# Putting a Compass on the Map of Elections

Niclas Boehmer, Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, and Stanisław Szufa

**Abstract**

Recently, Szufa et al. [AAMAS 2020] presented a "map of elections" that visualizes a set of 800 elections generated from various statistical cultures. While similar elections are grouped together on this map, there is no obvious interpretation of the elections' positions. We provide such an interpretation by introducing four canonical "extreme" elections, acting as a compass on the map. We use them to analyze both a dataset provided by Szufa et al. and a number of real-life elections. In effect, we find a new variant of the Mallows model and show that it captures real-life scenarios particularly well.

## 1 Introduction

Szufa et al. [21] recently proposed a technique for visualizing sets of ordinal elections—i.e., elections where each voter ranks the candidates from the most to the least appealing one—based on given distances between them. They have applied this technique to 800 elections coming from various statistical cultures, ranging from the classic urn and Mallows models to various types of restricted domains, and they obtained a *map of elections*, where elections with similar properties are grouped together (see Figure 1; each dot represents a single election and, generally, the closer two elections are in the picture, the smaller is their distance in terms of the metric of Szufa et al.). Indeed, we see that elections from the same statistical culture, represented with the same color, are nicely grouped together; Szufa et al. [21] have also shown other evidence that nearby elections are closely related.[1] Yet, the map has two major drawbacks. First, while similar elections are plotted next to each other, there is no clear meaning to absolute positions on the map. Second, the map regards statistical cultures only and it is not obvious where real-life elections—such as those stored in PrefLib [15]—would lie on the map. Our goal is to address both these issues.

We start by looking more closely at the distance metric for elections that Szufa et al. [21] used. The idea is that given an election with $m$ candidates, one computes an $m \times m$ *frequency matrix* which specifies what fraction of the voters ranks each candidate in each position (such matrices are bistochastic, i.e., their entries are nonnegative and each column and each row sums up to one). Measuring the distance between two elections boils down to computing their frequency matri-

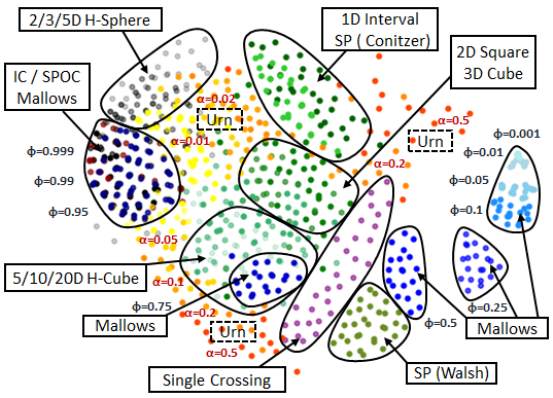
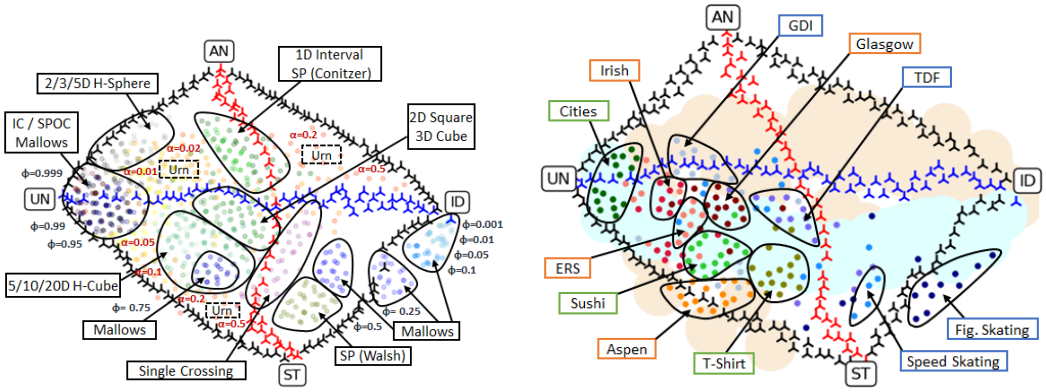



Figure 1: A map for the 10x100 dataset of Szufa et al. [21].

