Near-Optimal Dropout-Robust Sortiton

Maya Pal Gambhir
University of Pennsylvania

Bailey Flanigan
Massachusetts Institute of Technology

Aaron Roth
University of Pennsylvania

Abstract

Citizens' assemblies – small panels of citizens that convene to deliberate on policy issues – often face the issue of panelists dropping out at the last-minute. Without intervention, these dropouts compromise the size and representativeness of the panel, prompting the question: Without seeing the dropouts ahead of time, can we choose panelists such that *after* dropouts, the panel will be representative and appropriately-sized? We model this problem as a minimax game: the minimizer aims to choose a panel that minimizes the *loss*, i.e., the deviation of the ultimate panel from predefined representation targets. Then, an adversary defines a distribution over dropouts from which the realized dropouts are drawn. Our main contribution is an efficient loss-minimizing algorithm, which remains optimal as we vary the maximizer's power from worst-case to average case. Our algorithm iteratively plays a projected gradient descent subroutine against an efficient algorithm for computing the best-response dropout distribution. This approach addresses a key open question in the area: how to manage dropouts while ensuring that each potential panelist is chosen with relatively *equal* probabilities. Using real-world datasets, we compare our algorithms to existing benchmarks, and we offer the first characterizations of tradeoffs between robustness, loss, and equality in this problem.

1 INTRODUCTION

Citizens assemblies are a method for democratic decision-making where a representative group of citizens is randomly selected to convene, deliberate, and produce collective policy proposals. Citizens' assemblies are on the rise globally: in just the past few years, assemblies have been run by, e.g., the European Commission on mobility and food waste (European Commission, 2024, 2023); the Netherlands on climate policy (Nationaal Burgerberaad Klimaat, 2025); and Ireland on drug use (Citizens' Assembly (Ireland), 2024).

We study the task of *sortition*, i.e., the random selection of the ***panel*** of assembly participants. In sortition, one must randomly choose a panel from a given ***pool*** of volunteers. This pool is usually highly self-selected on important demographic and ideological dimensions. Motivated by the demands of sortition in practice and decades of political theory (e.g., see Carson & Martin (1999)), we adopt the same three algorithmic ideals for the sortition process as considered in past work (see Section 1.2):

I) Panel size. The panel should be of some fixed, practitioner-chosen size, due to logistical constraints and attributed costs per participant. Let this desired panel size be $k \in \mathbb{N}$.

II) Representation. The panel must be "representative" of the population. Although the precise definition of representation is debated, the ability for practitioners to actively *control* representation is essential, to ensure they do not replicate the skew of the pool. In practice, this control is usually achieved via hard *quotas*: practitioner-defined upper and lower bounds on the number of panelists from predefined groups (defined by, e.g., age, education, political leaning).

III) Equality. Finally, the sortition process should select all pool members with relatively equal *selection probabilities*, i.e., chances of being chosen for the panel. While the need to reverse the skew of the pool makes it mathematically impossible to give pool members *exactly equal* selection probabilities, making these selection probabilities *as equal as possible* is a core normative principle of sortition (Carson & Martin, 1999). Further, past work formally shows that very *high* or very *low* selection probabilities concretely compromise fairness and create manipulation incentives (Baharav

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

& Flanigan, 2024). We therefore adopt the perspective that a lottery is more *equal* if its maximum selection probability is lower and its minimum is higher; formally, for $\alpha, \beta \in [0,1]$ with $\alpha \leq \beta$, we say that a sortition procedure is (α, β)-equal if the selection probabilities it realizes are bounded in $[\alpha, \beta]$.

Given ideals I-III, the **standard sortition task** is then: given a pool, sample a panel from that pool that satisfies desiderata I and II, while achieving desideratum III to the highest degree possible.

The dropouts problem. After much research on the standard sortition task, a recent paper by Assos et al. (2025) pointed out a threat to I. PANEL SIZE and II. REPRESENTATION that was not encompassed by this task's standard formulation: ***dropouts***. The problem of attrition and last-minute dropouts are established and studied concerns in deliberative democracy and survey methodology literature Karjalainen & Rapeli (2015). In practice, some panelists often drop out on or around the first day of the assembly, compromising REPRESENTATION and/or leaving the PANEL SIZE too small. Assos et al. identified various measures that practitioners take in anticipation of dropouts: choosing *alternates* (people to have on hand to replace dropouts as needed), or choosing a larger panel initially to make it *robust* to dropouts. With either approach, the key challenge is that dropouts are last-minute, so the alternates/panel must be selected *before* seeing who drops out.

Assos et al. (2025) modeled each panelist's dropout event as an independent Bernoulli trial, and they proposed an empirical risk minimization (ERM)-based algorithm (henceforth called *ERM-Benchmark*) that does the following: given the probabilities that each pool member will drop out if chosen for the panel, find the optimal set of alternates / robust panel. Here, "optimal" means the set that minimizes the *Loss*, i.e., the deviation of the final panel from the quotas, in expectation over the randomly-drawn dropouts.

Our goal. Our high-level goal is to design a sortition procedure that directly produces a dropout-robust panel (rather than selecting alternates).[1] We want to satisfy ideals I - III *accounting for dropouts:* This means that the panel *after dropouts and corrective measures* would ideally be of size k (I) and satisfy the quotas (II). For ideal III, we want our procedure to ensure that no pool member has more than β or less than α chance of reaching the panel.

[1]Selecting a dropout-robust panel directly (rather than first selecting a panel, *then* choosing extras or alternates) is logistically simpler, and allows us to more directly examine fundamental trade-offs without the additional constraint that a panel has been pre-chosen by an external algorithm.

Our proposed sortition algorithm aims to close two remaining gaps in how the *ERM-Benchmark* addresses these ideals, outlined below.

Gap 1. *Robustness of* I. PANEL SIZE, II. REPRESENTATION *to correlated dropout events with unknown probabilities.* The *ERM-Benchmark* takes the dropout probabilities as inputs, but in practice they are actually unknown. While Assos et al. points out that they can be estimated from past assembly data, such data is limited by difficulties combining data containing heterogeneous features, and dropout probabilities can be unpredictably affected by assembly-specific shocks (e.g., the policy topic, communication failures, weather). Put more formally, we contend that if $\mathbf{d} \in [0,1]^n$ is the *true* vector of marginal probabilities with which each pool member in $1 \dots n$ will drop out, our estimated version of this vector $\tilde{\mathbf{d}} \in [0,1]^n$ could be quite different. Assos et al. (2025) do tightly characterize their algorithms' loss under reasonable prediction errors γ, satisfying $\|\mathbf{d} - \tilde{\mathbf{d}}\|_\infty \leq \gamma$ (where γ is generic in $[0,1]$). While this loss is bounded, it is potentially quite large.[2] This large loss is unsurprising because their algorithm is not *designed* to be robust to such errors — it simply accepts $\tilde{\mathbf{d}}$ as the truth.

This brings us to our first algorithmic goal: given $\tilde{\mathbf{d}}$, we want to compute a panel that does as well as possible on I. PANEL SIZE and II. REPRESENTATION after dropouts are drawn from the worst-case distribution with marginal dropout probabilities $\mathbf{d} : \|\mathbf{d} - \tilde{\mathbf{d}}\|_\infty \leq \gamma$. Here, γ can then be interpreted as the *robustness level* of such an algorithm, and can be chosen by the user. Importantly, while we constrain errors in *marginal* dropout probabilities the same way as Assos et al., we aim to be robust against an even strong adversary: while Assos et al. assume dropouts occur *independently* according to $\tilde{\mathbf{d}}$, we allow *arbitrary correlations* between panelists' dropout events.

Gap 2. *Achieving* III. (α, β)-EQUALITY. *ERM-Benchmark* does not consider the ideal of EQUALITY at all. In fact, it is *deterministic*: if there is a uniquely optimal set of panelists / alternates, their algorithm deterministically chooses it. As we confirm empirically in real data (Section 5), their algorithm often gives pool members selection probability 1. In addition to robustness γ, we therefore want to allow the user to control the (α, β)-EQUALITY of the selection process. This means they can provide $\alpha, \beta \in [0,1]$, and we want to guarantee that all selection probabilities induced by our sortition algorithm are in these bounds.

[2]Their bound is $\gamma|FV|$ (Thm. 4.2, 4.3, (Assos et al., 2025)). In practice, $|FV|$ is roughly 20-30, so if any $\tilde{d}_i$ is off by 10%, a quota can be violated by a factor of 2.

1.1 Approach and Contributions

More formally, our problem is as follows: We are given inputs including the pool N of size n, the desired panel size k, and the quotas; the estimated marginal dropout probabilities $\tilde{\mathbf{d}}$; the desired robustness level $\gamma \in [0,1]$; and equality requirements $\alpha, \beta \in [0,1]$. We want to output a panel $\tilde{K} \subseteq N$ such that after some set of agents D drops out, the post-dropout panel $\tilde{K} \setminus D$ has minimal *Loss*, i.e., only minimally violates the quotas and desired panel size. For the algorithm to be *optimally* γ-robust, it must minimize the expected loss of $\tilde{K} \setminus D$ assuming that D is sampled according to an adversarially chosen dropout distribution δ. Our algorithmic objective takes the form of a minimax optimization problem, resulting from the tension between loss-minimizing panel selection and the loss-maximizing adversary. The only constraint on δ is that the marginal dropout probabilities $\mathbf{d}$ it implies are constrained so that $\mathbf{d} : \|\mathbf{d} - \tilde{\mathbf{d}}\|_\infty \leq \gamma$ — i.e. its marginal dropout probabilities are in the γ-radius L_∞ ball around $\tilde{\mathbf{d}}$. We let $\Delta(\tilde{\mathbf{d}}, \gamma)$ be the set of all joint distributions over dropout sets D consistent with marginal probabilities $\mathbf{d}$ lying in the γ-ball around $\tilde{\mathbf{d}}$. The global minimax problem we would ideally like to solve is then

$$\min_{\tilde{K} \subseteq N} \max_{\delta \in \Delta(\tilde{\mathbf{d}}, \gamma)} \mathbb{E}_{D \sim \delta}[Loss(\tilde{K} \setminus D)].$$

Key challenge 1. The first challenge here is that the set of all possible dropout sets D is exponential, meaning the inner expectation is a sum over *exponentially many* possible events. Assos et al. contend with the same problem when optimizing $\min_{\tilde{K} \subseteq N} \mathbb{E}_{D \sim \tilde{\mathbf{d}}}[Loss(\tilde{K} \setminus D)]$ — our problem without the inner max — and they deal with it using ERM: they empirically estimate the distribution over D via sampling (showing that this estimate is good enough via uniform convergence), and then using integer programming to select $\tilde{K}$. We cannot use ERM due to the added maximization step, because there is *no fixed distribution to sample* — we need to simultaneously optimize our panel (an integer problem) and the dropout distribution playing against it (a continuous problem).

Contribution I: A sortition algorithm that is near-optimally γ-robust and (α, β)-Equal. We circumvent the intractability of this bilevel mixed-integer optimization problem by first solving its *continuous relaxation*, and then dependently rounding the resulting fractional panel. Our algorithm thus proceeds in two steps: first, it solves the *fractional version* of this problem optimally, and then it rounds this fractional panel into the panel $\tilde{K}$.

Step 1: Compute a fractional panel. We compute the *fractional panel* $\hat{\mathbf{p}} \in [0,1]^n$ that solves the following minmax optimization problem.

$$\min_{\mathbf{p}} \max_{\delta \in \Delta(\tilde{\mathbf{d}}, \gamma)} \mathbb{E}_{D \sim \delta}[Loss(\mathbf{p} \setminus D)] \quad \textbf{s.t.} \quad \mathbf{p} \in [\alpha, \beta]^n.$$

Here, $\mathbf{p} \setminus D$ represents the (fractional) panel post-dropout. Note that this is exactly the continuous relaxation (i.e., pool members are treated as divisible) of our overall set selection problem, up to one addition: we require that all agents are fractionally included to a degree within $[\alpha, \beta]$.

In Section 3, we give an algorithm, MinMax-Optimizer (Algorithm 1), for solving this minmax problem. This algorithm plays two subroutines against each other over T rounds: in each round t, the Minimizer (Algorithm 3) uses projected gradient descent to design a fractional panel $\mathbf{p}^t$, and then the Maximizer (Algorithm 2) best-responds with the distribution over dropout sets $\delta^t \in \Delta(\tilde{\mathbf{d}}, \gamma)$ that maximizes loss against $\mathbf{p}^t$. We prove that the *average* fractional panel $\mathbf{p}^t$ produced over iterates $t \in [T]$ converges to the optimal fractional panel $\mathbf{p}^*$ at a rate $\sqrt{n/T}$ (Theorem 3.1).

Key challenge 2. The key technical difficulty is that the inner maximization problem over the dropout distribution δ — which must be solved by the Maximizer algorithm — is defined over distributions with exponentially large support (all possible dropout sets D). This maximization problem can be expressed as a linear program with $O(n)$ constraints (one for each of the marginal probability upper and lower bounds), but 2^n variables (one for each dropout set D).

We show that the inner maximization problem can nevertheless be solved efficiently by instead solving the *dual* linear program (which has $O(n)$ variables but 2^n constraints) using the Ellipsoid algorithm with a polynomial-time separation oracle. Informally this is because our objective is the maximum of only polynomially many *modular* functions in D, whose maximizer must be the maximizer of at least one of the constituent modular functions. The set-valued maximizer for a modular function can be computed as a simple thresholding, and our separation oracle amounts to finding the optimizer for each of the constituent modular components of the objective and then taking the maximum over them. Finally we show that an optimal primal solution can be extracted from the optimal dual solution. Together this gives an efficient algorithm to solve the fractional minimax problem, despite the fact that the "strategy space of the maximization player" is exponentially large.

Step 2: Dependently round. We then dependently round our optimal fractional panel $\hat{\mathbf{p}}$ into the panel $\tilde{K}$. For this, we use a known polynomial-time dependent

rounding procedure, *Pipage Rounding* (Gandhi et al., 2006). In Section 4, we apply *Pipage*'s marginal preservation and negative association properties to prove bounds on our algorithm's achievement of ideals I-III.

