hylland and Zeckhauser, 1979, Brandt, 2017], although that literature also studies ordinal preferences and other ways of comparing lotteries.

A learning-theoretic view. Each lottery $\mathbf{x}$ defines a rejection region $\{(u, \tau) : \langle u, \mathbf{x}\rangle < \tau\}$ in the space of agent types. These regions form a finite-dimensional family of linear threshold regions, analogous to linear classifiers in statistical learning [Vapnik, 1998, Anthony and Bartlett, 1999, Shalev-Shwartz and Ben-David, 2014], which is what makes uniform convergence from sampled agents possible. Richer acceptability models give different families of regions and require different sample complexity, query complexity, and optimization analyses.

Scope. The model does not capture agents whose acceptability depends on variance, nonlinear probability weighting, ambiguity about the randomization, regret, or other departures from expected utility [Markowitz, 1952, Kahneman and Tversky, 1979, Quiggin, 1982, Machina, 1982, Gilboa and Schmeidler, 1989], nor requirements on the distribution of realized outcomes that cannot be written as a linear threshold. For example, an agent who trades off the mean and variance of realized outcomes generally has a nonlinear acceptable set. Section 8 explains which of our results extend beyond halfspaces. The linear model is nevertheless not trivial: by Theorem 3.1, minimizing violations is APX-hard even with full information and binary utilities, so the difficulty does not come from complicated utilities or nonlinear behavior.

B.2 Normalization and finite precision

As noted in Section 2, membership queries identify an acceptable set A_i, not a pair (u_i, τ_i), so margins $U_i(\mathbf{x}) - \tau_i$ are meaningful only after fixing a normalization. The normalization of Section 2 is justified by the following elementary fact.

Lemma B.1. *Let $A \subsetneq \Delta(S)$ be nonempty and suppose $A = \{\mathbf{x} \in \Delta(S) : \langle v, \mathbf{x}\rangle \geq \theta\}$ for some nonconstant $v \in [0,1]^m$ and some $\theta \in \mathbb{R}$. Then exactly one of the following holds.*

(i) A has nonempty relative interior in $\Delta(S)$. In this case there is a unique pair (u, τ) with $A = \{\mathbf{x} \in \Delta(S) : \langle u, \mathbf{x}\rangle \geq \tau\}$, $\min_{j\in[m]} u(s_j) = 0$, and $\max_{j\in[m]} u(s_j) = 1$.

(ii) A has empty relative interior in $\Delta(S)$. In this case $A = \Delta(T) := \{\mathbf{x} \in \Delta(S) : x_j = 0 \text{ for all } j \notin T\}$ for a unique nonempty proper subset $T \subsetneq [m]$, and A is also given by $u(s_j) = \mathbf{1}\{j \in T\}$ and $\tau = 1$.

In both cases, the normalized pair gives exactly the same accept/reject answers as the original pair.

Proof. Let $v_{\min} := \min_j v(s_j)$ and $v_{\max} := \max_j v(s_j)$. Since A is nonempty and not all of $\Delta(S)$, we have $v_{\min} < \theta \leq v_{\max}$.

If $\theta = v_{\max}$, then a lottery is accepted if and only if it puts all probability on alternatives attaining the maximum score, so $A = \Delta(T)$ with $T := \{j \in [m] : v(s_j) = v_{\max}\}$. Since v is nonconstant, T is a nonempty proper subset of $[m]$, so A has empty relative interior. Moreover, T is determined by A, as it is the set of j with $\mathbf{e}_j \in A$, and the indicator pair in (ii) gives exactly the set A.

Suppose now that $\theta < v_{\max}$. Then A contains all lotteries close enough to a pure lottery $\mathbf{e}_j$ with $v(s_j) = v_{\max}$, so it has nonempty relative interior. Let (u', τ') be any pair with $A = \{\mathbf{x} \in \Delta(S) : \langle u', \mathbf{x}\rangle \geq \tau'\}$, and consider the affine functions $f(\mathbf{x}) := \langle v, \mathbf{x}\rangle - \theta$ and $f'(\mathbf{x}) := \langle u', \mathbf{x}\rangle - \tau'$ on the affine hull of $\Delta(S)$. Since f is positive near a pure lottery $\mathbf{e}_j$ with $v(s_j) = v_{\max}$ and negative near one with $v(s_j) = v_{\min}$, it vanishes at some point $\mathbf{p}$ in the relative interior of $\Delta(S)$. Near $\mathbf{p}$, both $\{f \geq 0\}$ and $\{f' \geq 0\}$ coincide with A; hence f' is not constant, the zero sets of f and f' agree near $\mathbf{p}$ and therefore coincide, and both functions are nonnegative on the same side. Thus $f' = \lambda f$ on the affine hull for some $\lambda > 0$, and since $\langle \mathbf{1}, \mathbf{x}\rangle = 1$ there, we have $u' = \lambda v + \mu\mathbf{1}$ and $\tau' = \lambda\theta + \mu$ for some $\mu \in \mathbb{R}$. The conditions $\min_j u'(s_j) = 0$ and $\max_j u'(s_j) = 1$ hold exactly when $\lambda = 1/(v_{\max} - v_{\min})$ and $\mu = -\lambda v_{\min}$; conversely, with these values, the pair $(\lambda v + \mu\mathbf{1}, \lambda\theta + \mu)$ gives A and satisfies both conditions. Hence the normalized pair exists and is unique. □

The finite-grid convention of Section 2 is used only for explicit query budgets and running times; the geometric, statistical, and lower-bound results hold for arbitrary real-valued normalized pairs unless they explicitly invoke it. There are two reasons to make it explicit. First, exact recovery of real-valued coefficients from finitely many binary answers is impossible without a finite description or a tolerance. This already happens when $m = 2$: if an agent accepts $\lambda\mathbf{e}_1 + (1-\lambda)\mathbf{e}_2$ if and only if $\lambda \geq a$, a query at λ returns only whether $\lambda \geq a$. There are only countably many finite binary answer sequences, while a ranges over a continuum. Second, the grid gives every recovered pair a finite description, so the algorithms that use recovered agents have well-defined running times. We use the per-agent bound $T_\varepsilon = \mathcal{O}(m \log(1/\varepsilon))$ of Choo et al. [2026b] as a black box. The convention does not discretize the lottery: the output may be any point of $\Delta(S)$.

C Proof of Theorem 3.1

We give a polynomial-time approximation-preserving reduction from MINIMUM VERTEX COVER on cubic graphs, which is APX-hard [Alimonti and Kann, 2000]. The idea is that vertex agents ask for probability at least $1/q$ on their vertex, while many copies of each edge agent forbid this much probability on both endpoints of an edge; the vertices that do not receive probability $1/q$ then form a vertex cover.

Let $G = (V, E)$ be a cubic graph with vertices $V = \{v_1, \ldots, v_q\}$. Set $m := q + 1$, where s_j corresponds to v_j for $j \in [q]$ and s_m is an extra alternative. Create q vertex agents $a_1, \ldots, a_q$ and,

for each edge $e \in E$, q^2 copies $b_{e,1}, \ldots, b_{e,q^2}$ of an edge agent. Then $n = q + q^2|E| = \mathcal{O}(q^3)$, since $|E| = 3q/2$. The utilities and thresholds are as follows.

- For each $j \in [q]$, the *vertex agent* a_j has threshold $1/q$, utility 1 on s_j, and utility 0 on every other alternative. Thus $U_{a_j}(\mathbf{x}) = x_j$, and a_j accepts $\mathbf{x}$ if and only if $x_j \geq 1/q$.
- For each edge $e = \{v_j, v_{j'}\} \in E$, every copy $b_{e,r}$ has threshold $1 - 1/q$, utility 0 on s_j and $s_{j'}$, and utility 1 on every other alternative. Thus $U_{b_{e,r}}(\mathbf{x}) = 1 - x_j - x_{j'}$, and $b_{e,r}$ accepts $\mathbf{x}$ if and only if $x_j + x_{j'} \leq 1/q$.

Every utility vector lies in $\{0,1\}^m$ and takes both values 0 and 1, and every threshold lies in $(0,1)$ (a cubic graph has $q \geq 4$ vertices), so all pairs are normalized. Every agent is nondegenerate: a_j accepts $\mathbf{e}_j$ and rejects $\mathbf{e}_m$, while each edge agent of e accepts $\mathbf{e}_m$ and rejects the pure lotteries on the endpoints of e. Let $\mathrm{vc}(G)$ be the minimum size of a vertex cover of G. We show that $\mathrm{OPT} = \mathrm{vc}(G)$ for the constructed instance and that every lottery can be turned into a vertex cover of size at most its violation count.

$\mathrm{OPT} \leq \mathrm{vc}(G)$. Let $C \subseteq V$ be a vertex cover, and define $\mathbf{x}^C \in \Delta(S)$ by $x^C_j = 1/q$ if $v_j \notin C$, $x^C_j = 0$ if $v_j \in C$, and $x^C_m := 1 - (q - |C|)/q = |C|/q$. For every edge, at most one endpoint lies outside C, so the probabilities of its endpoints sum to at most $1/q$ and all its edge agents accept $\mathbf{x}^C$. The vertex agent a_j accepts $\mathbf{x}^C$ if and only if $v_j \notin C$. Therefore $\mathrm{viol}(\mathbf{x}^C) = |C|$, and minimizing over C gives $\mathrm{OPT} \leq \mathrm{vc}(G)$.

From lotteries to vertex covers. Given a lottery $\mathbf{x}$, let $C(\mathbf{x}) := V$ if $x_j + x_{j'} > 1/q$ for some edge $\{v_j, v_{j'}\} \in E$, and $C(\mathbf{x}) := \{v_j \in V : x_j < 1/q\}$ otherwise; $C(\mathbf{x})$ can be computed in polynomial time. In the first case, $\mathbf{x}$ violates all q^2 copies of some edge agent, so $|C(\mathbf{x})| = q \leq q^2 \leq \mathrm{viol}(\mathbf{x})$, and $C(\mathbf{x}) = V$ is a vertex cover. In the second case, no edge has both endpoints with probability at least $1/q$, since that would give $x_j + x_{j'} \geq 2/q$. Hence $C(\mathbf{x})$ is a vertex cover. No edge agent is violated, and the violated vertex agents are exactly those of the vertices in $C(\mathbf{x})$, so $|C(\mathbf{x})| = \mathrm{viol}(\mathbf{x})$. In both cases, $C(\mathbf{x})$ is a vertex cover with $|C(\mathbf{x})| \leq \mathrm{viol}(\mathbf{x})$. In particular, $\mathrm{vc}(G) \leq \mathrm{OPT}$, so $\mathrm{OPT} = \mathrm{vc}(G)$.

Approximation preservation. If $\mathbf{x}$ is an α-approximate solution of the constructed instance, then $|C(\mathbf{x})| \leq \mathrm{viol}(\mathbf{x}) \leq \alpha \cdot \mathrm{OPT} = \alpha \cdot \mathrm{vc}(G)$, so $C(\mathbf{x})$ is an α-approximate vertex cover of G. Hence a PTAS for MIN-VIOL would give a PTAS for MINIMUM VERTEX COVER on cubic graphs, which does not exist unless $\mathrm{P} = \mathrm{NP}$. Therefore MIN-VIOL is APX-hard, even with binary utilities.

D Omitted Proofs from Section 4

D.1 Proof of Theorem 4.1

The idea is that some optimal lottery is a vertex of the polytope of lotteries accepted by the agents that it satisfies, and every such vertex is the unique solution of a system of m linear equations; enumerating these systems therefore finds an optimal lottery. For any $N' \subseteq N$, let $P(N') := \{\mathbf{x} \in \Delta(S) : \langle u_i, \mathbf{x}\rangle \geq \tau_i \text{ for all } i \in N'\} = \bigcap_{i \in N'} A_i$, with the convention $P(\varnothing) = \Delta(S)$. We first prove two lemmas.