[1] The map in the figure regards elections with 10 candidates and 100 voters, whereas Szufa et al. [21] focused on the case of 100 candidates and 100 voters. Nonetheless, they also provided such smaller datasets and we focus on them because we want to compare them to real-life elections, which typically have few candidates.




(a) The compass and the dataset of Szufa et al. [21]. (b) The compass and various real-life elections.

Figure 2: Maps of elections for the 10x100 dataset of Szufa et al. [21] (on the left) and some real-life elections (on the right). The maps include the compass matrices and their connecting paths (shown in black, red, and blue). For clarity, the dots corresponding to the elections of Szufa et al. [21] (in the left figure) are shown in lighter colors than in Figure 1. On the right, the pale blue area is where Mallows elections end up (for various $\phi$ parameters) and the pale orange area is where urn elections end up (for various $\alpha$ parameters).

ces and summing up the earth mover's distances between their columns, where columns are reordered to minimize the distance (see Section 2); Szufa et al. [21] call this distance *positionwise*. Using frequency matrices makes it possible to compare elections with different numbers of voters (effectively, by reweighting them), and reordering the columns ensures that candidate names are irrelevant (as suggested for such settings by Faliszewski et al. [7]).

Our first result is an algorithm that, given a bistochastic matrix and a number $n$, finds *some* election with $n$ voters whose frequency matrix is (nearly) identical to the one from the input (achieving perfect accuracy is not always possible, but our algorithm achieves the best result one may, in general, hope for). As a consequence, instead of considering elections, we may directly look at the space of bistochastic matrices, which simplifies many discussions. Thus, we often speak of matrices and elections interchangeably.

Next we form what we call a *compass*. The idea is to pick matrices that, on the one hand, are far away from each other, and, on the other hand, have natural interpretations. Specifically, we consider the following four "extreme" matrices, corresponding to four types of (dis)agreement among the voters:

1. The identity matrix, ID, modelling perfect agreement (all voters agree on a single preference order).

2. The uniformity matrix, UN, modelling lack of agreement (each candidate takes each position equally often).

3. The stratification matrix, ST, modelling partial agreement (voters agree that half of the candidates are better than the other half, but lack agreement on anything else).

4. The antagonism matrix, AN, modelling conflict (half of the voters have opposite preference orders to the other half).

For each two of these "compass" matrices, we also consider a spectrum of their convex combinations ("paths" between the matrices). In visualizations, these paths appear as a parallelogram-like shape with corresponding "diagonals"; see, e.g., Figure 2a, where we

apply the compass method to the dataset from Figure 1 (the black, blue, and red points are certain convex combinations of the corresponding endpoints, which are the compass matrices). Throughout the rest of the paper we explain where these figures come from.

The compass allows us to make several observations. For example, in Figure 2a we see that 1D Interval elections are closer to the antagonism matrix, whereas higher-dimensional hypercube elections are closer to the stratification one. This is intriguing as, on a formal level, these two kinds of elections are very similar. Figure 2b, which shows a map of real-life elections (some from PrefLib and some new ones) is even more striking. Most of the real-life elections (including all political ones) end up in one "quadrant" of the parallelogram, and essentially all elections end up in the vicinity of some Mallows elections (in Figure 2b, the pale blue area is where Mallows elections end up, depending on the parameter of the model; the pale orange area is where urn elections end up). So, if one were to run experiments with a single statistical culture, the Mallows model might be a wise choice.

Yet, we find that natural ways of sampling Mallows elections (e.g., by choosing the Mallows parameter uniformly at random, or using a fixed parameter for different numbers of candidates), which are used in many research papers, are biased. We propose a normalization and argue, both theoretically and by considering the compass, that it produces more balanced results. In other words, we recommend using the Mallows model, but in conjunction with our normalization.

We provide a detailed analysis and discussion of the above-mentioned results in the main part of the paper and in the appendix (results marked by (★) are proven in the appendix.).

## 2 Preliminaries

Given an integer $t$, we write $[t]$ to denote the set $\{1, \dots, t\}$. By $\mathbb{R}_+$ we mean the set of nonnegative real numbers. Given a vector $x = (x_1, \dots, x_m)$, we interpret it as an $m \times 1$ matrix, i.e., we use column vectors. For a matrix $X$, we write $x_{i,j}$ to refer to the entry in its $i$-th row and $j$-th column.