Contribution II: Real-world trade-offs and algorithm performance. Our algorithms' optimality (at least in the fractional problem) allows us to explore trade-offs in this problem that were previously inaccessible. In Section 5, we ask: *Is it worse to be overly robust when $\tilde{d}$ is a good estimate, or non-robust when $\tilde{d}$ is a poor estimate?*, and *What is the cost to robustness and loss of achieving stronger* (α, β)-EQUALITY*?* We characterize these trade-offs, finding that overestimating the power of the adversary (i.e., being too robust) tends to be *costlier* than being not robust enough. We also find that tightening $[\alpha, \beta]$ has a roughly linear effect on the level of loss achievable. Finally, our algorithms' loss and that of *ERM-Benchmark* against adversaries of different strengths (i.e., values of γ).

1.2 Additional Related Work

An overview of the normative ideals of sortition, arising from the political science literature, can be found in Flanigan et al. (2021a). Our algorithmic approach is new to the sortition literature, but the peripheral tools and ideas we use track this literature closely. Ideals I-III have been pursued in various forms (Halpern et al., 2025; Flanigan et al., 2020, 2021a,b; Ebadian & Micha, 2025; Ebadian et al., 2022; Baharav & Flanigan, 2024; Assos et al., 2025); the dependent rounding procedure has been applied to the sortition problem (though for rounding a different object) (Flanigan et al., 2021b); prior work has also studied the continuous relaxation of the panel selection problem (Flanigan et al., 2020, 2024); and column generation is a common approach in the sortition literature for designing distributions over intractable set spaces Flanigan et al. (2021a).

The technique we use to solve the minimax optimization problem of playing a "no-regret learning algorithm" against a "best-response algorithm" originates with Freund & Schapire (1999). This core algorithmic technique has been widely used to solve constrained optimization problems with exponentially-many constraints (as in our case) — see e.g. examples in differential privacy (Gaboardi et al., 2014), algorithmic fairness (Kearns et al., 2018), and conditional calibration (Haghtalab et al., 2023). In all of these cases, the technique requires applying an appropriate no-regret learning algorithm (we use online gradient descent (Zinkevich, 2003)) and then giving an efficient algorithm to solve a "best response" problem (in our case, our algorithm for computing the worst-possible dropout distribution given a fractional panel). The details of these solutions typically differ in which algorithmic techniques are used to solve the "best response" problem for the player that has the large strategy space: in our case this manifests itself in our solution to the dropout player's best response problem using Ellipsoid on the dual of their best response problem.

The set of dropout distributions we solve the minimax optimization problem over are specified by linear moment constraints on marginals and expected size, which is related to strategies used for distributionally robust optimization (DRO). Classical and modern DRO results (moment sets, φ-divergences, and Wasserstein balls) typically study convex sample-average losses; in contrast, we have a combinatorial objective defined via quotas; this is what necessitates our no-regret vs. best response style algorithm rather than using a closed-form dual reformulation. (Delage & Ye, 2010; Wiesemann et al., 2014; Mohajerin Esfahani & Kuhn, 2018; Rahimian & Mehrotra, 2019).

2 MODEL

Instance. An instance of our problem $\mathcal{I} = (N, k, \boldsymbol{\ell}, \mathbf{u})$ consists of a *pool* N (a set of agents) of size n, a panel size $k \in \mathbb{N}_{>0}$, and *quotas*. These quotas are imposed on *feature-value* pairs, defined as follows: let the set of possible values of each feature $f \in F$ be V_f, and let $\text{FV} := \{(f, v) : f \in F,\ v \in V_f\}$ be the set of all feature-value pairs. As an simple example, we could have feature set $F = \{\text{age, height}\}$ with $V_{\text{height}} = \{\text{short}, \text{tall}\}$; a feature-value pair would be (height,tall). Then, a pair of lower and upper quotas on (f, v) ad defined respectively as $\ell_{f,v}, u_{f,v} \in \mathbb{N}$, where $1 \leq \ell_{f,v} \leq u_{f,v}$. We summarize the quotas as $\boldsymbol{\ell} = (\ell_{f,v} | f, v \in FV)$ and $\mathbf{u} = (u_{f,v} | f, v \in FV)$.

To map features to their respective values, we let $f \in F$ act as a function. Let $f : N \to V_f$ so that $f(i)$ is agent i's value for feature f, and let $N_{f,v} := \{i \in N : f(i) = v\}$ be the set of pool members with feature-value (f, v). Then, a *panel* $K \subseteq N$ *satisfies* the quotas if $|K \cap N_{f,v}| \in [\ell_{f,v}, u_{f,v}]$ for all $(f, v) \in \text{FV}$.

Panel size as a quota. We encode the panel size constraint via an extra feature f^k with a single value $V_{f^k} = \{1\}$. We set $\ell_{f^k,1} = u_{f^k,1} = k$, and implicitly include $(f^k, 1)$ in FV and its quotas in $\boldsymbol{\ell}, \mathbf{u}$.

Selection probabilities. A *selection algorithm* is any procedure for taking in an instance $(N, k, \boldsymbol{\ell}, \mathbf{u})$ and outputting a panel K. Any selection algorithm induces *selection probabilities* $\boldsymbol{\pi} = (\pi_i)_{i \in N}$, which we define such that $\pi_i = \Pr[i \in \tilde{K}]$ is the chance of being chosen for the *initial* panel $\tilde{K}$.[3]

[3]We define selection probabilities as the chance of being chosen for the *initial panel*, rather than being chosen *and not dropping out*. This choice is most faithful to the known

2.1 Distributional Dropout Adversary

Let $\tilde{\mathbf{d}} \in [0,1]^n$ be estimated per-agent dropout probabilities and fix a robustness radius $\gamma \in [0,1]$. Instead of assuming independent dropouts across agents, we allow the adversary to select an *arbitrary dropout distribution* δ over dropout sets $D \subseteq N$, subject to linear marginal constraints:

$$\Delta(\tilde{\mathbf{d}}, \gamma) := \Big\{ \delta \in \Delta(2^N) : \sum_i \Pr[i \in D] = \sum_i \tilde{d}_i \wedge \forall i, \Pr_{D\sim\delta}[i \in D] \in [\max(0, \tilde{d}_i - \gamma), \min(1, \tilde{d}_i + \gamma)] \Big\} \quad (1)$$

The first equality constraint fixes the *expected number* of dropouts in the pool to match $\tilde{\mathbf{d}}$; the second bounds the marginal deviation of $\mathbf{d}$ from $\tilde{\mathbf{d}}$ within a γ-ball.

This formulation allows an adversary to optimize over the set of all (possibly correlated) distributions over dropouts that satisfy marginal dropout probabilities constrained to be within a γ-ball around the given marginal probability vector — a more powerful adversary than one who optimizes over only independent Bernoulli dropouts.

2.2 Loss of a (Fractional) Panel

For any fractional panel $\mathbf{p}$, dropout set $D \subseteq N$, and feature-value pair $(f,v) \in FV$, let $S_{f,v}$ be the expected number of fractional panelists with feature value pair (f,v) post-dropout:

$$S_{f,v}(\mathbf{p}; D) := \sum_{i \in N_{f,v}} p_i \, \mathbb{I}(i \notin D), \quad (2)$$

Let the deviations of this post-dropout panel from the quotas on (f,v), normalized by the upper quota be

$$\mathrm{dev}^-_{f,v}(\mathbf{p}; D) = \tfrac{1}{u_{f,v}}(\ell_{f,v} - S_{f,v}(\mathbf{p}; D)), \quad (3)$$

$$\mathrm{dev}^+_{f,v}(\mathbf{p}; D) = \tfrac{1}{u_{f,v}}(S_{f,v}(\mathbf{p}; D) - u_{f,v}). \quad (4)$$

Now, let $h(\mathbf{p}; D)$ be the maximum normalized deviation across *all* feature-values. Finally, fixing a dropout distribution δ, let the overall *loss* be the expectation of this maximum deviation over dropouts $D \sim \delta$:

$$h(\mathbf{p}; D) := \max_{(f,v)\in \mathrm{FV},\ \dagger\in\{-,+\}} \mathrm{dev}^{\dagger}_{f,v}(\mathbf{p}; D) \quad (5)$$

$$\mathcal{L}(\mathbf{p}, \delta; \mathcal{I}) := \mathbb{E}_{D\sim\delta}\big[\, h(\mathbf{p}; D)\,\big]. \quad (6)$$

We remark that this objective is the *max* over features rather than the sum, as used by Assos et al. (2025); We select this version for both technical convenience and because we find guarantees on maximum deviation, rather than average, more interpretable as a measure of quota deviation.

relationship between selection probabilities and manipulation (Flanigan et al., 2024), and reflects the sortition principle of *equality of opportunity* (Carson & Martin, 1999).

Action spaces. The maximizer acts in $\Delta(\tilde{\mathbf{d}}, \gamma)$, as in from equation 1. We constrain the minimizer so they include each agent to a fractional degree within $[\alpha, \beta]$:

$$\mathfrak{P}(\alpha, \beta) := [\alpha, \beta]^n \subseteq [0,1]^n. \quad (7)$$

One may want to additionally constrain the fractional panel size as $\|\mathbf{p}\|_1 = k^+$, for some $k^+ \geq k$ — this does not affect the convexity. Our fractional optimization problem can now be expressed as the convex–concave saddle-point-problem:

$$\mathbf{p}^* := \arg \min_{\mathbf{p}\in\mathfrak{P}(\alpha,\beta)} \max_{\delta\in\Delta(\tilde{\mathbf{d}},\gamma)} \mathcal{L}(\mathbf{p}, \delta; \mathcal{I}). \quad (8)$$

Convexity–Concavity (proof in Section 7.2). For any fixed δ, $\mathcal{L}(\cdot, \delta; \mathcal{I})$ is convex in $\mathbf{p}$ because $h(\mathbf{p}; D)$ is a pointwise maximum of finitely many affine forms in $\mathbf{p}$, and expectation preserves convexity. For any fixed $\mathbf{p}$, $\mathcal{L}(\mathbf{p}, \cdot; \mathcal{I})$ is affine (hence concave) in δ by linearity of expectation. Since $\mathfrak{P}(\alpha, \beta)$ and $\Delta(\tilde{\mathbf{d}}, \gamma)$ are convex and compact, Sion's minimax theorem then implies a saddle point exists and the max and min in Equation (8) can be exchanged.

3 OUR OPTMIZATION ALGORITHM

We solve the minmax optimization problem in equation 8 by playing a no-regret minimizer against an exact best response of the maximizer. The minimizer runs projected subgradient descent; the maximizer solves a linear program by running Ellipsoid using an efficient separation oracle on its dual (Appendix 7.1).

3.1 Algorithm

Our overall algorithm is Algorithm 1, and its subroutines are Algorithms 3 and 2. Let $T \in \mathbb{N}$ and a stepsize $\eta > 0$ be given.

Algorithm 1 MinMax-Optimizer

Require: $\mathcal{I}$, $\tilde{\mathbf{d}}$, γ, α, β, η, T.
1: $\mathbf{p}^0 \leftarrow \{(\alpha+\beta)/2\}_{i=1}^n$; choose feasible $\delta^0 \in \Delta(\tilde{\mathbf{d}}, \gamma)$
2: **for** $t = 0, 1, \ldots, T-1$ **do**
3: $\quad \mathbf{p}^{t+1} \leftarrow$ Projected-Subgradient$(\mathbf{p}^t, \delta^t, \eta)$
4: $\quad \delta^{t+1} \leftarrow$ Best-Response-Ellipsoid$(\mathbf{p}^{t+1}, \tilde{\mathbf{d}}, \gamma)$
5: **end for**
6: **return** $\hat{\mathbf{p}} := \frac{1}{T}\sum_{t=1}^T \mathbf{p}^t$

Minimizer: Projected-Subgradient. Note that for fixed δ, the map $\mathbf{p} \mapsto \mathcal{L}(\mathbf{p}, \delta; \mathcal{I})$ is convex and

piecewise linear. A single gradient step proceeds as follows. For each dropout set in the support of δ we calculate the sub-gradient in accordance with the loss function, update $\mathbf{p}^t$ with the final gradient and clip the final probabilities to be in our action space $[\alpha, \beta]$. We defer the full details of this algorithm to the Appendix (algorithm 3).

Maximizer: BEST-RESPONSE-ELLIPSOID. For fixed $\mathbf{p}$, the best response problem for the adversary is to solve the following linear program.

$$\begin{aligned} \max_{\delta \geq 0} \quad & \sum_{D \subseteq N} \delta(D)\, h(\mathbf{p}; D) \\ \text{s.t.} \quad & \sum_{D \subseteq N} \delta(D) = 1, \\ \wedge \quad & \sum_{D \subseteq N} |D|\, \delta(D) = c \\ \wedge \quad & \sum_{D \ni i} \delta(D) \in [m_i, M_i]\ \forall i. \end{aligned}$$

with $m_i = \max(0, \tilde{d}_i - \gamma)$, $M_i = \min(1, \tilde{d}_i + \gamma)$, The dual linear program (below) has the following variables, corresponding to the constraints in the primal: $\lambda \in \mathbb{R}$ (normalization), $\tau \in \mathbb{R}$ (expected size), and $a_i, b_i \geq 0$ (upper, lower marginals).

$$\begin{aligned} \min_{\lambda, \tau, \mathbf{a}, \mathbf{b} \geq 0} \quad & \lambda + c\,\tau + \sum_i M_i a_i - \sum_i m_i b_i \\ \text{s.t.} \quad & \lambda + \tau |D| + \sum_{i \in D} (a_i - b_i) \geq h(\mathbf{p}; D) \quad \forall D \subseteq N. \end{aligned}$$

We show that this best-response problem can be solved in polynomial time via the Ellipsoid method applied to the dual, using a polynomial-time separation oracle. This separation oracle evaluates $\max_{D \subseteq N} \{h(\mathbf{p}; D) - \sum_{i \in D}(a_i - b_i + \tau)\}$ in $O(|\mathrm{FV}|\, n)$ time by enumerating the $2|\mathrm{FV}|$ linear pieces and thresholding. Ellipsoid yields a dual optimum in polynomial time (Theorem 7.2); we then recover a primal optimum by solving the primal restricted to the polynomial set of subsets returned during Ellipsoid (Theorem 7.3).