Lemma D.1. *Let $\mathbf{x}^* \in \arg\min_{\mathbf{x} \in \Delta(S)} \mathrm{viol}(\mathbf{x})$ and $N^* := \{i \in N : \mathbf{x}^* \in A_i\}$. Then there exists an extreme point $\bar{\mathbf{x}}$ of $P(N^*)$ such that $\mathrm{viol}(\bar{\mathbf{x}}) = \mathrm{viol}(\mathbf{x}^*)$.*

Proof. Since $\mathbf{x}^* \in P(N^*)$, the set $P(N^*)$ is nonempty. By the definition of A_i, $P(N^*) = \Delta(S) \cap \bigcap_{i \in N^*} \{\mathbf{x} \in \mathbb{R}^m : \langle u_i, \mathbf{x}\rangle \geq \tau_i\}$, so $P(N^*)$ is a compact polytope. Thus, $P(N^*)$ has at least one extreme point; fix any extreme point $\bar{\mathbf{x}} \in P(N^*)$. By definition of $P(N^*)$, we have $\langle u_i, \bar{\mathbf{x}}\rangle \geq \tau_i$ for every $i \in N^*$, so $\bar{\mathbf{x}}$ violates at most the agents in $N \setminus N^*$. Therefore, $\mathrm{viol}(\bar{\mathbf{x}}) \leq |N \setminus N^*| = \mathrm{viol}(\mathbf{x}^*)$. Since $\mathbf{x}^*$ is optimal, equality must hold and $\bar{\mathbf{x}}$ is also optimal. □

Lemma D.2. *Let $N' \subseteq N$ and let $\mathbf{x}$ be an extreme point of $P(N')$. Then there exist subsets $J \subseteq [m]$ and $N'_0 \subseteq N'$ with $|J| + |N'_0| = m - 1$ such that $\mathbf{x}$ is the unique solution to the linear system $\sum_{j=1}^m z_j = 1$, $z_j = 0$ for all $j \in J$, and $\langle u_i, \mathbf{z}\rangle = \tau_i$ for all $i \in N'_0$.*

Algorithm 1 Returns a lottery with the minimum number of violations.

Input: The normalized pairs $(u_i, \tau_i)_{i \in N}$.
Output: A lottery $\widehat{\mathbf{x}}$ minimizing viol.

$\widehat{\mathbf{x}} \leftarrow \mathbf{e}_1$
for each pair (J, N_0) with $J \subseteq [m]$, $N_0 \subseteq N$, and $|J| + |N_0| = m - 1$ **do**
  if $\mathcal{L}(J, N_0)$ has a unique solution $\mathbf{x}$, $\mathbf{x} \in \Delta(S)$, and $\mathrm{viol}(\mathbf{x}) < \mathrm{viol}(\widehat{\mathbf{x}})$ **then**
    $\widehat{\mathbf{x}} \leftarrow \mathbf{x}$
  end if
end for
return $\widehat{\mathbf{x}}$

Proof. Let $L := \{\mathbf{d} \in \mathbb{R}^m : \sum_{j=1}^m d_j = 0\}$ be the translation space of the affine hull of $\Delta(S)$, where $\dim(L) = m - 1$.

Let $T_0 := \{j \in [m] : x_j = 0\}$ and $T_1 := \{i \in N' : \langle u_i, \mathbf{x} \rangle = \tau_i\}$ be the sets of tight nonnegativity constraints and tight agent constraints at $\mathbf{x}$, respectively. Consider the family $\mathcal{F} := \{\mathbf{d} \mapsto d_j : j \in T_0\} \cup \{\mathbf{d} \mapsto \langle u_i, \mathbf{d} \rangle : i \in T_1\}$ of linear functionals on L.

We claim that $\mathcal{F}$ spans L^*. Indeed, suppose not. Then there exists a nonzero $\mathbf{d} \in L$ such that $d_j = 0$ for all $j \in T_0$ and $\langle u_i, \mathbf{d} \rangle = 0$ for all $i \in T_1$.

For each $j \notin T_0$, we have $x_j > 0$; and for each $i \in N' \setminus T_1$, we have $\langle u_i, \mathbf{x} \rangle > \tau_i$. Since there are only finitely many such indices, there exists $\alpha_0 > 0$ such that, for every $|\alpha| \leq \alpha_0$, we have $x_j + \alpha d_j \geq 0$ for all $j \notin T_0$ and $\langle u_i, \mathbf{x} + \alpha \mathbf{d} \rangle \geq \tau_i$ for all $i \in N' \setminus T_1$. For $j \in T_0$, we have $x_j + \alpha d_j = 0$. For $i \in T_1$, we have $\langle u_i, \mathbf{x} + \alpha \mathbf{d} \rangle = \langle u_i, \mathbf{x} \rangle + \alpha \langle u_i, \mathbf{d} \rangle = \tau_i$. Also, $\sum_{j=1}^m (x_j + \alpha d_j) = 1$ because $\mathbf{d} \in L$. Thus, $\mathbf{x} + \alpha \mathbf{d} \in P(N')$ for every $|\alpha| \leq \alpha_0$. Since $\mathbf{d} \neq 0$, this gives us a nontrivial line segment in $P(N')$ through $\mathbf{x}$, contradicting the fact that $\mathbf{x}$ is an extreme point. Therefore $\mathcal{F}$ spans L^*. Choose from $\mathcal{F}$ a basis $\{\mathbf{d} \mapsto d_j : j \in J\} \cup \{\mathbf{d} \mapsto \langle u_i, \mathbf{d} \rangle : i \in N_0'\}$ of L^*, where $J \subseteq T_0$, $N_0' \subseteq T_1$, and $|J| + |N_0'| = m - 1$.

We now show that $\mathbf{x}$ is the unique solution to the system $\sum_{j=1}^m z_j = 1$, $z_j = 0$ for all $j \in J$, and $\langle u_i, \mathbf{z} \rangle = \tau_i$ for all $i \in N_0'$. Clearly $\mathbf{x}$ satisfies this system. Conversely, if $\mathbf{z}$ is any solution, then $\mathbf{d} := \mathbf{z} - \mathbf{x} \in L$, $d_j = 0$ for all $j \in J$, and $\langle u_i, \mathbf{d} \rangle = 0$ for all $i \in N_0'$. Since the chosen functionals form a basis of L^*, this forces $\mathbf{d} = \mathbf{0}$, and thus $\mathbf{z} = \mathbf{x}$. Therefore, $\mathbf{x}$ is the unique solution. □

Now, for each pair (J, N_0) with $J \subseteq [m]$, $N_0 \subseteq N$, and $|J| + |N_0| = m - 1$, let $\mathcal{L}(J, N_0)$ denote the linear system $\sum_{j=1}^m x_j = 1$, $x_j = 0$ for all $j \in J$, and $\langle u_i, \mathbf{x} \rangle = \tau_i$ for all $i \in N_0$. Lemma D.2 shows that every extreme point of every polytope $P(N')$ is the unique solution of $\mathcal{L}(J, N_0)$ for some such pair (J, N_0). This motivates the following algorithm (Algorithm 1).

We first prove correctness. Let $\mathbf{x}^*$ be an optimal lottery and $N^* := \{i \in N : \mathbf{x}^* \in A_i\}$. By Lemma D.1, some extreme point $\bar{\mathbf{x}}$ of $P(N^*)$ is also optimal, and by Lemma D.2, $\bar{\mathbf{x}}$ is the unique solution of $\mathcal{L}(J, N_0)$ for some $J \subseteq [m]$ and $N_0 \subseteq N^*$ with $|J| + |N_0| = m - 1$. When Algorithm 1 considers this pair, it finds $\bar{\mathbf{x}} \in \Delta(S)$, so the returned lottery satisfies $\mathrm{viol}(\widehat{\mathbf{x}}) \leq \mathrm{viol}(\bar{\mathbf{x}}) = \mathrm{OPT}$; that is, it is optimal.

Next, we prove the running time of Algorithm 1. For each $t \in \{0, \ldots, m-1\}$, the number of pairs (J, N_0) with $|N_0| = t$ and $|J| = m - 1 - t$ is $\binom{n}{t}\binom{m}{m-1-t}$, so, by Vandermonde's identity, the total number of pairs is $\sum_{t=0}^{m-1} \binom{n}{t}\binom{m}{m-1-t} = \binom{n+m}{m-1}$. For each pair, the system $\mathcal{L}(J, N_0)$ has exactly m equations in m unknowns, so testing consistency and uniqueness and solving it can be done in $\mathcal{O}(m^3)$ time by Gaussian elimination. Evaluating $\mathrm{viol}(\mathbf{x})$ requires computing $\langle u_i, \mathbf{x} \rangle$ for all $i \in N$, which takes $\mathcal{O}(nm)$ time. Thus, the total running time is $\mathcal{O}(\binom{n+m}{m-1}(m^3 + nm))$.

Finally, the decision version (is OPT at most a given number?) is solved by running Algorithm 1 and comparing the violation count of the returned lottery with that number. Since $\binom{n+m}{m-1} \leq (n+m)^{m-1}$, the running time is polynomial for every constant m, and the decision problem is in XP when parameterized by m.

D.2 Proof of Theorem 4.2

The sampling step below is the argument of Castiglioni et al. [2023] for their relaxation of MAX-FS over the simplex; we include it for completeness. Draw $J_1, \dots, J_{s_\eta}$ independently from $\mathbf{x}$, i.e., $\Pr[J_t = j] = x_j$ for all $j \in [m]$ and $t \in [s_\eta]$, and let $\tilde{\mathbf{x}} := \frac{1}{s_\eta} \sum_{t=1}^{s_\eta} \mathbf{e}_{J_t}$ be their empirical distribution. Then $\tilde{\mathbf{x}} \in \Delta(S)$, and it uses at most s_η alternatives. For a fixed agent i, the values $u_i(s_{J_1}), \dots, u_i(s_{J_{s_\eta}})$ are independent, lie in $[0,1]$, and have mean $\sum_{j=1}^{m} x_j u_i(s_j) = U_i(\mathbf{x})$, and $U_i(\tilde{\mathbf{x}})$ is their average. Hoeffding's inequality gives $\Pr[U_i(\tilde{\mathbf{x}}) < U_i(\mathbf{x}) - \eta] \le e^{-2s_\eta \eta^2}$. By a union bound and the definition of s_η, $\Pr[\exists i \in N : U_i(\tilde{\mathbf{x}}) < U_i(\mathbf{x}) - \eta] \le n e^{-2s_\eta \eta^2} \le n e^{-\log(2n)} = 1/2$. Hence some realization of $\tilde{\mathbf{x}}$ satisfies $U_i(\tilde{\mathbf{x}}) \ge U_i(\mathbf{x}) - \eta$ for all $i \in N$; fix it.

We now bound the number of violations of $\tilde{\mathbf{x}}$. If agent i rejects $\tilde{\mathbf{x}}$, then $U_i(\mathbf{x}) < \tau_i + \eta$, since otherwise $U_i(\tilde{\mathbf{x}}) \ge U_i(\mathbf{x}) - \eta \ge \tau_i$; that is, agent i either rejects $\mathbf{x}$ or belongs to $B_\eta(\mathbf{x})$. Therefore $\mathrm{viol}(\tilde{\mathbf{x}}) \le \mathrm{viol}(\mathbf{x}) + |B_\eta(\mathbf{x})|$. If $\mathbf{x}^*$ is optimal and $B_\eta(\mathbf{x}^*) = \varnothing$, applying this with $\mathbf{x} = \mathbf{x}^*$ gives $\mathrm{viol}(\tilde{\mathbf{x}}) \le \mathrm{OPT}$, so $\tilde{\mathbf{x}}$ is an optimal lottery that uses at most s_η alternatives.