**Elections.** An election $E$ is a pair $(C, V)$, where $C = \{c_1, \dots, c_m\}$ is a set of candidates and $V = (v_1, \dots, v_n)$ is a collection of voters. Each voter $v \in V$ has a preference order $\succ_v$, which ranks the candidates from the most to the least desirable one according to $v$. If $v$ prefers candidate $a$ to candidate $b$, then we write $v\colon a \succ b$, and we extend this notation to more candidates in a natural way. For a voter $v$ and a candidate $c$, we write $\mathrm{pos}_v(c)$ to denote the position on which $v$ ranks $c$ (the top-ranked candidate has position 1, the next one has position 2, and so on). We refer to both the voters and their preference orders as the votes. The intended meaning will always be clear from the context.

**Position and Frequency Matrices.** Let $E = (C, V)$ be an election, where $C = \{c_1, \dots, c_m\}$ and $V = (v_1, \dots, v_n)$. For a candidate $c \in C$ and position $i \in [m]$, we write $\#\mathrm{pos}_E(c, i)$ to denote the number of voters in election $E$ that rank $c$ on position $i$. By $\#\mathrm{pos}_E(c)$ we mean the vector:

$$(\#\mathrm{pos}_E(c, 1), \#\mathrm{pos}_E(c, 2), \dots, \#\mathrm{pos}_E(c, m)).$$

The *position matrix* for election $E$, denoted $\#\mathrm{pos}(E)$, is the $m \times m$ matrix that has vectors $\#\mathrm{pos}_E(c_1), \dots, \#\mathrm{pos}_E(c_m)$ as its columns. We also consider vote frequencies rather than absolute counts. To this end, for a candidate $c$ and a position $i \in [m]$, let $\#\mathrm{freq}_E(c, i)$ be $\frac{\#\mathrm{pos}_E(c,i)}{n}$, let vector $\#\mathrm{freq}_E(c)$ be $(\#\mathrm{freq}_E(c, 1), \dots, \#\mathrm{freq}_E(c, m))$, and let the *frequency matrix* for election $E$, denoted $\#\mathrm{freq}(E)$, consist of columns $\#\mathrm{freq}_E(c_1), \dots, \#\mathrm{freq}_E(c_m)$.

Note that in each position matrix, each row and each column sums up to the number of voters in the election. Similarly, in each frequency matrix, the rows and columns sum up to

one (such matrices are called bistochastic). For a positive integer $m$, we write $\mathcal{F}(m)$ [$\mathcal{P}(m)$] to denote the set of all $m \times m$ frequency [position] matrices.

**Example 1.** *Let $E = (C, V)$ be an election, where $C = \{a, b, c\}$, $V = (v_1, \ldots, v_6)$, and the preference orders are $v_1 \colon a \succ b \succ c$, $v_2 \colon a \succ b \succ c$, $v_3 \colon a \succ b \succ c$, $v_4 \colon b \succ a \succ c$, $v_5 \colon c \succ a \succ b$, $v_6 \colon c \succ a \succ b$. The position and frequency matrices for this election are:*

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 3 & 1 & 2 \\ 3 & 3 & 0 \\ 0 & 2 & 4 \end{bmatrix} \end{array} \quad \textit{and} \quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 1/2 & 1/6 & 1/3 \\ 1/2 & 1/2 & 0 \\ 0 & 1/3 & 2/3 \end{bmatrix} \end{array}$$

**Earth Mover's Distance (EMD).** Let $x = (x_1, \ldots, x_t)$ and $y = (y_1, \ldots, y_t)$ be two vectors from $\mathbb{R}^t_+$, whose entries sum up to 1. The *earth mover's distance* between $x$ and $y$, denoted $\mathrm{EMD}(x, y)$, is defined as the lowest total cost of operations that transform vector $x$ into vector $y$, where each operation is of the form "*subtract $\delta$ from position $i$ and add $\delta$ to position $j$*" and costs $\delta \cdot |i - j|$. Such an operation is legal if the current value at position $i$ is at least $\delta$. $\mathrm{EMD}(x, y)$ can be computed in polynomial time using a greedy algorithm.