3.2 Convergence and Efficiency

We next prove two key properties of Algorithm 1: it converges to optimality as T grows (Theorem 3.1), and it is computationally efficient (Theorem 3.2).

Theorem 3.1 (Optimality) *Fix algorithm inputs $\mathcal{I}, \tilde{\mathbf{d}}, \alpha, \beta, \gamma, \eta, T$; let $\hat{\mathbf{p}}$ be the output of Algorithm 1 on these inputs; and let $\mathbf{p}^*$ be the optimal solution for these inputs, as in Equation (8). Then,*

$$\max_{\delta \in \Delta(\tilde{\mathbf{d}}, \gamma)} \mathcal{L}(\hat{\mathbf{p}}, \delta; \mathcal{I}) \;\leq\; \max_{\delta \in \Delta(\tilde{\mathbf{d}}, \gamma)} \mathcal{L}(\mathbf{p}^*, \delta; \mathcal{I}) \;+ O\Big(\sqrt{\tfrac{n}{T}}\Big).$$

Algorithm 2 BEST-RESPONSE-ELLIPSOID$(\mathbf{p}, \tilde{\mathbf{d}}, \gamma)$

1: Initialize collection of subsets $\mathcal{C} \leftarrow \emptyset$.
2: Run Ellipsoid on the dual variables $(\lambda, \tau, \mathbf{a}, \mathbf{b})$, using the separation oracle that returns a maximizing D for $h(\mathbf{p}; D) - \sum_{i \in D}(a_i - b_i + \tau)$.
3: Each time the oracle finds a violated constraint, add the returned D to $\mathcal{C}$.
4: After Ellipsoid terminates with a feasible dual, solve the primal LP restricted to variables $\{\delta(D) : D \in \mathcal{C}\}$ subject to the original constraints.
5: **return** the optimal restricted primal solution (extended by zeros off $\mathcal{C}$). This is optimal for the full primal (Theorem 7.3).

Proof sketch. Since $\mathcal{L}(\cdot, \delta)$ is convex and has bounded subgradients (each coordinate bounded by $1/u_{f,v} \leq 1$), projected subgradient descent enjoys $O(\sqrt{n/T})$ regret. Combining standard no-regret-to-saddle-point arguments for convex–concave games (e.g., Freund & Schapire (1999)) with the exact best responses yields the bound. See Appendix 7.4 for the formal proof. □

Theorem 3.2 (Efficiency) MINMAX-OPTIMIZER *is polynomial time.*

Proof Sketch. Each PROJECTED-SUBGRADIENT iteration computes $\mathbf{p}^{t+1}$ in $O(|\mathrm{supp}(\delta^t)|\, |\mathrm{FV}|\, n)$ time (where $\mathrm{supp}(\delta^t)$ must be polynomial-size because exactly one D is added to the support for each Ellipsoid iteration). Then, BEST-RESPONSE-ELLIPSOID computes δ^{t+1} in time polynomial in $(n, |\mathrm{FV}|, L)$ via Ellipsoid on the dual using the $O(|\mathrm{FV}|\, n)$ separation oracle, followed by solving a polynomial-size restricted primal (Theorems 7.2 and 7.3). Hence, MINMAX-OPTIMIZER is polynomial time. □

4 GUARANTEES ON IDEALS I–III

Our selection algorithm. Our overall selection algorithm, which we call MINMAX-PIPAGE, takes as input an instance $\mathcal{I}$ along with γ, α, β. It first runs MINMAX-OPTIMIZER to produce an optimal fractional panel $\hat{\mathbf{p}}$; then, we round $\hat{\mathbf{p}}$ into an initial panel $\tilde{K}$ via *Pipage Rounding*. At a high level, Pipage Rounding iteratively adjusts the fractional elements of $\hat{\mathbf{p}}$ while preserving feasibility until all entries become integral, forming $\tilde{K}$ (Gandhi et al., 2006). We note its key properties when we apply them below.

We first bound the extent to which MINMAX-PIPAGE satisfies I. PANEL SIZE and II. REPRESENTATION (Theorem 4.1). These ideals are evaluated on the post-dropout panel $\tilde{K} \setminus D$ in expectation over $D \sim \delta$, where $\delta \in \Delta(\tilde{\mathbf{d}}, \gamma)$ is the adversarial distribution against $\hat{\mathbf{p}}$. Formally, we give a high-probability bound (over the

randomness of *Pipage*) on the per-(f, v) increase in quota deviation between $\mathbf{p}^*$, the optimal *fractional* panel, and our integer panel $\tilde{K} \setminus D$.

Theorem 4.1 (Representation and Panel Size) *Fix inputs $\mathcal{I}, \tilde{\mathbf{d}}, \gamma, \alpha, \beta, \eta, T$, and let $\tilde{K}$ be the (random) panel produced by* MinMax-Pipage. *Fix any $\delta \in \Delta(\tilde{\mathbf{d}}, \gamma)$, any $(f, v) \in FV$, and any $z > 0$. With probability at least $1 - 2\exp\left(-\frac{2z^2}{\sigma_{f,v}^2}\right)$,*

$$\max_{\dagger \in \{-,+\}} \mathbb{E}_{D \sim \delta}\left[\mathrm{dev}_{f,v}^{\dagger}(\tilde{K}; D)\right] \leq \mathcal{L}(\mathbf{p}^*, \delta; \mathcal{I}) + \epsilon + \tfrac{z}{u_{f,v}},$$

where $\sigma_{f,v}^2 = \sum_{i \in N_{f,v}} \Pr_{D \sim \delta}[i \notin D]^2$, $\epsilon = O\left(\sqrt{\frac{n}{T}}\right)$.

With high probability, parametrized by z, our deviation from the quotas for any feature value pair is not worse than the deviation of the optimal panel, $\mathbf{p}^*$, by more than the sum of ϵ and a factor of z (normalized by quota magnitude).

Proof Sketch. The proof proceeds in two steps. First, we bound the gap between $\mathbf{p}^*$ and $\hat{\mathbf{p}}$ using Theorem 3.1, thus giving the ϵ term. Then, to bound the gap between $\tilde{K}$ and $\hat{\mathbf{p}}$, we apply two properties of *Pipage Rounding*: it is *marginal-preserving*, i.e., $\Pr[i \in \tilde{K}] = p_i$ and $\boldsymbol{\pi} = \hat{\mathbf{p}}$; and the indicators $\mathbb{I}(i \in \tilde{K})$ are negatively associated (Gandhi et al., 2006). From these properties — with some careful handling of the expectation over δ — we show that the number of panelists on $\tilde{K}$ with feature-value (f, v) concentrates around its expectation, i.e., the fractional number of panelists in this group on $\hat{p}_i$. □

Finally, (α, β)-Equality is evaluated on the selection probabilities $\boldsymbol{\pi}^*$ implied by MinMax-Pipage. That $\boldsymbol{\pi}^*$ satisfies this property simply follows from the fact that *Pipage Rounding* is marginal-preserving: for all i, $\pi_i^* = \hat{p}_i \in [\alpha, \beta]$, where the last step is due to the minimizer's action space (Equation (7)).

Theorem 4.2 (Equality) MinMax-Pipage *satisfies* (α, β)-Equality.

5 EMPIRICAL EVALUATION OF ALGORITHMS

Datasets. We evaluate our algorithms on six real-world datasets, all corresponding to citizens' assemblies from varying regions of Australia. Each dataset contains an instance $\mathcal{I} = (N, k, \boldsymbol{\ell}, \mathbf{u})$. For each pool member $i \in N$, we have their feature-value pairs $f, v \in FV$, plus two binary indicators: whether they were selected for the panel, and whether or not they dropped out conditional on being selected for the panel. See Section 9.1 for details.

Algorithm inputs $\tilde{\mathbf{d}}, \gamma, \alpha, \beta, \eta, T$. We estimate $\tilde{\mathbf{d}}$ for each instance by fitting the same *independent action* model used in Assos et al. (2025), which assumes each feature-value contributes independently to an agent's dropout probability. Our training data thus consists of agents' features (inputs) and observed dropout behavior (labels). For each instance, we train our model on three other instances (which three varies per instance). See Section 9.3 for details. We choose α and β jointly chosen to fall within a $\kappa \in (0, 1]$ fraction of the distance between the average selection probability k^+/n and the extremes 0 and 1, where $k^+ = k + k \cdot \mathbb{E}_{D \sim \tilde{\mathbf{d}}}[|D|]/n$, a slight inflation of the target panel size to offset the average dropout rate. Formally, for $\kappa \in [0, 1]$,

$$\alpha(\kappa) := (1 - \kappa) \cdot k^+/n, \quad \beta(\kappa) := k^+/n + (1 - k^+/n)\kappa.$$

All experiments use $T = 1000$ and $\eta = \sqrt{n/T}$. We show empirical convergence in Section 9.4.

True dropout probabilities d. While in theory δ should be computed by Best-Response-Ellipsoid, the Ellipsoid algorithm is impractical to implement. We thus restrict the space of dropout distributions to all *product distributions*. Then, our maximizer only needs to consider vectors of marginal dropout probabilities for each pool member $\mathbf{d} \in [0, 1]^n$. This restricted best-response problem has a simple, efficient, and optimal greedy solution we call Greedy-Maximizer. Details are in Section 9.2.

In each instance, we sample dropouts independently according to $\mathbf{d}$, the output of Greedy-Maximizer with inputs $\mathcal{I}, \tilde{\mathbf{d}}, \gamma', \mathbf{p}$. Here, γ' reflects the *true* errors in our predictions and $\mathbf{p}$ is the panel whose robustness is being evaluated. For ERM-Benchmark, $\mathbf{p}$ is a 0/1 vector representing an integer panel, and for us $\mathbf{p}$ is $\hat{\mathbf{p}}$.

5.1 Fundamental Tradeoffs

We now examine two fundamental trade-offs. We consider only our algorithms, as there *is no* prior work to compare to here: our algorithm is the first to enable these tradeoffs' exploration. We analyze MinMax-Optimizer's output $\hat{\mathbf{p}}$ directly to more crisply observe the trade-offs without the noise of rounding. Here, γ is the robustness level of our algorithm, and γ' is the true power of the adversary.

***Robustness* versus *Non-robustness*.** Here, we ask: what is the cost/benefit of being *robust* when $\tilde{\mathbf{d}}$ is closer to the truth $\mathbf{d}$, versus being *non-robust* when $\tilde{\mathbf{d}}$ is far from the truth $\mathbf{d}$? We explore this in Fig. 1 by examining $\mathcal{L}(\hat{\mathbf{p}}, \mathbf{d}; \mathcal{I})$ at each combination of $\gamma, \gamma' \in \{0, 0.15, 0.30, 0.45, 0.60, 0.75\}^2$. We show results for one representative instance; the rest are in Section 9.6. Across instances, we observe that the smallest value in each row lies on the diagonal; this indicates that ideally, $\gamma = \gamma'$ (i.e., we train with knowledge of the degree of errors). Given that γ' may not

actually be known, we can compare the losses on the left of the diagonal — where we underestimate $\tilde{\mathbf{d}}$'s errors ($\gamma < \gamma'$) — to those on the right, where we overestimate it ($\gamma > \gamma'$). The leftward values tend to be much smaller, indicating that *it is generally better to* ***underestimate*** $\tilde{\mathbf{d}}$*'s error than to overestimate it.*

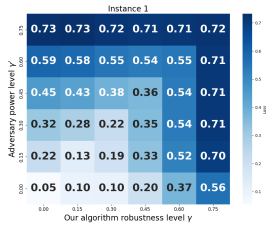


Figure 1: Here we show the loss of $\hat{\mathbf{p}}$ at each combination of γ and γ' (the adversary's strength we train and test on, respectively) in Instance 1.

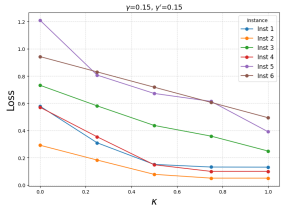


Figure 2: Here we show the loss of $\hat{\mathbf{p}}$ over $\kappa \in \{0, 0.25, 0.5, 0.75, 1\}$, i.e., as the range $[\alpha, \beta]$ expands. Errors are chosen to be moderate: $\gamma = \gamma' = 0.15$.

Equality versus Loss. Here, we ask: as we require stronger α, β-equality, how much do we compromise on loss? We test this across instances by gradually increasing the selection probability gap κ, and thus widening the interval $[\alpha, \beta]$. On the vertical axis is loss $\mathcal{L}(\hat{\mathbf{p}}, \mathbf{d}; \mathcal{I})$, where $\hat{\mathbf{p}}$, $\mathbf{d}$ are derived with $\gamma = \gamma' = 0.15$. In Fig. 2, we see that across all instances, increasing κ decreases the loss roughly linearly with some mild convexity. From $\kappa = 0$ (near equal probabilities) to $\kappa = 1$ (some probabilities equal to 0 or 1), the loss drops by 0.3-0.8, corresponding to the worst quota violation shrinking by 30%-80% of its upper quota's magnitude.

5.2 Comparison of Algorithms on I - III

We now compare our algorithm, MinMax-Rounding, to ERM-Benchmark (Assos et al., 2025), where ERM-Benchmark is configured to select an entire panel and modified to optimize a notion of loss comparable to ours. We describe our implementation of the benchmark in Section 9.5.

I. Panel Size: In Table 1, we compare (per instance) the ideal panel size k with the expected post-dropout panel size $E_{D\sim\mathbf{d}}[|K \setminus D|]$ for several panels K: the fractional optimum $\hat{\mathbf{p}}$, the panel produced by MinMax-Pipage $\tilde{K}$, and the panel produced by ERM-Benchmark K^{ERM}. Our algorithms are tested on dropout sets drawn from $\mathbf{d}$, the best-response to $\hat{\mathbf{p}}$; ERM-Benchmark is tested on $\mathbf{d}$ defined as the best-response to K^{ERM}. We see that our algorithms do far better than ERM-Benchmark on panel size.