D.3 Proof of Proposition 4.3

For $T \subseteq [m]$, recall from Lemma B.1 that $\Delta(T)$ is the set of lotteries that use only alternatives in T. Every lottery that uses at most s alternatives lies in $\Delta(T)$ for $T = \{j \in [m] : x_j > 0\}$, so it suffices to minimize viol over $\Delta(T)$ for each T with $1 \le t := |T| \le \min\{s, m\}$ and keep the best solution.

Fix such a set T. If $t = 1$, then $\Delta(T)$ consists of a single pure lottery, whose violation count can be computed in $\mathcal{O}(n)$ time. Now suppose $t \ge 2$. On $\Delta(T)$, agent i accepts $\mathbf{x}$ if and only if $\sum_{j \in T} u_i(s_j)\, x_j \ge \tau_i$, so minimizing viol over $\Delta(T)$ is the same problem with the t alternatives in T. Some agents may accept or reject every lottery in $\Delta(T)$, but the proof of Theorem 4.1 uses only the simplex constraints, compactness, and the linear acceptance inequalities, so it applies with m replaced by t. Hence an optimizer over $\Delta(T)$ can be computed in time $\mathcal{O}(\binom{n+t}{t-1}(t^3 + nt))$, which also covers the case $t = 1$. There are $\binom{m}{t}$ sets T of size t, and for each $t \le s$ we have $\binom{m}{t} \le m^t$, $\binom{n+t}{t-1} \le (n+s)^{t-1}$, and $t^3 + nt \le 2(n+s)^3$; hence all sets of size t together take time $\mathcal{O}(m^t (n+s)^{t+2})$. Summing over the at most s values of t gives total time $\mathcal{O}(s\, m^s (n+s)^{s+2})$, which is $(m(n+s))^{\mathcal{O}(s)}$. The best lottery found over all such T minimizes viol among all lotteries that use at most s alternatives.

D.4 Proof of Corollary 4.4

The running time is that of Proposition 4.3 with $s = s_\eta$. Let $\widehat{\mathbf{x}}$ be the returned lottery and let $\mathbf{x}$ be any lottery. By Theorem 4.2, some lottery $\tilde{\mathbf{x}}$ that uses at most s_η alternatives satisfies $\mathrm{viol}(\tilde{\mathbf{x}}) \le \mathrm{viol}(\mathbf{x}) + |B_\eta(\mathbf{x})|$. Since $\widehat{\mathbf{x}}$ minimizes viol among all such lotteries, $\mathrm{viol}(\widehat{\mathbf{x}}) \le \mathrm{viol}(\tilde{\mathbf{x}}) \le \mathrm{viol}(\mathbf{x}) + |B_\eta(\mathbf{x})|$. Taking $\mathbf{x}$ to be an optimal lottery $\mathbf{x}^*$ gives $\mathrm{viol}(\widehat{\mathbf{x}}) \le \mathrm{OPT} + |B_\eta(\mathbf{x}^*)|$, which equals OPT if $B_\eta(\mathbf{x}^*) = \varnothing$.

E Omitted Material from Section 4.3

This appendix proves the algorithm for $m = 2$ mentioned in Section 4.3, and Theorems 4.5 and 4.6.

E.1 Exact optimization on a line segment: $m = 2$

Proposition E.1. *When $m = 2$, there is a deterministic algorithm that, given all normalized pairs $(u_i, \tau_i)_{i \in N}$, returns a lottery minimizing* viol *in time $\mathcal{O}(n \log n)$.*

Proof. Since $m = 2$, every lottery can be written uniquely as $\mathbf{x}_\lambda := (\lambda, 1-\lambda)$ with $\lambda \in [0,1]$. With two alternatives, the normalization of Section 2 gives $u_i = (1,0)$ or $u_i = (0,1)$ for every agent i. Hence the set $\Lambda_i := \{\lambda \in [0,1] : \mathbf{x}_\lambda \in A_i\}$ is the interval $[\ell_i, r_i]$, where $(\ell_i, r_i) = (\tau_i, 1)$ if $u_i = (1,0)$ and $(\ell_i, r_i) = (0, 1-\tau_i)$ if $u_i = (0,1)$. Since $n - \mathrm{viol}(\mathbf{x}_\lambda) = |\{i \in N : \lambda \in \Lambda_i\}|$ for every $\lambda \in [0,1]$, it suffices to find a point of $[0,1]$ contained in the largest number of these intervals, which Algorithm 2 does by sorting the endpoints.

Algorithm 2 Returns a lottery with the minimum number of violations when $m = 2$.

Input: The normalized pairs $(u_i, \tau_i)_{i \in N}$, with $S = \{s_1, s_2\}$.
Output: A lottery $\mathbf{x}^*$ minimizing viol.

for $i = 1, \dots, n$ **do**
  if $u_i = (1, 0)$ **then**
    Set $(\ell_i, r_i) \leftarrow (\tau_i, 1)$
  else
    Set $(\ell_i, r_i) \leftarrow (0, 1 - \tau_i)$
  end if
end for
Form the list of tagged endpoints $(\ell_i, \texttt{L})$ and $(r_i, \texttt{R})$ for every $i \in N$, together with $(0, \texttt{sent})$ and $(1, \texttt{sent})$
Sort this list by coordinate and scan it once to obtain the distinct coordinates $\lambda_1 < \cdots < \lambda_d$ and, for each $j \in [d]$, the counts $L_j := |\{i \in N : \ell_i = \lambda_j\}|$ and $R_j := |\{i \in N : r_i = \lambda_j\}|$
$c \leftarrow 0$, $c_{\max} \leftarrow -1$, $\lambda^* \leftarrow 0$
for $j = 1, \dots, d$ **do**
  $c \leftarrow c + L_j$
  if $c > c_{\max}$ **then**
    $c_{\max} \leftarrow c$
    $\lambda^* \leftarrow \lambda_j$
  end if
  $c \leftarrow c - R_j$
end for
return $\mathbf{x}^* := (\lambda^*, 1 - \lambda^*)$

We now prove correctness. Let $\lambda_1 < \cdots < \lambda_d$ and L_j, R_j be the quantities produced by Algorithm 2. We claim that, for each $j \in [d]$, immediately before the update $c \leftarrow c + L_j$, the value of c is $|\{i \in N : \ell_i < \lambda_j \le r_i\}|$. This is true for $j = 1$ because $\lambda_1 = 0$ (by the sentinels), initially $c = 0$, and no interval satisfies $\ell_i < 0$. If the claim holds at the start of iteration j, then after the update $c \leftarrow c + L_j$ we have $c = |\{i \in N : \ell_i \le \lambda_j \le r_i\}| = |\{i \in N : \lambda_j \in \Lambda_i\}|$. Thus, the algorithm evaluates the overlap number at λ_j. After this evaluation, the update $c \leftarrow c - R_j$ removes exactly the intervals with $r_i = \lambda_j$, so the resulting value of c is $|\{i \in N : \ell_i \le \lambda_j < r_i\}|$. For $j < d$, since there is no endpoint in $(\lambda_j, \lambda_{j+1})$, this equals $|\{i \in N : \ell_i < \lambda_{j+1} \le r_i\}|$, which is precisely the invariant at the start of iteration $j + 1$.

Now, let $\lambda \in [0, 1]$. If $\lambda = \lambda_j$ for some $j \in [d]$, its overlap number is explicitly evaluated by the algorithm. Otherwise, $\lambda \in (\lambda_j, \lambda_{j+1})$ for some consecutive coordinates $\lambda_j < \lambda_{j+1}$. Since no interval endpoint lies in $(\lambda_j, \lambda_{j+1})$, the set $\{i \in N : \lambda \in \Lambda_i\}$ is independent of the choice of λ inside this open interval. Moreover, if $\lambda \in \Lambda_i = [\ell_i, r_i]$, then $\ell_i \notin (\lambda_j, \lambda_{j+1})$ and $r_i \notin (\lambda_j, \lambda_{j+1})$, while $\ell_i \le \lambda \le r_i$. Hence $\ell_i \le \lambda_j$ and $r_i \ge \lambda_{j+1}$, so $\lambda_j \in \Lambda_i$. Therefore, $|\{i \in N : \lambda \in \Lambda_i\}| \le |\{i \in N : \lambda_j \in \Lambda_i\}|$. So a maximum overlap point can always be chosen among $\lambda_1, \dots, \lambda_d$, and the algorithm returns some $\lambda^* \in [0, 1]$ maximizing $|\{i \in N : \lambda \in \Lambda_i\}|$.

Using $n - \mathrm{viol}(\mathbf{x}_\lambda) = |\{i \in N : \lambda \in \Lambda_i\}|$, we conclude that $\mathbf{x}^* := \mathbf{x}_{\lambda^*} \in \arg\min_{\mathbf{x} \in \Delta(S)} \mathrm{viol}(\mathbf{x})$. Finally, computing all values ℓ_i, r_i takes $\mathcal{O}(n)$ time, sorting the $2n + 2$ tagged endpoints takes $\mathcal{O}(n \log n)$ time, the scan that forms $(\lambda_j, L_j, R_j)_{j=1}^d$ is linear, and the final sweep is linear. Therefore, the total running time is $\mathcal{O}(n \log n)$. □

E.2 Proof of Theorem 4.5

Proof. We first show that an infeasible family of acceptable sets contains an infeasible subfamily of at most m sets, and that such a subfamily can be found with few LP feasibility checks. We then show that branching on which agent of this subfamily to drop decides the problem. Throughout, we use the convention $\bigcap_{i \in \varnothing} A_i = \Delta(S)$.

Lemma E.2. *Let $N' \subseteq N$. If $\bigcap_{i \in N'} A_i = \varnothing$, then there exists $W \subseteq N'$ with $|W| \le m$ such that $\bigcap_{i \in W} A_i = \varnothing$.*

Algorithm 3 `HellyBranch`(N', v) (deletes up to v agents from N')

Input: Index set $N' \subseteq N$, integer $v \geq 0$.
Output: Either NO, or a lottery $\mathbf{x} \in \Delta(S)$ such that there exists $V \subseteq N'$ with $|V| \leq v$ and $\mathbf{x} \in \bigcap_{i \in N' \setminus V} A_i$.

1: **if** $\bigcap_{i \in N'} A_i \neq \varnothing$ **then**
2: return any $\mathbf{x} \in \bigcap_{i \in N'} A_i$ (found by LP feasibility)
3: **end if**
4: **if** $v = 0$ **then**
5: return NO
6: **end if**
7: Compute $W \subseteq N'$ with $|W| \leq m$ and $\bigcap_{i \in W} A_i = \varnothing$ (Lemma E.3)
8: **for** each $i \in W$ **do**
9: $\mathbf{x} \leftarrow$ `HellyBranch`$(N' \setminus \{i\}, v - 1)$
10: **if** $\mathbf{x} \neq$ NO **then**
11: return $\mathbf{x}$
12: **end if**
13: **end for**
14: return NO

Proof. Each A_i is a convex subset of the affine hull of $\Delta(S)$, which has dimension $m-1$. By Helly's theorem [Helly, 1923] in dimension $m-1$, if a finite family of convex subsets of that affine hull has empty intersection, then some subfamily of size at most $(m-1)+1 = m$ already has empty intersection. Applying this to $\{A_i\}_{i \in N'}$ proves the claim. □

Lemma E.3. *There is a deterministic procedure that, given $N' \subseteq N$ with $\bigcap_{i \in N'} A_i = \varnothing$, returns $W \subseteq N'$ such that $|W| \leq m$ and $\bigcap_{i \in W} A_i = \varnothing$ using at most $|N'|$ feasibility checks of linear programs over $\Delta(S)$.*

Proof. Initialize $W \leftarrow N'$, and fix an arbitrary ordering $i_1, \ldots, i_{|N'|}$ of the elements of N'. For $t = 1, \ldots, |N'|$, if $\bigcap_{\ell \in W \setminus \{i_t\}} A_\ell = \varnothing$, then set $W \leftarrow W \setminus \{i_t\}$.