**Positionwise Distance.** Let $E = (C, V)$ and $F = (D, U)$ be two elections with $m$ candidates each (we do not require that $|V| = |U|$). The positionwise distance between $E$ and $F$, denoted $\mathrm{POS}(E, F)$, is defined in terms of frequency matrices $\#\mathrm{freq}(E) = (e_1, \ldots, e_m)$ and $\#\mathrm{freq}(F) = (f_1, \ldots, f_m)$ as follows [21]:

$$\mathrm{POS}(E, F) := \min_{\sigma \in S_m} \left( \sum_{i=1}^{m} \mathrm{EMD}(e_i, f_{\sigma(i)}) \right),$$

where $S_m$ is the permutation group on $m$ elements. In other words, the positionwise distance is the sum of the earth mover's distances between the frequency vectors of the candidates from the two elections, with candidates/columns matched optimally according to $\sigma$. The positionwise distance is invariant to renaming the candidates and reordering the voters.

**Statistical Cultures.** We define the following three statistical cultures, i.e., models for generating random elections:

1. Under the Impartial Culture (IC) model, we sample all votes uniformly at random.
2. The Pólya-Eggenberger urn model [2] uses a nonnegative parameter $\alpha$, which gives the level of correlation between the votes (this parameterization is due to McCabe-Dansted and Slinko [16]). To generate an election with $m$ candidates, we take an urn containing one copy of each possible preference order and generate the votes iteratively: In each step we draw an order from the urn (this is the newly generated vote) and return it to the urn together with $\alpha m!$ copies. For $\alpha = 0$, we obtain the IC model.
3. The Mallows model [14] uses parameter $\phi \in [0, 1]$ and a central preference order $v$. Each vote is generated randomly and independently. The probability of generating a vote $u$ is proportional to $\phi^{\kappa(u,v)}$, where $\kappa(u, v)$ is the swap distance between $u$ and $v$ (i.e., the minimum number of swaps of adjacent candidates that transform $u$ into $v$).

Sometimes, we refer to other statistical cultures used by Szufa et al. [21]. We do not define them formally here, but we attempt to make our discussions intuitively clear.

**Maps of Elections.** Szufa et al. [21] drew a *map of elections* by computing the positionwise distances between 800 elections drawn from various statistical cultures and visualizing them using the force-directed algorithm of Fruchterman and Reingold [8]. They focused on elections with 100 candidates and 100 voters, but also generated smaller datasets, available

on their website. We consider their dataset with 10 candidates and 100 voters (see Figure 1 for its map). We use the same algorithm as they do for our visualizations, except that for each two elections we set their attraction coefficient to be the square of their positionwise distance (and not the distance itself, as they do; our approach groups similar elections more tightly and gives more visually appealing results for elections with few candidates).

We stress that the maps that both we and Szufa et al. [21] provide are helpful tools to illustrate the distances between particular (families of) elections, but are certainly not perfect. For example, since the visualization algorithm is randomized, we can get slightly different maps for each run of the algorithm. The visualizations also depend on the exact composition of the set of depicted elections (for example, a map where 50% of the elections came from the IC model would make it seem that these elections cover a much larger proportion of the map than if IC elections constituted only 10% of the elections). Thus, whenever we say that some two elections are close to each other, we mean that their positionwise distance is small. While this is typically reflected by these two elections being close on the map, on its own, closeness on the map does not suffice for such a claim.

# 3 Recovering Elections from Matrices

Throughout this paper we often deal with frequency matrices of elections. While computing a frequency matrix of an election is straightforward, the reverse direction is less clear.

We first observe that each $m \times m$ position matrix has a corresponding $m$-candidate election with at most $m^2 - 2m + 2$ distinct preference orders. This was shown by Leep and Myerson [13, Theorem 7] (they speak of "semi-magic squares" and not "position matrices" and show a decomposition of a matrix into permutation matrices, which correspond to votes in our setting). Their proof lacks some algorithmic details which we provide in the appendix.

**Proposition 1 (★).** *Given a position matrix $X \in \mathcal{P}(m)$, one can compute in $O(m^{4.5})$ time an election $E$ that contains at most $m^2 - 2m + 2$ different votes such that $\#\mathrm{pos}(E) = X$.*

Next, we consider the issue of recovering elections based on frequency matrices. Given an $m \times m$ bistochastic matrix $X$ and a number $n$ of voters, we would like to find an election $E$ with position matrix $nX$. This may be impossible as $nX$ may have fractional entries, but we can get very close to this goal. The next proposition shows how to achieve it, and justifies speaking of elections and frequency matrices interchangeably.