II. Representation: In Figure 3, we compare the loss of our algorithm (both pre- and post-rounding) with that of ERM-Benchmark against increasingly adversarial dropouts $\gamma' \in \{0, 0.3, 0.6, 0.75, 1\}$ in Instance 1 (remaining instances are in Section 9.7). As expected, we see that our (fractional) algorithms eventually match or outperform ERM-Benchmark as γ' grows, i.e., as robustness becomes more important. We also see that while our fractional panel $\hat{\mathbf{p}} \setminus D$ usually outperforms or roughly matches the benchmark for low choices of γ, our rounded panel $\tilde{K} \setminus D$ has larger loss due to the cost of rounding fractional panels to integral ones via *Pipage*. This cost gap is greater for instances with smaller and/or tighter quotas, where there is less slack within which to make fractional changes and such adjustments cause larger relative loss.

Table 1: Each column (except k) is implicitly the expectation over $D \sim \mathbf{d}$. Since $\tilde{K}$ is random over rounding, we sample $\tilde{K}$ 100 times and report the mean and standard error. $\kappa = 1, \gamma = 0.15, \gamma' = 0.15$. Note that because panel size is encoded as a quota, deviations from optimal size trade off with deviations from other quotas.

$\mathcal{I}$	k	$\lvert\hat{\mathbf{p}} \setminus D\rvert$	$\lvert\tilde{K} \setminus D\rvert$	$\lvert K^{\text{ERM}} \setminus D\rvert$
1	51	46.2	46.5 ± 0.1	50.9
2	52	51.7	52.0 ± 0.1	54.5
3	41	40.3	40.5 ± 0.1	45.1
4	50	49.4	49.2 ± 0.1	54.6
5	47	49.5	49.9 ± 0.1	56.7
6	49	62.4	62.8 ± 0.1	67.8

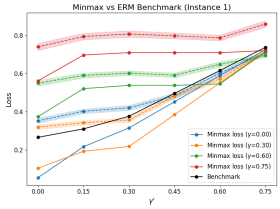


Figure 3: Here we compare $\mathcal{L}(\hat{\mathbf{p}}, \mathbf{d}; \mathcal{I})$ (solid lines), $\mathcal{L}(\tilde{K}, \mathbf{d}; \mathcal{I})$ (dotted lines) and $\mathcal{L}(K^{\text{ERM}}, \mathbf{d}; \mathcal{I})$ (black line), over true error γ' (x axis) and robustness level γ. For $\tilde{K}$, we show averages over 100 runs of MinMax-Pipage; shaded regions are standard errors.

III. Equality: While our algorithms are α, β-equal, we empirically show that ERM-Benchmark gives minimum and maximum selection probabilities of 0 and 1 respectively in all instances (see Section 9.8).

6 DISCUSSION AND FUTURE WORK

The question that remains is: between ERM-Benchmark and our algorithms, which should one use to select a dropout-robust panel? On the dimension of representation, the comparison is mixed. On some instances, our algorithm's performance is sensitive to the choice of γ, with small deviations from $\gamma = \gamma'$ causing notable increases in representation error. In contrast, ERM-Benchmark tends to perform more consistently across values of γ'. In settings where γ' is expected to be low or is uncertain, ERM-Benchmark may therefore be the more appropriate choice. Additionally, the cost of rounding fractional panels to integer ones via Pipage can introduce further loss beyond the fractional optimum, especially on instances with small or tight quotas.

On the other hand, our algorithm offers a distinct capability that ERM-Benchmark does not: explicit control over the level of α, β-equality. That is, while ERM-Benchmark is deterministic, our methods allow the user to guarantee (if feasible) that selection probabilities remain within a fixed range.

Taken together, these results suggest that while our algorithm is not always dominant over ERM-Benchmark as a practical algorithm, it does provide an important stepping stone toward algorithms that *are*, and which resolve significant issues with the benchmark. In particular, our work provides insights on two previously inaccessible tradeoffs: the first between hedging against high errors and performing well when error is low, and and the second between representation and randomness. Our findings here suggest that future work would be well-served to work on developing algorithms that move along the smooth trade-off between randomness and representation, while there may be less benefit to making sacrifices on these dimensions in favor of robustness to worst-case errors.

Future work. This naturally leads to several open problems on handling dropouts in the sortition process. The foremost is whether we can practicably (if not efficiently) solve the direct version of our optimization problem without the fractional relaxation, as formulated in the introduction. One could also formalize the differences between selecting *alternates* versus *entire robust panels*; while the latter is more convenient, alternates are in a sense fundamentally more "powerful", because they can be selectively deployed *after* seeing the dropout set. Finally, studying the true "predictability" of dropout events in larger datasets may allow more realistic models of adversarial action spaces, leading to algorithms whose robustness is more appropriately targeted and possibly less costly when true prediction errors are low.

References

Angelos Assos, Carmel Baharav, Bailey Flanigan, and Ariel Procaccia. Alternates, assemble! selecting optimal alternates for citizens' assemblies. In *Proceedings of the 26th ACM Conference on Economics and Computation*, pp. 719–738, 2025.

Carmel Baharav and Bailey Flanigan. Fair, manipulation-robust, and transparent sortition. In *Proceedings of the 25th ACM Conference on Economics and Computation*, pp. 756–775, 2024.

Lyn Carson and Brian Martin. *Random selection in politics*. Bloomsbury Publishing, 1999.

Citizens' Assembly (Ireland). Report of the citizens' assembly on drugs use, 2024. URL https://citizensassembly.ie/previous-assemblies/assembly-on-drugs-use/report/. Final report with 36 recommendations (Jan 25, 2024).

Eric Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010. doi: 10.1287/opre.1090.0741.

Devdatt P Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *BRICS Report Series*, 3(25), 1996.

Soroush Ebadian and Evi Micha. Boosting sortition via proportional representation. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 667–675, 2025.

Soroush Ebadian, Gregory Kehne, Evi Micha, Ariel D Procaccia, and Nisarg Shah. Is sortition both representative and fair? *Advances in Neural Information Processing Systems*, 35, 2022.

European Commission. European citizens' panel on food waste, 2023. URL https://citizens.ec.europa.eu/european-citizens-panels/european-citizens-food-waste-panel_en. Citizens' recommendations informing EU food-waste policy.

European Commission. Learning mobility european citizens' panel, 2024. URL https://citizens.ec.europa.eu/european-citizens-panels/learning-mobility-panel_en. European Citizens' Panels.

Bailey Flanigan, Paul Gölz, Anupam Gupta, and Ariel D Procaccia. Neutralizing self-selection bias in sampling for sortition. *Advances in Neural Information Processing Systems*, 33:6528–6539, 2020.

Bailey Flanigan, Paul Gölz, Anupam Gupta, Brett Hennig, and Ariel D Procaccia. Fair algorithms for selecting citizens' assemblies. *Nature*, 596(7873): 548–552, 2021a.

Bailey Flanigan, Gregory Kehne, and Ariel D Procaccia. Fair sortition made transparent. *Advances in Neural Information Processing Systems*, 34:25720–25731, 2021b.

Bailey Flanigan, Jennifer Liang, Ariel D Procaccia, and Sven Wang. Manipulation-robust selection of citizens' assemblies. In *Proceedings of the aaai conference on artificial intelligence*, volume 38, pp. 9696–9703, 2024.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data. In *International Conference on Machine Learning*, pp. 1170–1178. PMLR, 2014.

Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*, 53(3): 324–360, 2006.

Nika Haghtalab, Michael Jordan, and Eric Zhao. A unifying perspective on multi-calibration: Game dynamics for multi-objective learning. *Advances in Neural Information Processing Systems*, 36:72464–72506, 2023.

Daniel Halpern, Ariel D Procaccia, Ehud Shapiro, and Nimrod Talmon. Federated assemblies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 13897–13904, 2025.

Maija Karjalainen and Lauri Rapeli. Who will not deliberate? Attrition in a multi-stage citizen deliberation experiment. *Quality & Quantity*, 49(1): 407–422, 2015. doi: 10.1007/s11135-014-9993-y.

Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International conference on machine learning*, pp. 2564–2572. PMLR, 2018.

Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1-2):115–166, 2018. doi: 10.1007/s10107-017-1172-1.

Nationaal Burgerberaad Klimaat. Nationaal burgerberaad klimaat — officiële website, 2025. URL https://burgerberaadklimaat.nl/. 175 members; national climate citizens' assembly.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019. URL https://arxiv.org/abs/1908.05659.

Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014. doi: 10.1287/opre.2014.1314.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936, 2003.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [No]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [No]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [No]
   (d) Information about consent from data providers/curators. [No]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [No]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

7 Supplemental Materials from Sections 2 and 3

7.1 Projected-Subgradient

Algorithm 3 PROJECTED-SUBGRADIENT($\mathbf{p}^t, \delta^t, \eta$)

Initialize $g \leftarrow \mathbf{0} \in \mathbb{R}^n$
for each D in the support of δ^t **do**
  $(f^*, v^*, \dagger^*) \leftarrow \underset{(f,v)\in FV, \dagger\in\{-,+\}}{\arg\max} \mathrm{dev}^{\dagger}_{f,v}(\mathbf{p}^t; D)$
  if $\dagger^*$ is $-$ **then**
    for $i \in N_{f^*,v^*}$ **do**
      $g_i \mathrel{+}= -\delta(D)\,\mathbb{I}(i \notin D)/u_{f^*,v^*}$
    end for
  else if $\dagger^*$ is $+$ **then**
    for $i \in N_{f^*,v^*}$ **do**
      $g_i \mathrel{+}= +\delta(D)\,\mathbb{I}(i \notin D)/u_{f^*,v^*}$
    end for
  end if
end for
$\mathbf{p}^{t+1} \leftarrow \mathbf{p}^t - \eta\, g$
$\mathbf{p}^{t+1} \leftarrow \big(\min\{\max(p_i^{t+1}, \alpha), \beta\}\big)_{i=1}^n$
return $\mathbf{p}^{t+1}$

7.2 Convexity–Concavity and Saddle Point

Proposition 7.1 (Convexity in p, concavity in δ) *For fixed δ, $\mathcal{L}(\cdot, \delta; \mathcal{I})$ is convex in $\mathbf{p}$; for fixed $\mathbf{p}$, $\mathcal{L}(\mathbf{p}, \cdot; \mathcal{I})$ is affine (hence concave) in δ.*

Proof. For any $D \subseteq N$, $S_{f,v}(\mathbf{p}; D) = \sum_{i\in N_{f,v}} p_i\, \mathbb{I}(i \notin D)$ is affine in $\mathbf{p}$. Therefore, $\phi_{f,v}(\mathbf{p}; D) = (1/u_{f,v}) \max\{\ell_{f,v} - S_{f,v}(\mathbf{p}; D),\, S_{f,v}(\mathbf{p}; D) - u_{f,v}\}$ is a pointwise maximum of two affine forms, thus convex. The function $h(\mathbf{p}; D) = \max_{(f,v)\in FV} \phi_{f,v}(\mathbf{p}; D)$ is a pointwise maximum of finitely many convex functions, hence convex. Expectation preserves convexity, so $\mathbf{p} \mapsto \mathcal{L}(\mathbf{p}, \delta; \mathcal{I}) = \mathbb{E}_{D\sim\delta}[h(\mathbf{p}; D)]$ is convex. For fixed $\mathbf{p}$, linearity of expectation implies that $\delta \mapsto \mathcal{L}(\mathbf{p}, \delta; \mathcal{I})$ is linear (affine) in the coordinates $\{\delta(D)\}_{D\subseteq N}$. □

Theorem 7.1 (Sion's Minimax Theorem) *If $\mathfrak{P}(\alpha, \beta)$ and $\Delta(\tilde{\mathbf{d}}, \gamma)$ are convex and compact, and $\mathcal{L}$ is convex in $\mathbf{p}$ and concave in δ, then*

$$\min_{\mathbf{p}\in\mathfrak{P}(\alpha,\beta)} \max_{\delta\in\Delta(\tilde{\mathbf{d}},\gamma)} \mathcal{L}(\mathbf{p}, \delta; \mathcal{I}) \;=\; \max_{\delta\in\Delta(\tilde{\mathbf{d}},\gamma)} \min_{\mathbf{p}\in\mathfrak{P}(\alpha,\beta)} \mathcal{L}(\mathbf{p}, \delta; \mathcal{I}).$$

To verify that Sion's minimax theorem applies to our case, it remains to verify the convexity/compactness of the action spaces for the minimization and maximization players. $\mathfrak{P}(\alpha, \beta)$ is a product of closed intervals; $\Delta(\tilde{\mathbf{d}}, \gamma)$ is the intersection of the probability simplex with linear marginal constraints and (optionally) a linear expected-size equality; both are convex and compact. Combining with the previous proposition and applying Sion's minimax theorem, we see that we can interchange the min and the max in our formulation. As the max min value of the game is equal to the min max value of the game, henceforth we will refer to simply the *value* of the game when referring to either quantity.

7.3 Polynomial-Time Best Response via Ellipsoid

For fixed $\mathbf{p}$, the best response problem for the adversary is to solve the following linear program:

$$\max_{\delta \geq 0} \sum_{D \subseteq N} \delta(D)\, h(\mathbf{p}; D) \tag{9}$$

$$\text{s.t.} \sum_{D \subseteq N} \delta(D) = 1, \qquad \sum_{D \ni i} \delta(D) \in [\ell_i^{\text{marg}}, u_i^{\text{marg}}]\ (i = 1, \ldots, n), \tag{10}$$

$$\sum_{D \subseteq N} |D|\, \delta(D) = c, \tag{11}$$

with $\ell_i^{\text{marg}} = \max(0, \tilde{d}_i - \gamma)$, $u_i^{\text{marg}} = \min(1, \tilde{d}_i + \gamma)$, and $c = \sum_i \tilde{d}_i$. We now take the dual. The dual linear program has the following dual variables, corresponding to the constraints in the primal: $\lambda \in \mathbb{R}$ (normalization), $\tau \in \mathbb{R}$ (expected size), $a_i \geq 0$ (upper marginals), $b_i \geq 0$ (lower marginals). The dual is

$$\min_{\lambda, \tau, \mathbf{a}, \mathbf{b} \geq 0} \lambda + c\,\tau + \sum_i u_i^{\text{marg}} a_i - \sum_i \ell_i^{\text{marg}} b_i \tag{12}$$

$$\text{s.t. } \lambda + \tau |D| + \sum_{i \in D} (a_i - b_i) \geq h(\mathbf{p}; D) \quad \forall D \subseteq N. \tag{13}$$

We show that this best-response problem can be solved in polynomial time via the Ellipsoid method applied to the dual, using a polynomial-time separation oracle.