This procedure performs at most $|N'|$ feasibility checks, one for each i_t. Moreover, throughout the procedure we maintain $\bigcap_{\ell \in W} A_\ell = \varnothing$, because we delete i_t only when the remaining family is still infeasible.

We claim that the final set W is inclusion-minimal infeasible. Indeed, fix any i_t that remains in the final set W. When i_t was processed, it was not deleted, so at that moment the current set, say W', satisfied $\bigcap_{\ell \in W' \setminus \{i_t\}} A_\ell \neq \varnothing$. Afterwards the algorithm only deletes more agents, so $W \setminus \{i_t\} \subseteq W' \setminus \{i_t\}$. Feasibility is preserved when constraints are removed, and thus $\bigcap_{\ell \in W \setminus \{i_t\}} A_\ell \neq \varnothing$. Therefore, W is inclusion-minimal infeasible.

By Lemma E.2, every infeasible family contains an infeasible subfamily of size at most m. If $|W| > m$, then applying Lemma E.2 to the family $\{A_i\}_{i \in W}$ would give us a proper infeasible subfamily of W, contradicting the inclusion-minimality of W. Therefore $|W| \leq m$. □

We run Algorithm 3 on input (N, v).

We prove the following stronger recursive claim: for every $N' \subseteq N$ and every integer $r \geq 0$, the call `HellyBranch`(N', r) returns NO if and only if there is no set $V \subseteq N'$ with $|V| \leq r$ such that $\bigcap_{i \in N' \setminus V} A_i \neq \varnothing$. Moreover, whenever `HellyBranch`$(N', r)$ returns a lottery $\mathbf{x}$, there exists such a set $V \subseteq N'$ with $\mathbf{x} \in \bigcap_{i \in N' \setminus V} A_i$. We prove the claim by induction on r.

If $r = 0$, then the algorithm returns a lottery exactly when $\bigcap_{i \in N'} A_i \neq \varnothing$ (Line 1), and otherwise returns NO (Lines 4–5). This is precisely the claim with $V = \varnothing$.

Now let $r \geq 1$ and assume the claim holds for $r-1$. If $\bigcap_{i \in N'} A_i \neq \varnothing$, then Line 2 returns some $\mathbf{x} \in \bigcap_{i \in N'} A_i$, so the claim holds with $V = \varnothing$.

Suppose instead that $\bigcap_{i\in N'} A_i = \varnothing$. By Lemma E.3, the algorithm computes a set $W \subseteq N'$ such that $|W| \le m$ and $\bigcap_{i\in W} A_i = \varnothing$.

If, for some $i \in W$, the recursive call $\texttt{HellyBranch}(N' \setminus \{i\}, r-1)$ returns a lottery $\mathbf{x}$, then by the induction hypothesis there exists $V' \subseteq N' \setminus \{i\}$ with $|V'| \le r-1$ such that $\mathbf{x} \in \bigcap_{t\in N'\setminus(V'\cup\{i\})} A_t$. Thus, the current call returns a valid lottery with deleted set $V := V' \cup \{i\}$, and $|V| \le r$.

Conversely, suppose there exists $V^* \subseteq N'$ with $|V^*| \le r$ such that $\bigcap_{i\in N'\setminus V^*} A_i \neq \varnothing$. Since $\bigcap_{i\in W} A_i = \varnothing$, we must have $V^* \cap W \neq \varnothing$; otherwise $W \subseteq N' \setminus V^*$, which would imply $\bigcap_{i\in N'\setminus V^*} A_i \subseteq \bigcap_{i\in W} A_i = \varnothing$, a contradiction. Fix any $i^* \in V^* \cap W$. Then $V^* \setminus \{i^*\} \subseteq N' \setminus \{i^*\}$ and $|V^* \setminus \{i^*\}| \le r-1$, and deleting $V^* \setminus \{i^*\}$ from $N' \setminus \{i^*\}$ leaves exactly $N' \setminus V^*$, whose acceptable sets have a common point. By the induction hypothesis, the recursive call on $(N' \setminus \{i^*\}, r-1)$ returns a lottery, and since the algorithm tries every $i \in W$, the current call also returns a lottery. This proves the recursive claim. Applying it to the top-level call (N, v), we obtain the desired correctness.

If there exists a lottery $\mathbf{x} \in \Delta(S)$ with $\mathrm{viol}(\mathbf{x}) \le v$, define $V := \{i \in N : \langle u_i, \mathbf{x}\rangle < \tau_i\}$. By the definition of A_i, we have $|V| = \mathrm{viol}(\mathbf{x}) \le v$ and $\mathbf{x} \in \bigcap_{i\in N\setminus V} A_i$. Thus, by the recursive claim, $\texttt{HellyBranch}(N, v)$ does not return NO.

Conversely, if $\texttt{HellyBranch}(N, v)$ returns a lottery $\mathbf{x}$, then by the recursive claim there exists $V \subseteq N$ with $|V| \le v$ such that $\mathbf{x} \in \bigcap_{i\in N\setminus V} A_i$. Therefore, every violated agent lies in V, and so $\mathrm{viol}(\mathbf{x}) \le |V| \le v$. Thus Algorithm 3 returns NO if and only if there is no lottery $\mathbf{x} \in \Delta(S)$ with $\mathrm{viol}(\mathbf{x}) \le v$, and whenever it returns a lottery, that lottery has at most v violations.

For the running time, each internal node has at most m children and the recursion depth is at most v, so the recursion tree has at most $1 + m + \cdots + m^v = \mathcal{O}(m^v)$ nodes. At each node, Line 1 solves one LP feasibility problem over $\Delta(S)$; if the current family is infeasible and $r > 0$, Lemma E.3 performs at most $|N'| \le n$ additional LP feasibility checks. Each such LP has m variables and at most $m + |N'| + 1 \le m + n + 1$ linear constraints. Thus the algorithm performs $\mathcal{O}(m^v n)$ LP feasibility checks, each of which takes polynomial time, so the problem is fixed-parameter tractable when parameterized by (v, m). □

E.3 Proof of Theorem 4.6

Proof. We give a parameterized reduction from HITTING SET. Let $(q, \mathcal{F}, v)$ be an instance, where $\mathcal{F} = \{F_1, \ldots, F_t\}$ is a family of subsets of $[q]$, and the question is whether there exists a set $C \subseteq [q]$ with $|C| \le v$ such that $C \cap F_\ell \neq \varnothing$ for every $\ell \in \{1, \ldots, t\}$.

We may assume that $q \ge 1$, that every F_ℓ is nonempty, and that $v \le q$: if some F_ℓ is empty, the instance is a no-instance, and otherwise, if $q = 0$ or $v > q$, it is a yes-instance (take $C = [q]$); in these cases we output a fixed no-instance or yes-instance of MIN-VIOL with parameter 0.

From $(q, \mathcal{F}, v)$, construct an instance of MIN-VIOL as follows. Let the set of alternatives be $S = \{s_1, \ldots, s_q, s_{q+1}\}$, where s_{q+1} is an extra alternative.

For each $j \in [q]$, create an element agent a_j with threshold $\tau_{a_j} = 1/q$ and utility vector $u_{a_j}(s_j) = 1$, $u_{a_j}(s_\ell) = 0$ for every $\ell \neq j$. For any lottery $\mathbf{x} = (x_1, \ldots, x_{q+1}) \in \Delta(S)$, we have $U_{a_j}(\mathbf{x}) = x_j$. Thus, agent a_j accepts $\mathbf{x}$ if and only if $x_j \ge 1/q$.

Next, for each set $F_\ell \in \mathcal{F}$ and each copy index $r \in \{1, \ldots, v+1\}$, create a set agent $b_{\ell,r}$ with threshold $\tau_{b_{\ell,r}} = 1 - (|F_\ell| - 1)/q$ and utilities $u_{b_{\ell,r}}(s_j) = 0$ for $j \in F_\ell$ and $u_{b_{\ell,r}}(s_j) = 1$ for $j \in [q+1] \setminus F_\ell$. In particular, $u_{b_{\ell,r}}(s_{q+1}) = 1$. Therefore, for every $\mathbf{x} \in \Delta(S)$, $U_{b_{\ell,r}}(\mathbf{x}) = \sum_{j\notin F_\ell} x_j = 1 - \sum_{j\in F_\ell} x_j$. Thus $b_{\ell,r}$ accepts $\mathbf{x}$ if and only if $1 - \sum_{j\in F_\ell} x_j \ge 1 - (|F_\ell| - 1)/q$, that is, if and only if $\sum_{j\in F_\ell} x_j \le (|F_\ell| - 1)/q$.

All utilities in the construction are binary, and all pairs are normalized: every utility vector takes both values 0 and 1, and every threshold lies in $(0, 1]$. When the threshold is 1, the utility vector is precisely the indicator vector specified in case (ii) of Lemma B.1. Moreover, the constructed profile satisfies the nondegeneracy assumption: each element agent a_j accepts $\mathbf{e}_j$ and rejects $\mathbf{e}_{q+1}$; and, since $F_\ell \neq \varnothing$ implies $\tau_{b_{\ell,r}} = (q - |F_\ell| + 1)/q \in (0, 1]$, each set agent $b_{\ell,r}$ accepts $\mathbf{e}_{q+1}$ and rejects every $\mathbf{e}_j$ with $j \in F_\ell$.

We now prove correctness. Assume first that the HITTING SET instance is a yes-instance, and let $C \subseteq [q]$ be a hitting set with $|C| \le v$. Define a lottery $\mathbf{x}^C \in \Delta(S)$ by $x^C_j = 0$ for $j \in C$, $x^C_j = 1/q$ for $j \in [q] \setminus C$, and $x^C_{q+1} = 1 - \sum_{j=1}^{q} x^C_j = |C|/q$. For every $j \in [q]$, the element agent a_j is violated if and only if $j \in C$, since $U_{a_j}(\mathbf{x}^C) = x^C_j$ and $\tau_{a_j} = 1/q$. Thus, exactly $|C|$ element agents are violated.

Now fix any $\ell \in \{1, \dots, t\}$. Since C is a hitting set, $C \cap F_\ell \neq \varnothing$, so at most $|F_\ell| - 1$ elements of F_ℓ belong to $[q] \setminus C$. Since $x^C_j = 1/q$ for $j \in [q] \setminus C$ and $x^C_j = 0$ for $j \in C$, it follows that $\sum_{j \in F_\ell} x^C_j \le (|F_\ell| - 1)/q$. Hence every copy $b_{\ell,r}$ accepts $\mathbf{x}^C$. Thus no set agent is violated. Therefore, $\mathrm{viol}(\mathbf{x}^C) = |C| \le v$, so the constructed MIN-VIOL instance is a yes-instance.

Conversely, suppose there exists a lottery $\mathbf{x} \in \Delta(S)$ such that $\mathrm{viol}(\mathbf{x}) \le v$. We first show that $\mathbf{x}$ violates no set agent. Indeed, if $\mathbf{x}$ violated some $b_{\ell,r}$, then it would violate all $v + 1$ identical copies $b_{\ell,1}, \dots, b_{\ell,v+1}$, which would imply $\mathrm{viol}(\mathbf{x}) \ge v + 1$, contradicting $\mathrm{viol}(\mathbf{x}) \le v$.

Thus, every set agent accepts $\mathbf{x}$, and so for every $\ell \in \{1, \dots, t\}$ we have $\sum_{j \in F_\ell} x_j \le (|F_\ell| - 1)/q$. Define $C(\mathbf{x}) := \{j \in [q] : x_j < 1/q\}$. We claim that $C(\mathbf{x})$ is a hitting set. Suppose not. Then there exists some $\ell \in \{1, \dots, t\}$ such that $C(\mathbf{x}) \cap F_\ell = \varnothing$. Thus, $x_j \ge 1/q$ for every $j \in F_\ell$, and therefore $\sum_{j \in F_\ell} x_j \ge |F_\ell|/q > (|F_\ell| - 1)/q$, contradicting the acceptance of the set agents associated with F_ℓ. Thus $C(\mathbf{x})$ intersects every set F_ℓ, so it is a hitting set.