Dual separation oracle in $O(|\text{FV}|\, n)$ time. Define $w_i := a_i - b_i + \tau$. The family of dual constraints is equivalent to

$$\max_{D \subseteq N} \Big\{ h(\mathbf{p}; D) - \sum_{i \in D} w_i \Big\} \;\leq\; \lambda. \tag{14}$$

To find a violated constraint (if one exists) it suffices to find D maximizing the left hand side. We now show the left-hand maximum can be evaluated in $O(|\text{FV}|\, n)$ time.

Explicit decomposition and constants. We use the fact that the function $h(D; \mathbf{p})$ is the maximum of $2|\text{FV}|$ functions, each of which are linear (modular) in D, in which each term in the maximum is defined by feature value pairs (f, v) and a sign in $\{+, -\}$. The D that maximizes this maximum must be the maximizer of a single one of the constituent modular functions. Optimizing a modular function is simple and reduces to computing a threshold rule. Our separation oracle iterates over each of the constituent modular functions, finds the maximizer for each one, then amongst these sets chooses the one that maximizes the overall objective. We now make this explicit. Recall:

$$S_{f,v}(\mathbf{p}; D) = \sum_{i \in N_{f,v}} p_i\, \mathbb{I}(i \notin D) = \underbrace{\sum_{i \in N_{f,v}} p_i}_{=: M_{f,v}} - \sum_{i \in N_{f,v} \cap D} p_i.$$

Then the two sides of the per-constraint deviation in equation 5 decompose as the maximum over the following two linear/modular functions in D:

$$\frac{1}{u_{f,v}}(\ell_{f,v} - S_{f,v}) = \underbrace{\frac{\ell_{f,v} - M_{f,v}}{u_{f,v}}}_{=: C^-_{f,v}} + \sum_{i \in N_{f,v} \cap D} \frac{p_i}{u_{f,v}},$$

$$\frac{1}{u_{f,v}}(S_{f,v} - u_{f,v}) = \underbrace{\frac{M_{f,v} - u_{f,v}}{u_{f,v}}}_{=: C^+_{f,v}} - \sum_{i \in N_{f,v} \cap D} \frac{p_i}{u_{f,v}}.$$

Thus the final objective can be written as the maximimum over $2 \cdot |\text{FV}|$ such linear/modular functions in D: $h(\mathbf{p}; D) = \max_{(f,v) \in \text{FV}} \{H^-_{f,v}(D),\, H^+_{f,v}(D)\}$ where

$$H^-_{f,v}(D) = C^-_{f,v} + \sum_{i \in N_{f,v} \cap D} \tfrac{p_i}{u_{f,v}}, \qquad H^+_{f,v}(D) = C^+_{f,v} - \sum_{i \in N_{f,v} \cap D} \tfrac{p_i}{u_{f,v}}.$$

Per-element gains and thresholding. Each of these modular functions has a simple optimizer characterized by a threshold. For any fixed term $t \in \{(f, v, \text{-}), (f, v, +)\}$, the objective in equation 14 specialized to t equals

$$H_t(D) - \sum_{i \in D} w_i = C_t + \sum_{i \in D} g_i^t, \quad \text{with} \begin{cases} g_i^{(f,v,\text{-})} = \mathbb{I}(i \in N_{f,v}) \frac{p_i}{u_{f,v}} - w_i, \\ g_i^{(f,v,+)} = -\mathbb{I}(i \in N_{f,v}) \frac{p_i}{u_{f,v}} - w_i. \end{cases}$$

Hence the inner maximum is attained by the *threshold set*

$$D_t^* := \{i \in N : g_i^t > 0\}, \qquad \text{value } = C_t + \sum_{i : g_i^t > 0} g_i^t.$$

Evaluating all $2|\mathrm{FV}|$ candidates and selecting the best one evaluates the left-hand side of equation 14 in $O(|\mathrm{FV}|\, n)$ time, furnishing a separation oracle.

Bounding the dual region. Standard LP bit-complexity bounds imply there exists an optimal dual solution with each coordinate bounded in magnitude by $2^{\mathrm{poly}(L)}$, where L is the input bit-length (of $\ell, u, \tilde{d}, \gamma, \mathbf{p}$). Intersecting the dual with this axis-parallel box preserves optimality and provides the initial ellipsoid for Ellipsoid.

Theorem 7.2 (Dual optimization via Ellipsoid) *Given* $\mathbf{p}$ *and the separation oracle above, the Ellipsoid method computes an* ε*-optimal feasible dual solution in time polynomial in* $(n, |\mathrm{FV}|, L, \log(1/\varepsilon))$*. Choosing* $\varepsilon = 2^{-\Theta(L)}$ *yields an exact rational optimum.*

Proof. The dual feasible set is a convex polyhedron contained in a box of polynomial bit-size. The separation oracle for constraints equation 14 runs in $O(|\mathrm{FV}|\, n)$ time by the decomposition and thresholding argument above. The Ellipsoid method with polynomial-size initialization and polynomial-time separation yields an ε-optimal point in a number of iterations polynomial in the dimension and $\log(1/\varepsilon)$. Since coefficients have bit-length $\le L$, taking $\varepsilon = 2^{-\Theta(L)}$ recovers an exact optimum. □

Primal recovery from violated constraints. During the Ellipsoid run, whenever a constraint is violated the oracle returns a subset $D \subseteq N$. Let $\mathcal{C}$ be the collection of all such subsets gathered by Ellipsoid (its size is polynomial).

Theorem 7.3 (Primal recovery) *Let* $\mathcal{C}$ *be as above. Consider the primal LP restricted to variables* $\{\delta(D) : D \in \mathcal{C}\}$ *and the original constraints. An optimal solution to this restricted LP, extended by zeros outside* $\mathcal{C}$*, is optimal for the full primal.*

Proof. Let $(\lambda, \tau, \mathbf{a}, \mathbf{b})$ be the Ellipsoid-output dual, which is feasible for all constraints indexed by $D \in \mathcal{C}$ and satisfies equation 14 for all $D \subseteq N$. Let $\delta^{\mathcal{C}}$ be an optimal solution of the restricted primal. By weak duality, its value is at most the dual value; by construction, all constraints corresponding to $\mathcal{C}$ are enforced. Suppose there existed a full primal solution $\tilde{\delta}$ with strictly larger objective. Then some positive mass must lie on a subset $D \notin \mathcal{C}$ with strictly positive reduced cost at optimality, contradicting that the final dual satisfies equation 14. Therefore $\delta^{\mathcal{C}}$ (extended by zeros) is optimal for the full primal. □

Corollary 7.1 (Polynomial-time best response) *The adversary's best response can be computed in time polynomial in* $(n, |\mathrm{FV}|, L)$ *by running Ellipsoid on the dual with the oracle above and then solving the primal restricted to the polynomial set of columns collected during the run; a final oracle call certifies optimality.*

Variant: inequality on expected size. If the expected-size equality is replaced by $\sum_D |D|\, \delta(D) \le c$, then $\tau \ge 0$ in the dual and the separation becomes $\max_D \{h(\mathbf{p}; D) - \sum_{i \in D}(a_i - b_i) - \tau|D|\} \le \lambda$. The same oracle and analysis apply and the statements above remain valid.

Subgradient bounds. For use in no-regret analysis, observe that each coordinate of a subgradient of $\mathcal{L}(\cdot, \delta; \mathcal{I})$ at $\mathbf{p}$ has magnitude at most $\max_{(f,v)} 1/u_{f,v} \le 1$ and arises from at most one active side at each D in the support of δ. Thus $\|g\|_2 \le \sqrt{n}$ and standard $O(\sqrt{n/T})$ rates apply.

7.4 Proof of Theorem 3.1

We restate the claim. Let $\{\mathbf{p}^t\}_{t=0}^{T-1}$ be generated by Algorithm 1 and define $\hat{\mathbf{p}} := \frac{1}{T}\sum_{t=1}^{T}\mathbf{p}^t$. Then

$$\max_{\delta\in\Delta(\tilde{\mathbf{d}},\gamma)} \mathcal{L}(\hat{\mathbf{p}},\delta;\mathcal{I}) \;\le\; \min_{\mathbf{p}\in\mathfrak{P}(\alpha,\beta)} \max_{\delta\in\Delta(\tilde{\mathbf{d}},\gamma)} \mathcal{L}(\mathbf{p},\delta;\mathcal{I}) \;+\; O\Big(\sqrt{\tfrac{n}{T}}\Big). \tag{15}$$

Setup. Recall that $\delta^t \in \arg\max_{\delta\in\Delta(\tilde{\mathbf{d}},\gamma)} \mathcal{L}(\mathbf{p}^t,\delta;\mathcal{I})$ for each round t (i.e., a best response to $\mathbf{p}^t$). Define the sequence of convex functions on $\mathfrak{P}(\alpha,\beta)$

$$f_t(\mathbf{p}) := \mathcal{L}(\mathbf{p},\delta^t;\mathcal{I}), \qquad t = 0,\dots,T-1.$$

By Proposition 7.1, each f_t is convex, and a subgradient $g_t \in \partial f_t(\mathbf{p}^t)$ is produced by Algorithm 3.

Projected subgradient regret. Let $\Pi_{\mathfrak{P}}$ be Euclidean projection onto $\mathfrak{P}(\alpha,\beta)$. The projected subgradient update is $\mathbf{p}^{t+1} = \Pi_{\mathfrak{P}}(\mathbf{p}^t - \eta g_t)$. For any $\mathbf{p}\in\mathfrak{P}(\alpha,\beta)$, the standard regret bound for projected subgradient descent (e.g., Zinkevich (2003)) yields

$$\sum_{t=0}^{T-1} f_t(\mathbf{p}^t) - \sum_{t=0}^{T-1} f_t(\mathbf{p}) \;\le\; \frac{\|\mathbf{p}-\mathbf{p}^0\|_2^2}{2\eta} \;+\; \frac{\eta}{2}\sum_{t=0}^{T-1}\|g_t\|_2^2. \tag{16}$$

Since $\mathfrak{P}(\alpha,\beta) = [\alpha,\beta]^n$, its Euclidean diameter is at most $D := \sqrt{n}\,(\beta-\alpha) \le \sqrt{n}$, whence $\|\mathbf{p}-\mathbf{p}^0\|_2 \le D$ for all $\mathbf{p}$. By the subgradient bound above, $\|g_t\|_2 \le G$ with $G \le \sqrt{n}$.

Choosing the optimal stepsize $\eta = D/(G\sqrt{T})$ in equation 16 and dividing by T yields

$$\frac{1}{T}\sum_{t=0}^{T-1} f_t(\mathbf{p}^t) - \frac{1}{T}\sum_{t=0}^{T-1} f_t(\mathbf{p}) \;\le\; \frac{DG}{\sqrt{T}} \;=\; O\Big(\sqrt{\tfrac{n}{T}}\Big). \tag{17}$$

From regret to saddle-point suboptimality. Let $\mathbf{p}^\star \in \arg\min_{\mathbf{p}\in\mathfrak{P}(\alpha,\beta)} \max_{\delta\in\Delta(\tilde{\mathbf{d}},\gamma)} \mathcal{L}(\mathbf{p},\delta;\mathcal{I})$ and denote the game value by v^* (recall the max min value of the game is equal to the min max value of the game by Sion's minimax theorem). Using $\mathbf{p} = \mathbf{p}^\star$ in equation 17 gives

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathcal{L}(\mathbf{p}^t,\delta^t;\mathcal{I}) \;\le\; \frac{1}{T}\sum_{t=0}^{T-1} \mathcal{L}(\mathbf{p}^\star,\delta^t;\mathcal{I}) \;+\; O\Big(\sqrt{\tfrac{n}{T}}\Big). \tag{18}$$

By definition of best response, $\mathcal{L}(\mathbf{p}^t,\delta^t;\mathcal{I}) = \max_\delta \mathcal{L}(\mathbf{p}^t,\delta;\mathcal{I})$. By convexity of $\mathbf{p}\mapsto\mathcal{L}(\mathbf{p},\delta;\mathcal{I})$ and Jensen's inequality,

$$\max_\delta \mathcal{L}(\hat{\mathbf{p}},\delta;\mathcal{I}) \;=\; \max_\delta \mathcal{L}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{p}^t,\delta;\mathcal{I}\right) \;\le\; \frac{1}{T}\sum_{t=0}^{T-1}\max_\delta \mathcal{L}(\mathbf{p}^t,\delta;\mathcal{I}) \;=\; \frac{1}{T}\sum_{t=0}^{T-1}\mathcal{L}(\mathbf{p}^t,\delta^t;\mathcal{I}). \tag{19}$$

Combining equation 18 and equation 19,

$$\max_\delta \mathcal{L}(\hat{\mathbf{p}},\delta;\mathcal{I}) \;\le\; \frac{1}{T}\sum_{t=0}^{T-1}\mathcal{L}(\mathbf{p}^\star,\delta^t;\mathcal{I}) \;+\; O\Big(\sqrt{\tfrac{n}{T}}\Big) \;\le\; \max_\delta \mathcal{L}(\mathbf{p}^\star,\delta;\mathcal{I}) \;+\; O\Big(\sqrt{\tfrac{n}{T}}\Big) \;=\; v^* + O\Big(\sqrt{\tfrac{n}{T}}\Big). \tag{20}$$

This is exactly the claim in Theorem 3.1.