Finally, an element agent a_j is violated if and only if $x_j < 1/q$, that is, if and only if $j \in C(\mathbf{x})$. Since no set agent is violated, it follows that $|C(\mathbf{x})| = \mathrm{viol}(\mathbf{x}) \le v$. So $C(\mathbf{x})$ is a hitting set of size at most v.

We have shown that in the case $q \ge 1$, every $F_\ell \neq \varnothing$, and $v \le q$, the instance $(q, \mathcal{F}, v)$ is a yes-instance of HITTING SET if and only if the constructed MIN-VIOL instance is a yes-instance. In this branch the construction uses $q + 1$ alternatives and $q + t(v + 1)$ agents, and since $v \le q$, its size is polynomial in the size of $(q, \mathcal{F}, v)$. The output parameter is v, or 0 in the trivial cases above. Therefore this is a parameterized reduction from HITTING SET parameterized by v to the decision version of MIN-VIOL. Since HITTING SET is W[2]-hard parameterized by v, the latter problem is also W[2]-hard when parameterized by v alone.

The same hardness immediately carries over to the search version: any algorithm that returns a lottery $\mathbf{x}$ with $\mathrm{viol}(\mathbf{x}) \le v$ whenever one exists also decides the decision problem. □

F Omitted Proofs from Section 5

Throughout this appendix, $i_1, \dots, i_k$ are sampled independently from $\mathrm{Unif}(N)$. For a lottery $\mathbf{x}$, let $R_{\mathbf{x}} := \{i \in N : \mathbf{x} \notin A_i\}$ be the set of agents who reject $\mathbf{x}$, let $r(\mathbf{x}) := |R_{\mathbf{x}}|/n = \mathrm{viol}(\mathbf{x})/n$, and let $\widehat{r}(\mathbf{x}) := \frac{1}{k} \sum_{t=1}^{k} \mathbf{1}\{i_t \in R_{\mathbf{x}}\}$ be the fraction of sampled agents who reject $\mathbf{x}$, counted with multiplicity. The lotteries $\widehat{\mathbf{x}}$ in Theorems 5.1 and 5.2 are exactly the minimizers of $\widehat{r}$. Both r and $\widehat{r}$ take finitely many values, so all minima below are attained.

F.1 Proof of Theorem 5.1

Since $\sum_{j=1}^{m} x_j = 1$ for every lottery, we have $i \in R_{\mathbf{x}}$ if and only if $\langle u_i, \mathbf{x} \rangle < \tau_i$, that is, if and only if $\sum_{j=1}^{m} (u_i(s_j) - \tau_i) x_j < 0$. If we identify agent i with the point $(u_i(s_1) - \tau_i, \dots, u_i(s_m) - \tau_i) \in \mathbb{R}^m$, then $R_{\mathbf{x}}$ is the set of agents whose points lie in the open halfspace $\{\mathbf{y} \in \mathbb{R}^m : \langle \mathbf{y}, \mathbf{x} \rangle < 0\}$. Hence the family $\{R_{\mathbf{x}} : \mathbf{x} \in \Delta(S)\}$ is induced on N by a subfamily of the open halfspaces of $\mathbb{R}^m$ whose boundary passes through the origin, and its VC dimension is at most m. Since $r(\mathbf{x}) = \Pr_{i \sim \mathrm{Unif}(N)}[i \in R_{\mathbf{x}}]$, the VC inequality [Vapnik, 1998, Anthony and Bartlett, 1999] together with Sauer's lemma (using $k \ge m$) gives, with probability at least $1 - \delta$, $\sup_{\mathbf{x} \in \Delta(S)} |\widehat{r}(\mathbf{x}) - r(\mathbf{x})| \le 2\sqrt{2(m \log(2ek/m) + \log(4/\delta))/k} \le \alpha$, where $\alpha := 4\sqrt{(m \log(2ek/m) + \log(8/\delta))/k}$. On this event, if $\mathbf{x}^*$ minimizes r and $\widehat{\mathbf{x}}$ minimizes $\widehat{r}$, then $r(\widehat{\mathbf{x}}) \le \widehat{r}(\widehat{\mathbf{x}}) + \alpha \le \widehat{r}(\mathbf{x}^*) + \alpha \le r(\mathbf{x}^*) + 2\alpha$. Multiplying by n gives $\mathrm{viol}(\widehat{\mathbf{x}}) \le \mathrm{OPT} + 2\alpha n$, which is the claim.

F.2 Proof of Theorem 5.2

The proof follows that of Theorem 5.1, with the VC inequality replaced by a bound on the relative error, which is much smaller for lotteries rejected by few agents. By the proof of Theorem 5.1, the family $\{R_{\mathbf{x}} : \mathbf{x} \in \Delta(S)\}$ has VC dimension at most m. Let $\beta := \zeta/20$. By the relative-error bound of Li et al. [2001, Theorem 5], applied with parameter ρ and accuracy β, there is an absolute constant C such that, if $k \geq C(m\log(1/\rho) + \log(1/\delta))/(\zeta^2\rho)$, then with probability at least $1-\delta$, $|\widehat{r}(\mathbf{x}) - r(\mathbf{x})| \leq \beta(\widehat{r}(\mathbf{x}) + r(\mathbf{x}) + \rho)$ for all $\mathbf{x} \in \Delta(S)$. Fix this event. Rearranging the two directions of this inequality gives $(1-\beta)\, r(\mathbf{x}) \leq (1+\beta)\, \widehat{r}(\mathbf{x}) + \beta\rho$ and $(1-\beta)\, \widehat{r}(\mathbf{x}) \leq (1+\beta)\, r(\mathbf{x}) + \beta\rho$ for every lottery $\mathbf{x}$. Let $\mathbf{x}^*$ minimize r, so that $r(\mathbf{x}^*) = \mathrm{OPT}/n$. Since $\widehat{\mathbf{x}}$ minimizes $\widehat{r}$,

$$(1-\beta)\, r(\widehat{\mathbf{x}}) \leq (1+\beta)\, \widehat{r}(\widehat{\mathbf{x}}) + \beta\rho \leq (1+\beta)\, \widehat{r}(\mathbf{x}^*) + \beta\rho \leq \frac{(1+\beta)^2}{1-\beta}\, r(\mathbf{x}^*) + \frac{1+\beta}{1-\beta}\, \beta\rho + \beta\rho,$$

and dividing by $1-\beta$ gives $r(\widehat{\mathbf{x}}) \leq ((1+\beta)/(1-\beta))^2\, r(\mathbf{x}^*) + 2\beta\rho/(1-\beta)^2$. For $0 < \beta \leq 1/20$, we have $((1+\beta)/(1-\beta))^2 \leq 1 + 10\beta$ (the left-hand side is convex in β, equals 1 at $\beta = 0$, and is below 1.23 at $\beta = 1/20$) and, since $(1-\beta)^2 \geq 9/10$, $2\beta/(1-\beta)^2 \leq 10\beta$. Since $10\beta = \zeta/2$, multiplying by n gives $\mathrm{viol}(\widehat{\mathbf{x}}) \leq (1+\zeta/2)\mathrm{OPT} + (\zeta/2)\rho n$. If $\mathrm{OPT} \geq \rho n$, then $(\zeta/2)\rho n \leq (\zeta/2)\mathrm{OPT}$, so $\mathrm{viol}(\widehat{\mathbf{x}}) \leq (1+\zeta)\mathrm{OPT}$.

F.3 Proof of Theorem 5.3

We show that the LP penalty is an upper bound on the violation indicator (Lemma F.1), that its sample average is uniformly close to its population average (Lemma F.2), and that its population average for any lottery $\mathbf{x}$ is at most $((1+1/\eta)\mathrm{viol}(\mathbf{x}) + |B_\eta(\mathbf{x})|)/n$. For an agent i and a lottery $\mathbf{x}$, let $\ell(i,\mathbf{x}) := \max\{0, (\tau_i + \eta - U_i(\mathbf{x}))/\eta\}$ be the penalty of Section 5.1, and let $\widehat{L}(\mathbf{x}) := \frac{1}{k}\sum_{t=1}^{k} \ell(i_t, \mathbf{x})$ and $L(\mathbf{x}) := \frac{1}{n}\sum_{i \in N} \ell(i,\mathbf{x})$ be its sample and population averages.

Lemma F.1. *For every agent i and lottery $\mathbf{x}$, $\mathbf{1}\{U_i(\mathbf{x}) < \tau_i\} \leq \ell(i,\mathbf{x})$. Moreover, the optimal value of* (1) *is $\min_{\mathbf{x}\in\Delta(S)} \widehat{L}(\mathbf{x})$, and the $\mathbf{x}$-part of every optimal solution of* (1) *minimizes $\widehat{L}$.*

Proof. If $U_i(\mathbf{x}) \geq \tau_i$, the indicator is $0 \leq \ell(i,\mathbf{x})$. If $U_i(\mathbf{x}) < \tau_i$, then $\tau_i + \eta - U_i(\mathbf{x}) > \eta$, so $\ell(i,\mathbf{x}) > 1$. For a fixed $\mathbf{x}$, the smallest feasible value of z_t in (1) is $\ell(i_t,\mathbf{x})$, so the smallest objective value with this $\mathbf{x}$ is $\widehat{L}(\mathbf{x})$. Hence the optimal value of (1) is $\min_{\mathbf{x}} \widehat{L}(\mathbf{x})$, and if $(\mathbf{x}, z_1, \ldots, z_k)$ is optimal, then $\widehat{L}(\mathbf{x}) \leq \frac{1}{k}\sum_t z_t = \min_{\mathbf{x}'} \widehat{L}(\mathbf{x}')$. □

Lemma F.2. *There is an absolute constant $C > 0$ such that, for every integer $k \geq 1$, $\delta \in (0,1)$, and $\eta \in (0,1]$, with probability at least $1-\delta$, $\sup_{\mathbf{x}\in\Delta(S)} |\widehat{L}(\mathbf{x}) - L(\mathbf{x})| \leq (C/\eta)\sqrt{(\log m + \log(1/\delta))/k}$.*

Proof. We use the empirical Rademacher complexity of a class $\mathcal{H}$ of functions on N, namely $\widehat{\mathfrak{R}}(\mathcal{H}) := \mathbb{E}_\sigma[\sup_{g\in\mathcal{H}} \frac{1}{k}\sum_{t=1}^{k} \sigma_t\, g(i_t)]$, where $\sigma_1, \ldots, \sigma_k$ are independent and uniform on $\{-1, 1\}$, together with the standard facts about it in Shalev-Shwartz and Ben-David [2014, Chapter 26].

For each lottery $\mathbf{x}$, define $f_{\mathbf{x}}(i) := U_i(\mathbf{x}) - \tau_i$, and for each alternative s_j, define $h_j(i) := u_i(s_j) - \tau_i$. Since $x_j \geq 0$ and $\sum_{j=1}^{m} x_j = 1$, we have $f_{\mathbf{x}} = \sum_{j=1}^{m} x_j h_j$, so the class $\mathcal{F} := \{f_{\mathbf{x}} : \mathbf{x} \in \Delta(S)\}$ is the convex hull of $h_1, \ldots, h_m$. Taking the convex hull does not change the Rademacher complexity, and since each h_j takes values in $[-1, 1]$, Massart's lemma gives $\widehat{\mathfrak{R}}(\mathcal{F}) = \widehat{\mathfrak{R}}(\{h_1, \ldots, h_m\}) \leq 2\sqrt{(2\log m)/k}$.

Let $\psi(z) := \max\{0, 1 - z/\eta\} - 1$, so that $\ell(i,\mathbf{x}) = 1 + \psi(f_{\mathbf{x}}(i))$, and let $\mathcal{G} := \{\psi \circ f_{\mathbf{x}} : \mathbf{x} \in \Delta(S)\}$. Since ψ is $(1/\eta)$-Lipschitz, the contraction lemma gives $\widehat{\mathfrak{R}}(\mathcal{G}) \leq \widehat{\mathfrak{R}}(\mathcal{F})/\eta$. Moreover, $f_{\mathbf{x}}(i) \in [-1, 1]$ because $U_i(\mathbf{x}) \in [0,1]$ and $\tau_i \in (0,1]$, so every function in $\mathcal{G}$ takes values in $[-1, 1/\eta] \subseteq [-1/\eta, 1/\eta]$. The uniform convergence bound in terms of $\widehat{\mathfrak{R}}$ [Shalev-Shwartz and Ben-David, 2014, Theorem 26.5], applied to $\mathcal{G}$ and to $-\mathcal{G}$ (which has the same Rademacher complexity, since σ is symmetric), each with $\delta/2$ in place of δ, shows that with probability at least $1-\delta$,

$$\sup_{g\in\mathcal{G}} \left| \frac{1}{k}\sum_{t=1}^{k} g(i_t) - \frac{1}{n}\sum_{i\in N} g(i) \right| \leq 2\widehat{\mathfrak{R}}(\mathcal{G}) + \frac{4}{\eta}\sqrt{\frac{2\log(8/\delta)}{k}} \leq \frac{4}{\eta}\sqrt{\frac{2\log m}{k}} + \frac{4}{\eta}\sqrt{\frac{2\log(8/\delta)}{k}}.$$

Since the constant 1 in $\ell(i, \mathbf{x}) = 1 + \psi(f_{\mathbf{x}}(i))$ cancels, the left-hand side equals $\sup_{\mathbf{x} \in \Delta(S)} |\widehat{L}(\mathbf{x}) - L(\mathbf{x})|$. Finally, $\log(8/\delta) \le 3 \log m + \log(1/\delta)$ because $m \ge 2$, which gives the lemma. □

Let $\kappa := (C/\eta)\sqrt{(\log m + \log(1/\delta))/k}$ be the bound of Lemma F.2, and fix the event of that lemma. We use three bounds. First, by Lemma F.1, $\mathrm{viol}(\widehat{\mathbf{x}}_\eta) \le nL(\widehat{\mathbf{x}}_\eta)$. Second, since $\widehat{\mathbf{x}}_\eta$ minimizes $\widehat{L}$ (Lemma F.1), every lottery $\mathbf{x}$ satisfies $L(\widehat{\mathbf{x}}_\eta) \le \widehat{L}(\widehat{\mathbf{x}}_\eta) + \kappa \le \widehat{L}(\mathbf{x}) + \kappa \le L(\mathbf{x}) + 2\kappa$. Third, we bound $nL(\mathbf{x})$ for a fixed lottery $\mathbf{x}$. If $U_i(\mathbf{x}) \ge \tau_i + \eta$, then $\ell(i, \mathbf{x}) = 0$. If $i \in B_\eta(\mathbf{x})$, then $0 < \ell(i, \mathbf{x}) \le 1$. If $U_i(\mathbf{x}) < \tau_i$, then $\ell(i, \mathbf{x}) = (\tau_i + \eta - U_i(\mathbf{x}))/\eta \le (1 + \eta)/\eta = 1 + 1/\eta$, because $\tau_i \le 1$ and $U_i(\mathbf{x}) \ge 0$. Therefore $nL(\mathbf{x}) \le (1 + 1/\eta)\mathrm{viol}(\mathbf{x}) + |B_\eta(\mathbf{x})|$. Combining the three bounds gives $\mathrm{viol}(\widehat{\mathbf{x}}_\eta) \le nL(\mathbf{x}) + 2\kappa n \le (1 + 1/\eta)\mathrm{viol}(\mathbf{x}) + |B_\eta(\mathbf{x})| + 2\kappa n$ for every lottery $\mathbf{x}$, which is the claim of Theorem 5.3 with constant $2C$. If the LP is run on all n agents, each once, then $\widehat{L} = L$, and the same argument with $\kappa = 0$ gives the full-information bound stated in Section 5.1.

G Converting Sample Sizes into Query Budgets

Suppose each agent can be recovered with at most $T_\varepsilon = \mathcal{O}(m \log(1/\varepsilon))$ queries, as under the finite-grid convention. With a budget of Q queries, sample $k := \lfloor Q/T_\varepsilon \rfloor$ agents independently from $\mathrm{Unif}(N)$ and recover only the distinct sampled agents. This uses at most $kT_\varepsilon \le Q$ queries, and the sampled agents, with their multiplicities, then define the problem on the sample exactly.

Proposition G.1. *Assume the finite-grid convention and that m is a constant. Fix $\delta \in (0, 1)$, let $k := \lfloor Q/T_\varepsilon \rfloor \ge m$, sample and recover agents as above, and minimize the number of rejecting sampled agents exactly with the algorithm of Theorem 4.1. This uses at most Q queries, the optimization takes time polynomial in k, and with probability at least $1 - \delta$ the returned lottery satisfies the bound of Theorem 5.1. If, in addition, δ, ρ, ζ, and k satisfy the conditions of Theorem 5.2 and $\mathrm{OPT} \ge \rho n$, then with probability at least $1 - \delta$ the returned lottery has at most $(1 + \zeta)\mathrm{OPT}$ violations.*

Proof. The query count was shown above. The sampled agents, counted with multiplicity, form a profile of k agents, and Theorem 4.1 minimizes the number of violations on it exactly in time polynomial in k for constant m; for $m = 2$, Proposition E.1 takes time $\mathcal{O}(k \log k)$. The guarantees then follow from Theorems 5.1 and 5.2. □

Proposition G.2. *Assume the finite-grid convention. Fix $\delta \in (0, 1)$ and $\eta \in (0, 1]$, let $k := \lfloor Q/T_\varepsilon \rfloor \ge 1$, sample and recover agents as above, and solve the linear program* (1)*. This uses at most Q queries, the optimization takes time polynomial in k and m, and with probability at least $1 - \delta$ the returned lottery satisfies the bound of Theorem 5.3.*

Proof. The query count was shown above. Once the sampled agents are recovered, (1) is a linear program with $m + k$ variables and $\mathcal{O}(m + k)$ constraints, which can be solved in polynomial time. The guarantee follows from Theorem 5.3. □

H Omitted Proofs from Section 6

H.1 Proof of Theorem 6.1

The idea is to choose utilities that reproduce every answer the algorithm received, while making every unqueried agent accept only a pure lottery other than $\widehat{\mathbf{x}}$. Consider the run of $\mathcal{A}$ in which every query $(i, \mathbf{x})$ is answered "accept" if and only if $x_3 = \cdots = x_m = 0$, with queried set M and output $\widehat{\mathbf{x}}$. For each queried agent $i \in M$, set $u_i = (1, 1, 0, \ldots, 0)$ and $\tau_i = 1$. Then $A_i = \{\mathbf{x} \in \Delta(S) : x_1 + x_2 \ge 1\} = \{\mathbf{x} \in \Delta(S) : x_3 = \cdots = x_m = 0\}$, so these agents give exactly the answers of the run above. For each unqueried agent $i \in N \setminus M$, set $\tau_i = 1$ and $u_i = \mathbf{z}$, where $\mathbf{z} := \mathbf{e}_1$ if $\widehat{\mathbf{x}} \ne \mathbf{e}_1$ and $\mathbf{z} := \mathbf{e}_2$ if $\widehat{\mathbf{x}} = \mathbf{e}_1$. Let $\mathcal{I} := ((u_i, \tau_i))_{i \in N}$. Every unqueried agent accepts exactly one lottery, the pure lottery $\mathbf{z}$, so it rejects $\widehat{\mathbf{x}}$. All agents are nondegenerate: queried agents accept exactly the lotteries $(\lambda, 1 - \lambda, 0, \ldots, 0)$ with $\lambda \in [0, 1]$ and reject $\mathbf{e}_3$, which exists because $m \ge 3$, while each unqueried agent accepts only $\mathbf{z}$. All pairs are indicator vectors of proper subsets of alternatives with threshold 1, so they are normalized.

We claim that running $\mathcal{A}$ on $\mathcal{I}$ produces the same query history as the original run. We show by induction on t that the first t query–answer pairs agree, for every t up to the number of queries in the original run. The case $t = 0$ is trivial. If the first t pairs agree, then, because $\mathcal{A}$ is deterministic, its $(t+1)$-st query is the same in both runs, so the queried agent belongs to M. For agents in M, the answers under $\mathcal{I}$ agree with the answer rule of the original run, so the $(t+1)$-st answer is also the same. Hence the entire query history is the same, and since $\mathcal{A}$ is deterministic, it returns $\widehat{\mathbf{x}}$ on $\mathcal{I}$.

Every agent in $N \setminus M$ rejects $\widehat{\mathbf{x}}$, so $\mathrm{viol}_{\mathcal{I}}(\widehat{\mathbf{x}}) \geq n - |M|$. Every queried agent accepts $\mathbf{z} \in \{\mathbf{e}_1, \mathbf{e}_2\}$, and every unqueried agent accepts $\mathbf{z}$ by construction, so $\mathrm{viol}_{\mathcal{I}}(\mathbf{z}) = 0$ and $\mathrm{OPT}(\mathcal{I}) = 0$.

H.2 Proof of Theorem 6.2

Upper bound. Fix $M \subseteq N$ with $|M| = K$. Consider the deterministic algorithm that queries only agents in M, recovers each of them, and returns a lottery $\mathbf{x}_M$ minimizing the number of violated agents in M, chosen by a fixed tie-breaking rule. This algorithm belongs to $\mathcal{C}_K$. For any profile $\mathcal{I}$, let $\mathbf{x}^*$ be optimal for $\mathcal{I}$. Then $\mathbf{x}_M$ violates at most as many agents of M as $\mathbf{x}^*$ does, and at most $n - K$ agents outside M, so $\mathrm{viol}_{\mathcal{I}}(\mathbf{x}_M) \leq \mathrm{viol}_{\mathcal{I}}(\mathbf{x}^*) + (n - K) = \mathrm{OPT}(\mathcal{I}) + (n - K)$. Hence $\inf_{\mathcal{A} \in \mathcal{C}_K} \sup_{\mathcal{I}} (\mathrm{viol}_{\mathcal{I}}(\mathcal{A}(\mathcal{I})) - \mathrm{OPT}(\mathcal{I})) \leq n - K$.

Lower bound. Fix any $\mathcal{A} \in \mathcal{C}_K$ and run it on the profile in which every agent has utility vector $(1, 1, 0, \ldots, 0)$ and threshold 1; its answers are exactly those of the answer rule in Theorem 6.1. Let M be the set of agents queried and $\widehat{\mathbf{x}}$ the output. This profile satisfies the model assumptions, so $|M| \leq K$. By Theorem 6.1, there is a profile $\mathcal{I}$ on which $\mathcal{A}$ returns $\mathcal{A}(\mathcal{I}) = \widehat{\mathbf{x}}$, every agent in $N \setminus M$ rejects $\widehat{\mathbf{x}}$, and $\mathrm{OPT}(\mathcal{I}) = 0$. Therefore $\mathrm{viol}_{\mathcal{I}}(\mathcal{A}(\mathcal{I})) - \mathrm{OPT}(\mathcal{I}) = \mathrm{viol}_{\mathcal{I}}(\widehat{\mathbf{x}}) \geq n - |M| \geq n - K$. The profile $\mathcal{I}$ satisfies the model assumptions: it is nondegenerate by the proof of Theorem 6.1, and since its utilities are binary and its thresholds equal 1, it satisfies the finite-grid convention for every ε. Hence $\sup_{\mathcal{I}} (\mathrm{viol}_{\mathcal{I}}(\mathcal{A}(\mathcal{I})) - \mathrm{OPT}(\mathcal{I})) \geq n - K$ for every $\mathcal{A} \in \mathcal{C}_K$, which proves the claim.