8 Supplemental Materials from Section 4

8.1 Dependent Rounding: Negative Association (NA) and Concentration

We gather concentration tools used in the representation guarantees.

Lemma 8.1 (Negative association, marginal preservation of Pipage rounding) *Let $\tilde{K} \subseteq N$ be the random panel produced by Pipage rounding applied to $\mathbf{p} \in [0,1]^n$. Then the indicator variables $X_i := \mathbb{I}(i \in \tilde{K})$ are* negatively associated *(NA). In particular, for any two disjoint index sets $A, B \subseteq N$ and coordinatewise nondecreasing functions f, g, $\mathbb{E}[f((X_i)_{i\in A})\, g((X_i)_{i\in B})] \le \mathbb{E}[f((X_i)_{i\in A})]\,\mathbb{E}[g((X_i)_{i\in B})]$. Moreover, $\mathbb{E}[X_i] = p_i$ for all i.*

Proof sketch. See Gandhi et al. (2006) for result that Pipage rounding preserves marginals and induces NA. □

Lemma 8.2 (Hoeffding bound for NA variables - see, e.g., (Dubhashi & Ranjan, 1996)) *For generic set S, let $(X_i)_{i\in S}$ be NA, $X_i \in \{0,1\}$ with $\mathbb{E}[X_i] = p_i$. For fixed weights $a_i \in [0,1]$, define $Y = \sum_{i\in S} a_i X_i$ and $\mu = \mathbb{E}[Y] = \sum a_i p_i$. Then for all $z > 0$,*

$$\Pr[\,Y - \mu \ge z\,] \le \exp\!\left(-\frac{2z^2}{\sum_{i\in S} a_i^2}\right), \qquad \Pr[\,\mu - Y \ge z\,] \le \exp\!\left(-\frac{2z^2}{\sum_{i\in S} a_i^2}\right).$$

8.2 Proof of Theorem 4.1

Theorem 8.1 (Affine expected-quota concentration for MinMax-Pipage) *Fix inputs $\mathcal{I}, \tilde{\mathbf{d}}, \gamma, \alpha, \beta, \eta, T$, and let $\tilde{K}$ be the (random) panel produced by* MinMax-Pipage. *Fix any $\delta \in \Delta(\tilde{\mathbf{d}}, \gamma)$, any $(f, v) \in FV$, and any $z > 0$. with probability at least $1 - 2\exp\left(-\frac{2z^2}{\sigma_{f,v}^2}\right)$,*

$$\max_{\dagger\in\{-,+\}} \mathbb{E}_{D\sim\delta}\left[\mathrm{dev}^{\dagger}_{f,v}(\tilde{K}; D)\right] \le \mathcal{L}(\mathbf{p}^*, \delta; \mathcal{I}) + \epsilon + \tfrac{z}{u_{f,v}},$$

where $\sigma_{f,v}^2 = \sum_{i\in N_{f,v}} \Pr_{D\sim\delta}[i \notin D]^2$, $\epsilon = O\left(\sqrt{\frac{n}{T}}\right)$

Proof. Let $X_i := \mathbb{I}(i \in \tilde{K}) \in \{0,1\}$ be the indicator variables produced by Pipage rounding on $\hat{\mathbf{p}}$. By the properties of Pipage (Lemma 8.1): (i) $\mathbb{E}[X_i] = \hat{p}_i$ (marginals preserved), and (ii) the family $(X_i)_{i\in N}$ is negatively associated (NA).

Fix (f, v). For any $D \subseteq N$,

$$S_{f,v}(\tilde{K}; D) = \sum_{i\in N_{f,v}} X_i\, \mathbb{I}(i \notin D), \qquad S_{f,v}(\hat{\mathbf{p}}; D) = \sum_{i\in N_{f,v}} \hat{p}_i\, \mathbb{I}(i \notin D).$$

Taking expectation over $D \sim \delta$ (independent of $\tilde{K}$) and using $a_i = \Pr[i \notin D]$,

$$\mathbb{E}_D[S_{f,v}(\tilde{K}; D) \mid \tilde{K}] = \sum_{i\in N_{f,v}} a_i X_i =: Y_{f,v}(X), \qquad \mathbb{E}_D[S_{f,v}(\hat{\mathbf{p}}; D)] = \sum_{i\in N_{f,v}} a_i \hat{p}_i =: \mu_{f,v}.$$

Since (X_i) are NA 0–1 and the weights $a_i \in [0,1]$, the NA weighted Hoeffding inequality (Lemma 8.2) applied to the linear form $Y_{f,v}(X) = \sum_{i\in N_{f,v}} a_i X_i$ with variance proxy $\sigma_{f,v}^2 = \sum_{i\in N_{f,v}} a_i^2$ yields, for any $z > 0$,

$$\Pr\left[\ |Y_{f,v}(X) - \mu_{f,v}| \ge z\ \right] \ \le\ 2\exp\!\Big(-\tfrac{2z^2}{\sigma_{f,v}^2}\Big).$$

Therefore, by the definitions of $\mathrm{dev}^-_{f,v}$ and $\mathrm{dev}^+_{f,v}$,

$$\mathbb{E}_D\left[\mathrm{dev}^-_{f,v}(\tilde{K})\right] - \mathbb{E}_D\left[\mathrm{dev}^-_{f,v}(\hat{\mathbf{p}})\right] = \mathbb{E}_D\left[\tfrac{\ell_{f,v} - S_{f,v}(\tilde{K};D)}{u_{f,v}} - \tfrac{\ell_{f,v} - S_{f,v}(\hat{\mathbf{p}};D)}{u_{f,v}} \,\middle|\, \tilde{K}\right] = \tfrac{\mu_{f,v} - Y_{f,v}(X)}{u_{f,v}},$$

$$\mathbb{E}_D\left[\mathrm{dev}^+_{f,v}(\tilde{K})\right] - \mathbb{E}_D\left[\mathrm{dev}^+_{f,v}(\hat{\mathbf{p}})\right] = \mathbb{E}_D\left[\tfrac{S_{f,v}(\tilde{K};D) - u_{f,v}}{u_{f,v}} - \tfrac{S_{f,v}(\hat{\mathbf{p}};D) - u_{f,v}}{u_{f,v}} \,\middle|\, \tilde{K}\right] = \tfrac{Y_{f,v}(X) - \mu_{f,v}}{u_{f,v}}.$$

Hence both absolute deviations are equal to $|Y_{f,v}(X) - \mu_{f,v}|/u_{f,v}$.

By dividing both sides of $|Y_{f,v}(X) - \mu_{f,v}| \ge z$ by $u_{f,v}$ we get,

$$\left|\mathbb{E}_D\left[\mathrm{dev}^-_{f,v}(\tilde{K})\right] - \mathbb{E}_D\left[\mathrm{dev}^-_{f,v}(\hat{\mathbf{p}})\right]\right| \le \tfrac{z}{u_{f,v}}, \qquad \left|\mathbb{E}_D\left[\mathrm{dev}^+_{f,v}(\tilde{K})\right] - \mathbb{E}_D\left[\mathrm{dev}^+_{f,v}(\hat{\mathbf{p}})\right]\right| \le \tfrac{z}{u_{f,v}}.$$

Moreover, since $\mathbb{E}_D\left[\text{dev}^-_{f,v}(\hat{\mathbf{p}})\right] \leq \mathcal{L}(\hat{\mathbf{p}}, \delta; \mathcal{I})$ and $\mathbb{E}_D\left[\text{dev}^+_{f,v}(\hat{\mathbf{p}})\right] \leq \mathcal{L}(\hat{\mathbf{p}}, \delta; \mathcal{I})$], and Theorem 3.1 gives

$$\mathcal{L}(\hat{\mathbf{p}}, \delta; \mathcal{I}) \leq \max_{\delta \in \Delta(\tilde{\mathbf{d}}, \gamma)} \mathcal{L}(\mathbf{p}^*, \delta; \mathcal{I}) + O(\sqrt{n/T})$$

This is equivalently a bound on arbitrary $\delta \in \Delta(\tilde{\mathbf{d}}, \gamma)$, as the RHS can only grow with the removal of the maximum,

$$\implies \mathcal{L}(\hat{\mathbf{p}}, \delta; \mathcal{I}) \leq \mathcal{L}(\mathbf{p}^*, \delta; \mathcal{I}) + O(\sqrt{n/T})$$

Define $v^* := \mathcal{L}(\mathbf{p}^*, \delta; \mathcal{I})$. We can write the following equivalent probability statements and subsequently apply lemma 8.2:

$$\begin{aligned}\Pr\Big[\ |\mathbb{E}_D\left[\text{dev}^-_{f,v}(\tilde{K})\right] - \mathbb{E}_D\left[\text{dev}^-_{f,v}(\hat{\mathbf{p}})\right]| \leq \tfrac{z}{u_{f,v}}\ \Big] &= \Pr\Big[\mathbb{E}_D\left[\text{dev}^+_{f,v}(\tilde{K})\right] \ \leq\ \mathcal{L}(\mathbf{p}^*, \delta; \mathcal{I}) + O\big(\sqrt{n/T}\big) + \tfrac{z}{u_{f,v}}\Big] \\ &\geq 1 - 2\exp\Big(-\tfrac{2z^2}{\sigma^2_{f,v}}\Big)\end{aligned}$$

$$\begin{aligned}\Pr\Big[\ |\mathbb{E}_D\left[\text{dev}^+_{f,v}(\tilde{K})\right] - \mathbb{E}_D\left[\text{dev}^+_{f,v}(\hat{\mathbf{p}})\right]| \leq \tfrac{z}{u_{f,v}}\Big] &= \Pr\Big[\mathbb{E}_D\left[\text{dev}^+_{f,v}(\tilde{K})\right] \ \leq\ \mathcal{L}(\mathbf{p}^*, \delta; \mathcal{I}) + O\big(\sqrt{n/T}\big) + \tfrac{z}{u_{f,v}}\Big] \\ &\geq 1 - 2\exp\Big(-\tfrac{2z^2}{\sigma^2_{f,v}}\Big)\end{aligned}$$

Combining these statements and taking a maximum over the direction $\dagger \in \{+, -\}$, we arrive at the theorem statement for any $\delta \in \Delta(\tilde{\mathbf{d}}, \gamma)$, any $(f, v) \in FV$, and any $z > 0$:

$$\Pr\left[\max_{\dagger \in \{-,+\}} \mathbb{E}_{D\sim\delta}\left[\text{dev}^\dagger_{f,v}(\tilde{K}; D)\right] \leq \mathcal{L}(\mathbf{p}^*, \delta; \mathcal{I}) + O\left(\sqrt{\frac{n}{T}}\right) + \tfrac{z}{u_{f,v}}\right] \geq 1 - 2\exp\Big(-\tfrac{2z^2}{\sigma^2_{f,v}}\Big),$$

□

Remark: The rounding procedure is deterministically sum-preserving, so if one has constrained $\|\hat{\mathbf{p}}\|_1 = k^+$, $\tilde{K}$ is size k^+ deterministically, and $\mathbb{E}_{D\sim\delta}[|\tilde{K} \setminus D|] = k^+ - \mathbb{E}_{D\sim\delta}[|D \cap \tilde{K}|]$.

9 Supplemental Materials from Section 5

9.1 Data Information

Our data consists of six instances of previously occurred panels. These datasets include various demographic features of pool members, such as housing status, disability status, ancestry, language, and individuals' roles in their locality. Each instance has columns for the individual features, inclusion in the panel (SELECTED) and whether they dropped out(IS_DROPOUT) if they *were* selected. Instances 1,3,5,6 had predetermined, practitioner-constructed quotas. Both Instance 2 and 4 did not have practitioner constructed quotas, and we manually constructed a set to reflect the makeup of the *original panel*. F each feature-value pair, we counted the number of people selected for that panel with that feature-value pair ($n_{f,v}$), and set the associated quota to be $(\ell_{f,v}, u_{f,v}) = n_{f,v} \pm 2$. In the table below, we report statistics on the size of the pool n, desired size of the panel k and number of distinct feature value pairs $|FV|$.

$\mathcal{I}$	n	k	$\lvert FV\rvert$
1	250	51	17
2	125	52	12
3	133	41	22
4	290	50	12
5	518	47	11
6	427	49	28

9.2 Greedy Best Response over Product Distributions

We formally define the modified action space from section 5 below. The key alteration being is the change from a distribution over dropout sets $D \subseteq N$ to a distribution over marginal inclusion probabilities $\mathbf{d} \in [0,1]^n$. This represents a strict subset of the original action space in our theoretical formulation, and consequently, the best response on this more constrained set is only an *approximation* to the best response problem over **all** distributions over subsets. The gain is that the algorithm is substantially more efficient to run compared to our exact Ellipsoid-based best response algorithm, and still leads to strong experimental results.

The (new) action space of the maximizer. As in Assos et al. (2025), we model the dropouts from any panel $\tilde{K} \subseteq N$ as the result of independent Bernoulli coin-flips per agent $i \in \tilde{K}$, with agent-specific rate d_i. d_i is then i's *dropout probability*. For mathematical convenience, we assume that every *pool member* has an associated dropout probability, summarized as $\mathbf{d} = (d_i | i \in N)$. We define the *dropout set* $D \subset \tilde{K}$ as the random set of dropouts; we abuse notation slightly and express this random process as $D \sim \mathbf{d}$.

Given inputs $\tilde{\mathbf{d}}$ and γ, the maximizer's action space is

$$\mathfrak{D}(\tilde{\mathbf{d}}, \gamma) := [0,1]^n \ \cap \ B_\gamma^\infty(\tilde{\mathbf{d}}) \ \cap \ \{\mathbf{d} : \|\mathbf{d}\|_1 = \|\tilde{\mathbf{d}}\|_1\}, \tag{21}$$

where $B_\gamma^\infty(\tilde{\mathbf{d}})$ is the γ-radius L_∞-norm ball around $\tilde{\mathbf{d}}$. The first set requires $\mathbf{d}$'s entries to be valid probabilities; the second is the requirement specified by our notion of robustness; and the third is a weak requirement to avoid the adversary gaining power by dropping out a different expected number of *pool members* (not panelists) than would $\tilde{\mathbf{d}}$.