I Bounds After a Partial Query History

This appendix studies what can be said after an arbitrary query history, before all agents have been recovered: we bound the unknown optimum, and the worst-case loss of a chosen lottery, using only the answers observed so far.

I.1 Bounds from observed queries

Throughout this subsection, no finite-precision assumption is needed.

A partial query history is the set of accept/reject answers observed so far. We write it as a finite set $h \subseteq N \times \Delta(S) \times \{0, 1\}$, where $(i, \mathbf{x}, b) \in h$ means that querying agent i on lottery $\mathbf{x}$ returned answer b. For each agent $i \in N$, let $h_i := \{(\mathbf{x}, b) : (i, \mathbf{x}, b) \in h\}$ be the answers observed for agent i. We say that h' *extends* h if $h \subseteq h'$, and that h is *consistent* if there exists at least one profile satisfying our model whose answers reproduce every triple in h.

For each agent i, let $\Theta_i(h_i)$ be the set of normalized pairs (u, τ), in the sense of Section 2, that agree with all observed answers, i.e., $\mathbf{1}\{\langle u, \mathbf{x} \rangle \geq \tau\} = b$ for every $(\mathbf{x}, b) \in h_i$. Thus $\Theta_i(h_i)$ contains the pairs that remain possible for agent i, and h is consistent if and only if $\Theta_i(h_i) \neq \varnothing$ for every $i \in N$.

A profile $\mathcal{I} = ((u_i, \tau_i))_{i \in N}$ is *consistent with* h if $(u_i, \tau_i) \in \Theta_i(h_i)$ for every $i \in N$; we use $\mathrm{viol}_{\mathcal{I}}$ and $\mathrm{OPT}(\mathcal{I})$ as in Section 6. The answers observed so far determine two sets of lotteries for each agent: $\underline{A}_i(h_i) := \{\mathbf{x} \in \Delta(S) : \langle u, \mathbf{x} \rangle \geq \tau \text{ for every } (u, \tau) \in \Theta_i(h_i)\}$ and $\overline{A}_i(h_i) := \{\mathbf{x} \in \Delta(S) : \langle u, \mathbf{x} \rangle \geq \tau \text{ for some } (u, \tau) \in \Theta_i(h_i)\}$. The first set contains the lotteries that agent i is guaranteed to accept given the observed answers, and the second contains those that agent i accepts under at least one consistent profile. If h is consistent, then $\Theta_i(h_i) \neq \varnothing$, so $\underline{A}_i(h_i) \subseteq \overline{A}_i(h_i)$, and if $\Theta_i(h_i) = \{(u_i, \tau_i)\}$, then both sets equal A_i.

For any lottery $\mathbf{x} \in \Delta(S)$, define the lower and upper violation counts consistent with the observed answers as $\underline{\mathrm{viol}}_h(\mathbf{x}) := |\{i \in N : \mathbf{x} \notin \overline{A}_i(h_i)\}|$ and $\overline{\mathrm{viol}}_h(\mathbf{x}) := |\{i \in N : \mathbf{x} \notin \underline{A}_i(h_i)\}|$. The lower count $\underline{\mathrm{viol}}_h(\mathbf{x})$ counts agents who must reject $\mathbf{x}$ under every profile still consistent with the observed answers, while the upper count $\overline{\mathrm{viol}}_h(\mathbf{x})$ counts agents who could reject $\mathbf{x}$ under at least one consistent profile.

Theorem I.1. *For every consistent query history h, every lottery $\mathbf{x} \in \Delta(S)$, and every profile $\mathcal{I}$ consistent with h, $\underline{\mathrm{viol}}_h(\mathbf{x}) \le \mathrm{viol}_{\mathcal{I}}(\mathbf{x}) \le \overline{\mathrm{viol}}_h(\mathbf{x})$. Consequently, if $L(h) := \min_{\mathbf{x}\in\Delta(S)} \underline{\mathrm{viol}}_h(\mathbf{x})$ and $H(h) := \min_{\mathbf{x}\in\Delta(S)} \overline{\mathrm{viol}}_h(\mathbf{x})$, then every profile $\mathcal{I}$ consistent with h satisfies $L(h) \le \mathrm{OPT}(\mathcal{I}) \le H(h)$. Moreover, if $\widehat{\mathbf{x}}_h \in \arg\min_{\mathbf{x}\in\Delta(S)} \overline{\mathrm{viol}}_h(\mathbf{x})$, then $\mathrm{viol}_{\mathcal{I}}(\widehat{\mathbf{x}}_h) \le \mathrm{OPT}(\mathcal{I}) + (H(h) - L(h))$. In particular, if $H(h) = L(h)$, then $\widehat{\mathbf{x}}_h$ is optimal for every profile consistent with h.*

Proof. Fix $\mathbf{x} \in \Delta(S)$ and a profile $\mathcal{I}$ consistent with h. For each agent i, we have $(u_i, \tau_i) \in \Theta_i(h_i)$. Therefore:

- if $\mathbf{x} \notin \overline{A}_i(h_i)$, then every pair in $\Theta_i(h_i)$ rejects $\mathbf{x}$, so agent i rejects $\mathbf{x}$ in profile $\mathcal{I}$;
- if $\mathbf{x} \in \underline{A}_i(h_i)$, then every pair in $\Theta_i(h_i)$ accepts $\mathbf{x}$, so agent i accepts $\mathbf{x}$ in profile $\mathcal{I}$.

Summing over agents gives $\underline{\mathrm{viol}}_h(\mathbf{x}) \le \mathrm{viol}_{\mathcal{I}}(\mathbf{x}) \le \overline{\mathrm{viol}}_h(\mathbf{x})$. Minimizing each side over $\mathbf{x} \in \Delta(S)$ gives $L(h) \le \mathrm{OPT}(\mathcal{I}) \le H(h)$. Finally, $\mathrm{viol}_{\mathcal{I}}(\widehat{\mathbf{x}}_h) \le \overline{\mathrm{viol}}_h(\widehat{\mathbf{x}}_h) = H(h)$, and combining this with $\mathrm{OPT}(\mathcal{I}) \ge L(h)$ gives $\mathrm{viol}_{\mathcal{I}}(\widehat{\mathbf{x}}_h) - \mathrm{OPT}(\mathcal{I}) \le H(h) - L(h)$. □

Proposition I.2. *Let h' extend h, and assume that h' is consistent. Then, for every lottery $\mathbf{x} \in \Delta(S)$, $\underline{\mathrm{viol}}_h(\mathbf{x}) \le \underline{\mathrm{viol}}_{h'}(\mathbf{x}) \le \overline{\mathrm{viol}}_{h'}(\mathbf{x}) \le \overline{\mathrm{viol}}_h(\mathbf{x})$. Consequently, $L(h) \le L(h') \le H(h') \le H(h)$. In particular, the uncertainty gap $H(h) - L(h)$ cannot increase as more queries are answered.*

Proof. If h' extends h, then $\Theta_i(h'_i) \subseteq \Theta_i(h_i)$ for every agent i. Thus, $\underline{A}_i(h_i) \subseteq \underline{A}_i(h'_i)$ and $\overline{A}_i(h'_i) \subseteq \overline{A}_i(h_i)$. Therefore, for each fixed lottery $\mathbf{x}$, the set of agents for whom $\mathbf{x}$ is not even possibly acceptable can only grow, while the set of agents for whom $\mathbf{x}$ is guaranteed acceptable can only grow. Equivalently, $\underline{\mathrm{viol}}_h(\mathbf{x})$ can only increase and $\overline{\mathrm{viol}}_h(\mathbf{x})$ can only decrease. This proves the first inequality, and minimizing over $\mathbf{x} \in \Delta(S)$ gives the second inequality. □

A useful extreme case is an agent with no observed answers. If $h_i = \varnothing$, then $\Theta_i(h_i)$ is the set of all normalized pairs. Thus, every lottery $\mathbf{x} \in \Delta(S)$ is accepted by some pair in $\Theta_i(h_i)$ and rejected by some (possibly different) pair in $\Theta_i(h_i)$: to accept $\mathbf{x}$, choose any j with $x_j > 0$ and take $u = \mathbf{e}_j$, $\tau = x_j/2$; to reject $\mathbf{x}$, choose any ℓ with $x_\ell < 1$ (which exists because $m \ge 2$) and take $u = \mathbf{e}_\ell$, $\tau = (x_\ell + 1)/2$. Both pairs are normalized, since their thresholds lie strictly between 0 and 1. Therefore $\overline{A}_i(h_i) = \Delta(S)$ and $\underline{A}_i(h_i) = \varnothing$. So an unseen agent contributes 0 to $\underline{\mathrm{viol}}_h(\mathbf{x})$ and 1 to $\overline{\mathrm{viol}}_h(\mathbf{x})$ for every lottery $\mathbf{x}$. In particular, if exactly K agents have nonempty histories, then $\overline{\mathrm{viol}}_h(\mathbf{x}) - \underline{\mathrm{viol}}_h(\mathbf{x}) \ge n - K$ for every $\mathbf{x} \in \Delta(S)$, and thus $H(h) - L(h) \ge n - K$. This shows, for an arbitrary partial history, why unqueried agents drive the lower bound of Theorem 6.1.

The bound $H(h) - L(h)$ for $\widehat{\mathbf{x}}_h$ can be loose. To state the exact worst-case additive gap of an arbitrary fixed lottery $\mathbf{x}$ over all profiles consistent with h, define, for $\mathbf{x}, \mathbf{z} \in \Delta(S)$,

$$D_h(\mathbf{x}, \mathbf{z}) := \sum_{i \in N} \max_{(u,\tau)\in\Theta_i(h_i)} (\mathbf{1}\{\langle u, \mathbf{x}\rangle < \tau\} - \mathbf{1}\{\langle u, \mathbf{z}\rangle < \tau\}).$$

Theorem I.3. *For every consistent query history h and every lottery $\mathbf{x} \in \Delta(S)$, the supremum of $\mathrm{viol}_{\mathcal{I}}(\mathbf{x}) - \mathrm{OPT}(\mathcal{I})$ over all profiles $\mathcal{I}$ consistent with h equals $\max_{\mathbf{z}\in\Delta(S)} D_h(\mathbf{x}, \mathbf{z})$. Consequently, the smallest worst-case additive gap that any lottery can guarantee after observing h is $\min_{\mathbf{x}\in\Delta(S)} \max_{\mathbf{z}\in\Delta(S)} D_h(\mathbf{x}, \mathbf{z})$.*

Proof. Fix $\mathbf{x} \in \Delta(S)$. For each agent i, the expression maximized in the i-th summand of D_h takes values in $\{-1, 0, 1\}$ on the nonempty set $\Theta_i(h_i)$, so its maximum is attained; and since $D_h(\mathbf{x}, \mathbf{z})$ is an integer in $\{-n, \dots, n\}$, the maximum over $\mathbf{z} \in \Delta(S)$ is attained as well. For the upper bound, let $\mathcal{I}$ be any profile consistent with h, and let $\mathbf{z}_{\mathcal{I}} \in \arg\min_{\mathbf{z}\in\Delta(S)} \mathrm{viol}_{\mathcal{I}}(\mathbf{z})$. Then $\mathrm{viol}_{\mathcal{I}}(\mathbf{x}) - \mathrm{OPT}(\mathcal{I}) = \mathrm{viol}_{\mathcal{I}}(\mathbf{x}) - \mathrm{viol}_{\mathcal{I}}(\mathbf{z}_{\mathcal{I}})$, which is the sum over $i \in N$ of $\mathbf{1}\{\langle u_i, \mathbf{x}\rangle < \tau_i\} - \mathbf{1}\{\langle u_i, \mathbf{z}_{\mathcal{I}}\rangle < \tau_i\}$. Since $(u_i, \tau_i) \in \Theta_i(h_i)$ for every i, each summand is at most the corresponding term in the definition of $D_h(\mathbf{x}, \mathbf{z}_{\mathcal{I}})$. Thus $\mathrm{viol}_{\mathcal{I}}(\mathbf{x}) - \mathrm{OPT}(\mathcal{I}) \le D_h(\mathbf{x}, \mathbf{z}_{\mathcal{I}}) \le \max_{\mathbf{z}\in\Delta(S)} D_h(\mathbf{x}, \mathbf{z})$, and taking the supremum over all profiles $\mathcal{I}$ consistent with h gives one direction of the claimed equality.