We present an updated loss function of $\mathbf{p}$ on dropout distribution $\mathbf{d}$ and the associated optimization problem:

$$\mathcal{L}(\mathbf{p}, \mathbf{d}; \mathcal{I}) := \max_{f,v \in FV} \mathbb{E}_{D \sim \mathbf{d}}[\mathcal{L}_{f,v}(\mathbf{p} \setminus D; \mathcal{I})] \tag{22}$$

$$\mathbf{p}^*(\mathcal{I}, \tilde{\mathbf{d}}, \gamma, \alpha, \beta) := \arg \min_{\mathbf{p} \in \mathfrak{P}(\alpha,\beta)} \max_{\mathbf{d} \in \mathfrak{D}(\tilde{\mathbf{d}},\gamma)} \mathcal{L}(\mathbf{p}, \mathbf{d}; \mathcal{I}) \tag{23}$$

This new setup allows for a simple, greedy maximizer best response algorithm. The algorithm works as follows: for each feature value pair f, v, the algorithm calculates the result of *dropping out* the maximum possible number of pool members with f, v, or conversely *retaining* the maximal number of pool members with f, v, each

time projecting results with PROJECT-DROPOUT-PROBS. The best response to the panel probabilities $\mathbf{p}$ is the argument maximum of the loss-maximizing $\mathbf{d}$ for each f, v pair and upper and lower quota violation.

Algorithm 4 GREEDY-BEST-RESPONSE

Require: $\mathcal{I}$, $\tilde{\mathbf{d}}$, γ, $\mathbf{p}$

$\mathbf{d}^{min} \leftarrow \max(0, \tilde{\mathbf{d}} - \gamma)$, $\mathbf{d}^{max} \leftarrow \min(1, \tilde{\mathbf{d}} + \gamma)$
for each $f, v \in FV$ **do**
  Initialize $N_{f,v} \leftarrow \{i : f(i) = v\}$ sorted in ascending order of $p(i)$, $\overline{N}_{f,v} \leftarrow N \setminus N_{f,v}$,
  $\mathbf{d}^{\text{keep}} \leftarrow \mathbf{0}$, $\mathbf{d}^{\text{drop}} \leftarrow \mathbf{0}$
  $d_i^{\text{keep}} \leftarrow d_i^{\min}$ for all $i \in N_{f,v}$ ▷ **Strategy 1**: keep as many people with $f(i) = v$ as possible
  $d_i^{\text{keep}} \leftarrow d_i^{\max}$ for all $i \in \overline{N}_{f,v}$
  $\bar{\mathbf{d}}^{\text{keep}} \leftarrow$ PROJECT-DROPOUT-PROBS$(\mathcal{I}, \tilde{\mathbf{d}}, \mathbf{d}^{\text{keep}}, \mathbf{d}^{min}, \mathbf{d}^{max}, \overline{N}_{f,v} + N_{f,v})$
  $d_i^{\text{drop}} \leftarrow d_i^{\max}$ for all $i \in N_{f,v}$ ▷ **Strategy 2**: drop as many people with $f(i) = v$ as possible
  $d_i^{\text{drop}} \leftarrow d_i^{\min}$ for all $i \in N_{f,v}$
  $\bar{\mathbf{d}}^{\text{drop}} \leftarrow$ PROJECT-DROPOUT-PROBS$(\mathcal{I}, \tilde{\mathbf{d}}, \mathbf{d}^{\text{drop}}, \mathbf{d}^{min}, \mathbf{d}^{max}, \overline{N}_{f,v} + N_{f,v})$
  $\mathbf{d}^*_{f,v} \leftarrow \text{argmax}_{\mathbf{d} \in \{\bar{\mathbf{d}}^{\text{keep}}, \bar{\mathbf{d}}^{\text{drop}}\}} \mathcal{L}(\mathbf{p}, \mathbf{d})$
end for
return $\text{argmax}_{f,v \in FV} \mathbf{d}^*_{f,v}$

Algorithm 5 PROJECT-DROPOUT-PROBS

Require: $\mathcal{I}, \tilde{\mathbf{d}}, \mathbf{d}^{\text{keep}}, \mathbf{d}^{min}, \mathbf{d}^{max}, \overline{N}_{f,v} + N_{f,v}$

$\text{diff} \leftarrow \sum_{i=1}^n \tilde{\mathbf{d}}_i - \sum_{i=1}^n d_i$
if $\text{diff} = 0$ **then return d**
end if
for each $i \in \overline{N}_{f,v}$, then $N_{f,v}$ **do**
  if $\text{diff} < 0$ **then**
    $\text{adj} \leftarrow \min(\text{diff}, d_i - \mathbf{d}_i^{min})$
    $d_i \leftarrow d_i - \text{adj}$
  else
    $\text{adj} \leftarrow \min(-\text{diff}, \mathbf{d}_i^{max} - d_i)$
    $d_i \leftarrow d_i + \text{adj}$
  end if
  if $\sum_{i=1}^n \tilde{\mathbf{d}}_i - \sum_{i=1}^n d_i = 0$ **then break**
  end if
end for
return d

Theorem 9.1 (Runtime of Greedy-Best-Response) *Algorithm 4 (*GREEDY-BEST-RESPONSE*) runs in time* $O(|\text{FV}| \cdot n)$.

Proof. For each feature–value pair $(f, v) \in \text{FV}$, the algorithm:

1. Constructs candidate dropout vectors $\mathbf{d}^{\text{keep}}$ and $\mathbf{d}^{\text{drop}}$ in $O(n)$ time;
2. Projects each via PROJECT-DROPOUT-PROBS, which iterates once through all agents and hence also runs in $O(n)$ time;
3. Evaluates the loss function $\mathcal{L}(\mathbf{p}, \mathbf{d})$ twice, costing $O(n)$ total.

Thus, the total work per feature–value pair is $O(n)$, and iterating over all $|\text{FV}|$ such pairs gives $O(|\text{FV}| \cdot n)$ overall. □

Theorem 9.2 (Correctness of Greedy-Best-Response) *Fix $\mathcal{I}, \gamma, \tilde{\mathbf{d}}$ and selection probabilities $\mathbf{p}$. Algorithm 4 returns $\mathbf{d}$ such that*

$$\mathbf{d} = \arg\max_{\mathbf{d} \in \mathfrak{D}(\tilde{\mathbf{d}}, \gamma)} \mathcal{L}(\mathbf{p}, \mathbf{d}\, \mathcal{I})$$

.

Proof. We prove that the GREEDY-BEST-RESPONSE algorithm (with the projection subroutine run using the index order "complement first, then group members in ascending order of p_i") returns a global maximizer of

$$\mathcal{L}(\mathbf{p}, \mathbf{d}) \;=\; \max_{(f,v) \in \mathrm{FV}} \max\left\{ \frac{\ell_{f,v} - S_{f,v}(\mathbf{p}, \mathbf{d})}{u_{f,v}},\; \frac{S_{f,v}(\mathbf{p}, \mathbf{d}) - u_{f,v}}{u_{f,v}} \right\},$$

where

$$S_{f,v}(\mathbf{p}, \mathbf{d}) \;=\; \sum_{i \in N_{f,v}} p_i (1 - d_i).$$

Let $\mathbf{d}^\star$ be the vector returned by the algorithm. Suppose, for contradiction, that there exists a feasible $\mathbf{d}' \in \mathfrak{D}(\hat{\mathbf{d}}, \gamma)$ with

$$\mathcal{L}(\mathbf{p}, \mathbf{d}') \;>\; \mathcal{L}(\mathbf{p}, \mathbf{d}^\star).$$

Then there must exist some group $g = (f, v)$ such that the group term

$$G_g(\mathbf{d}) := \max\left\{ \frac{\ell_g - S_g(\mathbf{d})}{u_g}, \frac{S_g(\mathbf{d}) - u_g}{u_g} \right\}$$

satisfies $G_g(\mathbf{d}') > G_g(\mathbf{d}^\star)$. We analyze the two exhaustive cases: overshooting the upper quota and undershooting the lower quota.

Preliminary: we give maximal slack to N_g. Because $G_g(\mathbf{d})$ depends only on coordinates $\{d_i : i \in N_g\}$, any feasible changes to coordinates outside N_g do not affect the group term. Therefore, without loss of generality, we may first modify $\mathbf{d}'$ on $\overline{N}_g$ to allocate as much allowable probability as possible there (respecting the box constraints) so that N_g is left with maximal slack and still consistent with feasibility. This is exactly the adjustment the algorithm performs when forming its candidates: it sets complement coordinates to their upper (or lower) bounds as needed before touching group members. Doing this cannot decrease $G_g(\mathbf{d}')$, and it simplifies the comparison to $\mathbf{d}^\star$ because any remaining difference between $\mathbf{d}'$ and $\mathbf{d}^\star$ will lie inside N_g.

Case 1: $\mathbf{d}'$ violates the upper quota more than $\mathbf{d}^\star$. That is,

$$G_g(\mathbf{d}') = \frac{S_g(\mathbf{d}') - u_g}{u_g} > G_g(\mathbf{d}^\star),$$

so

$$S_g(\mathbf{d}') \;>\; S_g(\mathbf{d}^\star), \quad \text{equivalently} \quad \sum_{i \in N_g} p_i d'_i \;<\; \sum_{i \in N_g} p_i d^\star_i.$$

Let $i^\star$ be an index in N_g with the smallest p_i for which $d'_{i^\star} < d^\star_{i^\star}$ (such an index exists because the weighted sum under $\mathbf{d}'$ is strictly smaller). By the preliminary reduction we may assume the complement coordinates in $\mathbf{d}'$ are already set to give maximal slack to N_g, so any further decrease of $d_{i^\star}$ would require compensating increases somewhere (either within N_g on other indices or in the complement if slack remained). But the algorithm's 'keep' construction already sets complement coordinates to their allowed maxima and sets group coordinates to their allowed minima before projection. Under the complement-first, ascending-p projection the algorithm only increases group coordinates in order of increasing p_i, so it produces the allocation that minimizes $\sum_{i \in N_g} p_i d_i$ (equivalently maximizes S_g) subject to feasibility. Hence no feasible $\mathbf{d}'$ can have strictly smaller $\sum_{i \in N_g} p_i d_i$ than $\mathbf{d}^\star$, contradicting the assumption.

Case 2: $\mathbf{d}'$ violates the lower quota more than $\mathbf{d}^\star$. That is,

$$G_g(\mathbf{d}') = \frac{\ell_g - S_g(\mathbf{d}')}{u_g} > G_g(\mathbf{d}^\star),$$

so

$$S_g(\mathbf{d}') \;<\; S_g(\mathbf{d}^\star), \quad \text{equivalently} \quad \sum_{i \in N_g} p_i d'_i \;>\; \sum_{i \in N_g} p_i d^\star_i.$$

Let $i^\star$ be the index in N_g with the smallest p_i for which $d'_{i^\star} > d^\star_{i^\star}$ (such an index exists because the weighted sum under $\mathbf{d}'$ is strictly larger). Again, by the preliminary reduction we may assume complement coordinates in $\mathbf{d}'$ are already adjusted to give minimal complement mass (maximal group mass) consistent with feasibility. The algorithm's 'drop' construction initializes group coordinates at their upper bounds and complement coordinates at their lower bounds, then uses the complement-first, ascending-p projection (which reduces the smallest-p group members first) to realize the allocation that maximizes $\sum_{i \in N_g} p_i d_i$ (equivalently minimizes S_g). Therefore no feasible $\mathbf{d}'$ can achieve a strictly larger $\sum_{i \in N_g} p_i d_i$ than $\mathbf{d}^\star$, a contradiction.

Both exhaustive cases lead to contradictions. Hence no feasible $\mathbf{d}' \in \mathfrak{D}(\hat{\mathbf{d}}, \gamma)$ has strictly larger loss than $\mathbf{d}^\star$, and $\mathbf{d}^\star$ is a global maximizer of $\mathcal{L}(\mathbf{p}, \cdot)$. □

9.3 Methods for prediction of $\tilde{\mathbf{d}}$

We construct $\tilde{\mathbf{d}}$ with a simple independent action model, calculating dropout probabilities as a function of two features that are shared across instances: age and gender.

Although these features are shared, the particular values are not consistent across instances. Thus, we preprocess the demographic panel data to ensure consistent feature representations across the multiple datasets. We standardize the `age` and `gender` features and learn dropout probabilities with these features and their second order co-occurrences.

Age Standardization Age data were reported in across datasets, using various age ranges.

To standardize ages across datasets, we first converted reported age ranges into single numeric ages by sampling uniformly within the reported range.

For an age range "a–b", we sampled an integer uniformly from $[a, b]$ (inclusive of a. For open-ended ranges like "75+", we sampled uniformly between 75 and 85 as a reasonable upper bound. For single numeric ages, we used the reported value directly.

After assigning numeric ages, participants were bucketed into consistent age groups.

Gender Standardization Gender entries were also partially heterogeneous across datasets. We standardized gender into four categories:

- **Female**: entries such as "female", "woman", "girl", "f".
- **Male**: entries such as "male", "man", "boy", "m".
- **Nonbinary**: entries such as "non-binary", "nonbinary"
- **Other**: all remaining or missing entries.

Learning Dropout Probabilities We model the probability that a selected participant drops out of a panel using multiplicative beta parameters associated with individual features and their interactions.

Feature tuples: For a chosen feature list (e.g., `age`, `gender`), we generate all unique tuples of features and values to account for second-order interactions). This defines new features and values of the form $f', v' = (f^1 - f^2, v^1 - v^2)$. For each panel instance, we collect all unique value combinations for each feature tuple among selected participants, and we learn beta parameters on each of them.

Likelihood function: For each participant, we compute the probability a person drops out as the complement of the probability they *stay* in the panel as

$$1 - \tilde{\mathbf{d}}_i = \Pr(\text{person } i \text{ stays in}) = \beta_0 \prod_{f \in F} \beta_{f,f(i)},$$

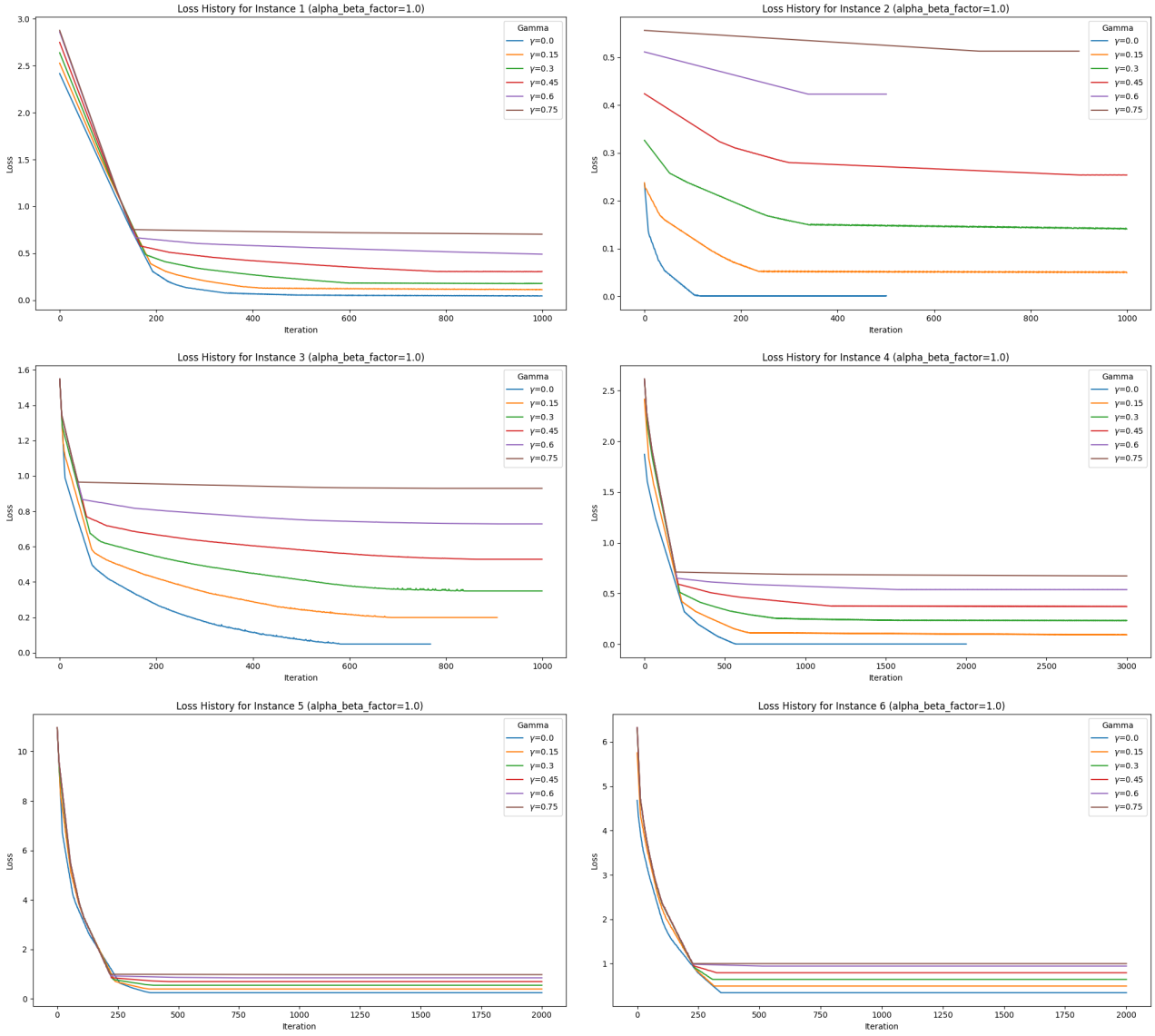
where β_0 is a base probability, F is the set of feature tuples, and $v = f(i)$ is the participant's value for feature or feature tuple f. These parameters are estimated via MLE, with the implementation matching that of Assos et al. (2025).

We provide summary statistics of our estimated dropout probabilities for each instance i below. For each instance i, we find the $\tilde{\mathbf{d}}$ by building a model with parameters derived from the features, values and dropouts of all instances *except* instance i .

Instance	Min	Max	Mean	Std Dev
Instance 1	0.0004	0.2778	0.1157	0.0856
Instance 2	0.0003	0.3750	0.1433	0.1351
Instance 3	0.0004	0.6769	0.1241	0.1116
Instance 4	0.0004	0.3662	0.1874	0.1115
Instance 5	0.0004	0.4616	0.2115	0.0959
Instance 6	0.0004	0.4286	0.1334	0.1022

9.4 Convergence Results

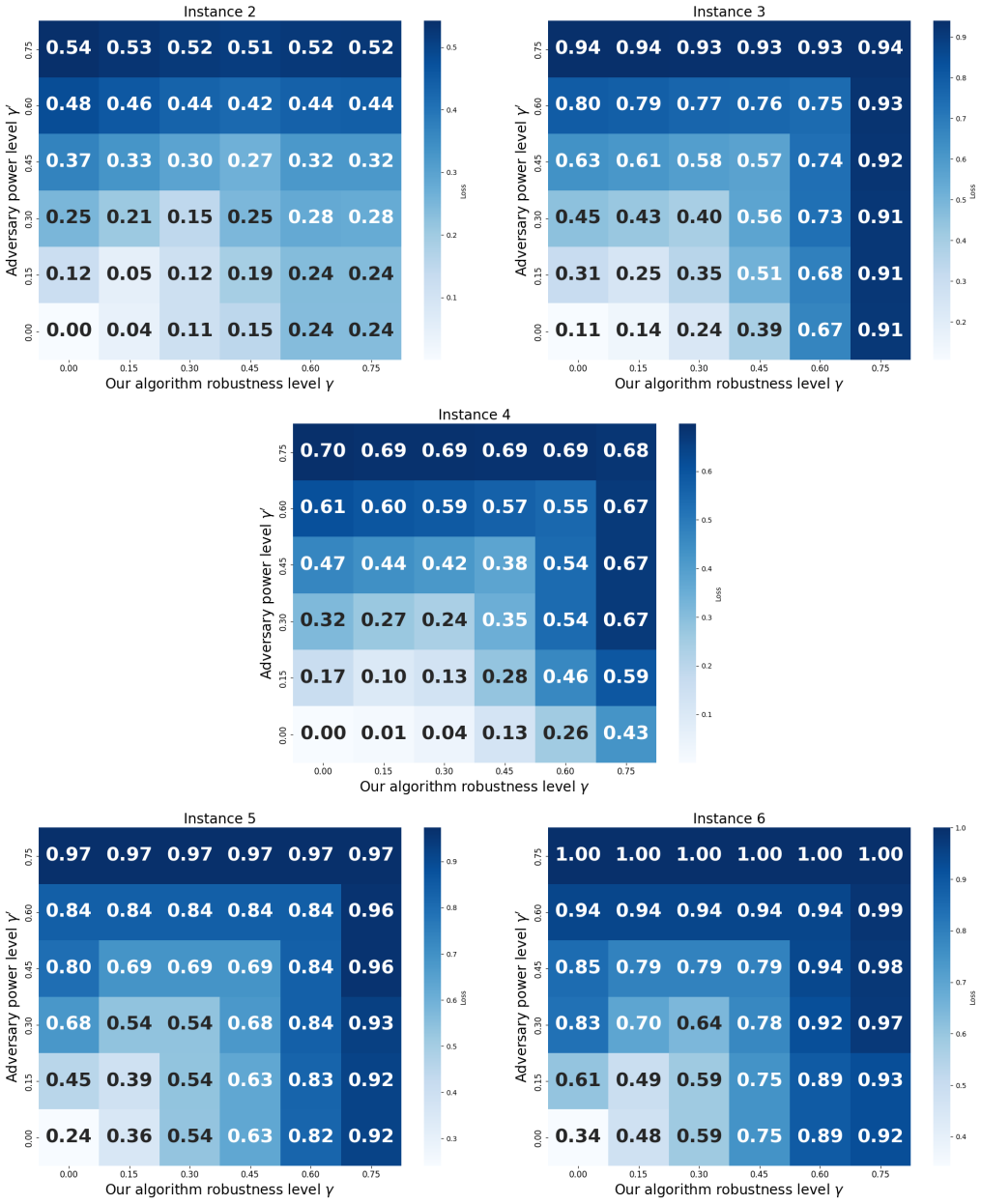
We provide plots demonstrating convergence to a stable loss value over rounds below.



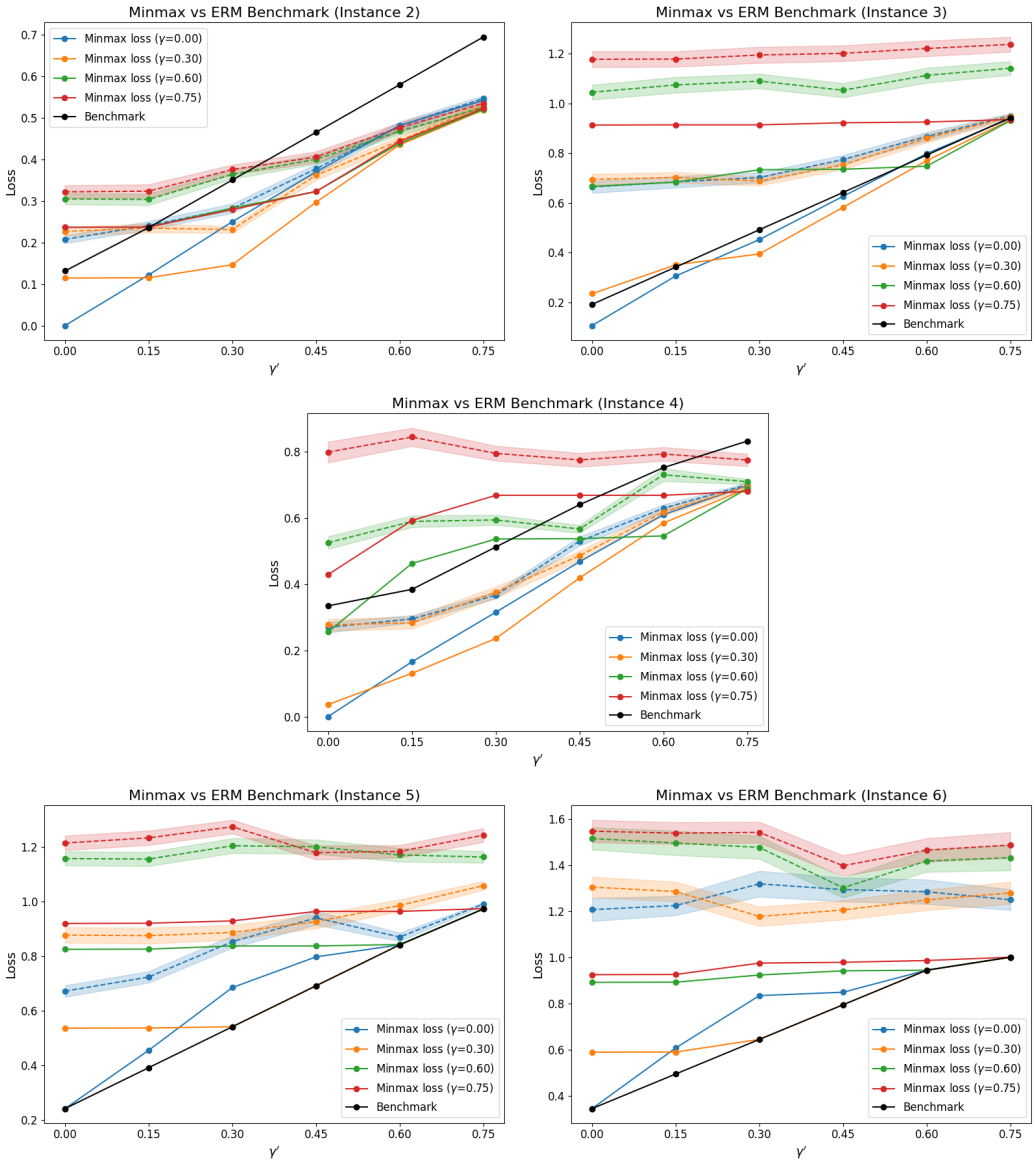
9.5 Implementation of ERM-Benchmark

We make two key alterations to the ERM algorithm presented in main text of Assos et al. (2025). First, we make use of a different formulation of the alternates problem, which is presented in Appendix F.3 of Assos et al. (2025). This formulation selects an *entire panel* instead of alternates. We alter the objective slightly, instead optimizing on our loss function, rather than the loss function from the paper - for consistency. While the core contribution of the paper, as outlined in section 1, selects a set of alternates from which replacements can be drawn, this does not exactly match our work. Instead, we consider the extension of the algorithm in which ERM optimization is used to select a maximally robust panel from the start, parallel to our approach. Assos et al. also measures loss as *either* the *sum* of all (scaled) deviation over all feature values, or a binary measure of whether all quotas were achieved. We instead optimize and evaluate on our loss function from section 2, which measures the *maximum* scaled deviation over any feature value. Both of these choices are in an effort to make the algorithms more comparable in both approach and evaluation.

9.6 Remaining instances: Figure 1



9.7 Remaining instances: Figure 3



9.8 Selection probabilities of ERM-Benchmark

Although ERM-Benchmark returns a deterministic panel of people, it does randomize over identical pool members with equivalent feature value pairs. This makes intuitive sense, as it makes no difference to the objective if we replace pool member with one that is demographically identical. Thus, we calculate the selection probability of each panelist as:

$$\Pr[\text{Pool member } i \text{ is selected for the panel}] = \frac{\sum_{j=1}^{n} \mathbf{1}[f(j) = f(i) \quad \forall f \in F]}{\text{n}}$$

i.e. the proportion of the pool that is people with identical feature-value composition to person i.