For the reverse inequality, let $\mathbf{z}^* \in \arg\max_{\mathbf{z}\in\Delta(S)} D_h(\mathbf{x}, \mathbf{z})$. For each agent i, choose $(u_i^*, \tau_i^*) \in \Theta_i(h_i)$ attaining the maximum in the i-th summand of $D_h(\mathbf{x}, \mathbf{z}^*)$. Since the consistency constraints

are separate across agents, these choices combine into a profile $\mathcal{I}^* := ((u_i^*, \tau_i^*))_{i \in N}$ that is consistent with h. By construction, $\mathrm{viol}_{\mathcal{I}^*}(\mathbf{x}) - \mathrm{viol}_{\mathcal{I}^*}(\mathbf{z}^*) = D_h(\mathbf{x}, \mathbf{z}^*)$. Since $\mathrm{OPT}(\mathcal{I}^*) \leq \mathrm{viol}_{\mathcal{I}^*}(\mathbf{z}^*)$, we get $\mathrm{viol}_{\mathcal{I}^*}(\mathbf{x}) - \mathrm{OPT}(\mathcal{I}^*) \geq D_h(\mathbf{x}, \mathbf{z}^*) = \max_{\mathbf{z} \in \Delta(S)} D_h(\mathbf{x}, \mathbf{z})$. Combining with the upper bound proves the claim. □

This suggests a query rule: choose the next query so as to minimize the largest value that $\min_{\mathbf{x}} \max_{\mathbf{z}} D_h(\mathbf{x}, \mathbf{z})$ can take after the answer. Such a rule targets the MIN-VIOL objective directly, rather than learning each agent's halfspace separately.

J Details of the Synthetic Experiments

The experiments measure how far the worst-case guarantees are from the behavior of the algorithms on random instances.

Populations. For each of two settings, we generated 20 independent populations with $n = 500$ agents and $m = 3$ alternatives. In each population, the planted lottery $\mathbf{x}$ was drawn uniformly from $\Delta(S)$. Each agent's utilities were 3 independent uniform values on $[0, 1]$, rescaled so that the smallest is 0 and the largest is 1; this gives the normalized pair of Section 2, in which margins are measured. For 425 agents (85%), τ_i is $U_i(\mathbf{x})$ minus a margin drawn uniformly from $[0.05, 0.25]$ in the separated setting and from $[0, 0.05)$ in the near-threshold setting, so these agents accept $\mathbf{x}$. For the other 75 agents, τ_i is $U_i(\mathbf{x})$ plus an amount drawn uniformly from $[0.05, 0.25]$, so they reject $\mathbf{x}$. An agent whose threshold fell outside $(0, 1)$ was drawn again. Hence $B_{0.05}(\mathbf{x}) = \varnothing$ in the separated setting, whereas in the near-threshold setting $B_{0.05}(\mathbf{x})$ consists of all 425 accepting agents, and $B_{0.02}(\mathbf{x})$ of about 169 of them on average. The planted lottery was optimal in every near-threshold population ($\mathrm{OPT} = 75$); in the separated setting, OPT was between 71 and 75.

Computing the optimum. Since $m = 3$, we computed OPT exactly by checking all candidate lotteries formed by the edges of the simplex and the agents' acceptance boundaries, i.e., the vertex enumeration of Theorem 4.1. We compared this exact solver with a mixed-integer program that has one binary variable per agent, indicating whether the agent's constraint may be violated, on all 40 populations and on all 240 samples of size $k = 120$ described below; the optimal values agreed in every case.

Sampling and exact minimization. For each population and each sample size $k \in \{30, 120, 480\}$, we drew 6 independent samples of k agents uniformly with replacement, minimized the number of rejecting sampled agents exactly, and measured the extra rejection rate $(\mathrm{viol}(\widehat{\mathbf{x}}) - \mathrm{OPT})/n$ on all 500 agents. As in Section 5, the normalized pairs of the sampled agents are taken as recovered; we did not simulate the membership queries. Among the minimizers on the sample, vertex enumeration returns the first minimizing vertex it finds; Theorem 5.1 applies to every minimizer. Table 1 reports the mean over all populations and samples and the 95th percentile. For comparison, with $m = 3$ and $k \leq 480$, the explicit bound of Theorem 5.1 exceeds n for every $\delta \in (0, 1)$, whereas the observed mean extra rejection rate is below 2% from $k = 120$ on.

The LP. Theorem 5.3 contains an unspecified absolute constant, so we did not evaluate its bound numerically. Instead, we separated the error caused by the LP from the error caused by sampling: we ran the LP with $\eta \in \{0.02, 0.05\}$ on all 500 agents and on the same samples of $k = 120$ agents used for exact minimization (Table 1). In the bound of Theorem 5.3, a larger η reduces the factor $1 + 1/\eta$ and the sampling term but enlarges $B_\eta(\mathbf{x})$, and the results follow this trade-off. In the separated setting, where $B_{0.05}(\mathbf{x}) = \varnothing$ for the planted lottery $\mathbf{x}$, increasing η from 0.02 to 0.05 lowered the mean extra rejection rate from 0.51% to 0.28% on all agents and from 1.96% to 0.53% on the samples, below the 1.67% of exact minimization on the same samples. This is possible because exact minimization returns a vertex, which typically lies on the boundary of some sampled agents' acceptable sets, whereas the LP favors lotteries that the sampled agents accept with margin. In the near-threshold setting, where $B_{0.05}(\mathbf{x})$ contains every accepting agent, increasing η raised the rate from 1.86% to 5.22% on all agents and from 3.60% to 9.20% on the samples. Thus the LP can be more accurate than exact minimization when the agents who accept a good lottery do so with a clear margin, but it loses accuracy when many of them are close to their thresholds and η is too large. The guarantee of exact minimization does not depend on margins, which is why we prefer it when it is affordable (Section 5.1).

Querying K agents. For each population and each $K \in \{10, 25, 50, 100, 200, 400\}$, we chose K distinct agents uniformly at random 6 times, minimized the number of violations among them exactly, and measured the additive gap among all 500 agents. For $K = 100$, the mean gap was 8.46 agents in the separated setting and 7.99 agents in the near-threshold setting, that is, 2.11% and 2.00% of the worst-case value $n - K = 400$. Table 3 gives all values: the mean gap was between 0.57% and 19.17% of $n - K$, and the largest gap in any single run was 202 agents, or 41% of $n - K$ (with $K = 10$). We also implemented the profile constructed in the proof of Theorem 6.1 for the algorithm that recovers K fixed agents and minimizes the number of violations among them; on it, the gap was exactly $n - K$ for every tested K.

Table 3: Mean additive gap $\mathrm{viol}(\widehat{\mathbf{x}}) - \mathrm{OPT}$ of exact minimization over K distinct agents chosen uniformly at random, as a percentage of the worst-case value $n - K$, with $n = 500$ and $m = 3$.

K	10	25	50	100	200	400
Separated	13.01	8.62	3.77	2.11	1.23	0.58
Near-threshold	19.17	7.79	3.66	2.00	0.88	0.57

Few alternatives. With $n = 1000$, $m = 500$, and $\eta = 0.15$, Theorem 4.2 gives $s_\eta = \lceil \log(2000)/0.045 \rceil = 169$. In each of 480 trials, we drew a new population: each agent's utilities were 500 independent uniform values on $[0, 1]$, rescaled as above, and the starting lottery $\mathbf{x}$ was drawn uniformly from $\Delta(S)$, so that it uses every alternative. Every agent accepts $\mathbf{x}$: τ_i is $U_i(\mathbf{x})$ minus a margin drawn uniformly from $[\eta, \eta + 0.05]$, so that $B_\eta(\mathbf{x}) = \varnothing$. In a harder setting, the margins of 20% of the agents were instead drawn uniformly from $[0, \eta)$, so that $|B_\eta(\mathbf{x})| = 200$. For $s \in \{50, 169\}$, we drew s alternatives independently from $\mathbf{x}$, took their empirical distribution as in the proof of Theorem 4.2, and counted the agents who reject it. When $B_\eta(\mathbf{x}) = \varnothing$, sampling 169 alternatives created no new rejection in any trial; in fact, no agent's expected utility dropped by more than η in any trial, whereas the proof guarantees this only with probability at least $1/2$. Sampling only 50 alternatives created no new rejection in 471 of the 480 trials and one new rejection in each of the others. In the harder setting, the mean new rejection rate was 1.19% with 169 alternatives (95th percentile 1.70%) and 2.19% with 50 alternatives (95th percentile 3.00%), whereas Theorem 4.2 allows up to $|B_\eta(\mathbf{x})|/n = 20\%$.

Implementation and compute. We used Python 3.13 with NumPy and SciPy, both distributed under the BSD 3-Clause license, and solved the LP and the mixed-integer program with the HiGHS solvers (MIT license) in SciPy. In floating point, agent i counts as accepting $\mathbf{x}$ if $U_i(\mathbf{x}) \geq \tau_i - 10^{-9}$. All random draws use fixed seeds. All instances are small ($n \leq 1000$ and $m \leq 500$): on a cloud virtual machine with 2 Intel Xeon vCPUs at 2.1 GHz and 8 GB of memory, and no GPU, all experiments together take about 2 minutes, of which about 80 seconds are spent on the mixed-integer check. Source files are available at `https://github.com/nicteh/Learning-Lotteries-Minimal-Stakeholder-Violations`.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1.1 state the paper's contributions and scope. These claims are matched by the named results in Sections 3–6 and the proof appendices.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the main modeling and technical limitations, including the linear acceptability/expected score assumption, normalization and finite-precision assumptions for query-budget and runtime statements, computational hardness, and the need for stochastic sampling assumptions to generalize beyond queried agents. Relevant discussion appears in Section 2, Section 6, Appendix B.1, and Appendix B.2.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper is primarily theoretical, and the assumptions for the main results are stated in the model and theorem statements. All proofs are provided in the appendices.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 7 and Appendix J describe the synthetic populations, sampling procedures, optimization methods, repetitions, numerical tolerance, and compute resources. The accompanying source archive includes the reproduction script and raw results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The accompanying source archive includes the reproduction script and raw results in the experiments directory. Appendix J gives the execution command, and the script documents the software environment and fixed random seeds.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 7 and Appendix J specify the population distributions, sample sizes, margin parameters, optimization methods, and evaluation measures.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report means and empirical 95th percentiles over repeated synthetic experiments. These describe the observed variation; we do not report confidence intervals or statistical significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix J reports the software, CPU resources, memory, and running time of the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is theoretical with synthetic experiments and does not involve human-subject experiments, private data, scraped datasets, model deployment, or high-risk model releases. The work is consistent with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper motivates the model using applications such as platform governance, public allocation, and fairness-related constraints, and it notes that the violation-count objective is not a complete normative account of fairness. We feel there is no need to separately discuss both potential positive and negative societal impacts in a dedicated broader impact discussion.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The experiments use NumPy, SciPy, and HiGHS, which are credited with their licenses in Appendix J. No external datasets or pretrained models are used.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The accompanying reproduction script documents the command, software environment, fixed seeds, and output format. Appendix J describes the generation and interpretation of the synthetic results; no human data are used.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.