**Proposition 2.** *Given an $m \times m$ bistochastic matrix $X$ and an integer $n$, one can compute in polynomial time an election $E$ with $n$ voters whose position matrix $P$ satisfies $|nx_{i,j} - p_{i,j}| \leq 1$ for each $i, j \in [m]$ and, under this condition, minimizes the value $\sum_{1 \leq i,j \leq m} |nx_{i,j} - p_{i,j}|$.*

*Proof.* We use randomized dependent rounding in the following algorithm; see Appendix A for a deterministic algorithm, which also performs the minimization step.

We start by computing matrix $Y$ where each entry $y_{i,j}$ is equal to $nx_{i,j} - \lfloor nx_{i,j} \rfloor$. All entries of $Y$ are between 0 and 1, and each row and each column of $Y$ sums up to a (possibly different) integer. We construct an edge-weighted bipartite graph $G$ with vertex sets $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_m\}$. For each two vertices $a_i$ and $b_j$, we have a connecting edge of weight $y_{i,j}$. For each vertex $c \in A \cup B$, we let its fractional degree $\delta_G(c)$ be the sum of the weights of the edges touching it. Then we invoke the dependent rounding procedure of Gandhi et al. [9] on this graph, and in polynomial time we obtain an unweighted bipartite graph $G'$ with the same two vertex sets, such that the (standard) degree of each vertex $c \in A \cup B$ in $G'$ is equal to $\delta_G(c)$ (note that dependent rounding is computed via a randomized algorithm, but this condition on the degrees is always satisfied, independently of the random bits selected). Using $G'$, we form an $m \times m$ matrix $D$ such that for each

$i, j \in [m]$, $d_{i,j}$ is 1 if $G'$ contains an edge between $i$ and $j$, and $d_{i,j} = 0$ otherwise. Finally, we compute matrix $P = \lfloor nX \rfloor + D$.

The entries of $P$ differ from those of $nX$ by at most one, and the rows and columns of $P$ sum up to $n$ (to see it, consider the degrees of the vertices in $G'$). So, we obtain the desired election by invoking Proposition 1 on matrix $P$. □

# 4 Setting up the Compass

Our "compass" consists of two main components: Four matrices that occupy very different areas of the election space and represent different types of (dis)agreement among the voters, and six paths consisting of their convex combinations. We describe these components below.

## 4.1 The Four Matrices

The first two matrices are the *identity* matrix, $\mathrm{ID}_m$, with ones on the diagonal and zeros elsewhere, and the *uniformity* matrix, $\mathrm{UN}_m$, with each entry equal to $1/m$. The identity matrix corresponds to elections where each voter has the same preference order, i.e., there is a common ordering of the candidates from the most to the least desirable one. In contrast, the uniformity matrix captures elections where each candidate is ranked on each position equally often, i.e., where, in aggregate, all the candidates are viewed as equally good. Uniformity elections are quite similar to the IC ones and, in the limit, indistinguishable from them. Yet, for a fixed number of voters, typically IC elections are at some (small) positionwise distance from uniformity.

The next matrix, *stratification*, is defined as follows (we assume that $m$ is even):

$$\mathrm{ST}_m = \begin{bmatrix} \mathrm{UN}_{m/2} & 0 \\ 0 & \mathrm{UN}_{m/2} \end{bmatrix}.$$

Stratification matrices correspond to elections where the voters agree that half of the candidates are more desirable than the other half, but, in aggregate, are unable to distinguish between the qualities of the candidates in each group.

For the next matrix, we need one more piece of notation. Let $\mathrm{rID}_m$ be the matrix obtained by reversing the order of the columns of the identity matrix $\mathrm{ID}_m$. We define the *antagonism* matrix, $\mathrm{AN}_m$, to be $1/2\mathrm{ID}_m + 1/2\mathrm{rID}_m$. Such matrices are generated, e.g., by elections where half of the voters rank the candidates in one order, and half of the voters rank them in the opposite one, so there is a clear conflict. In some sense, stratification and antagonism are based on similar premises. Under stratification, the group of candidates is partitioned into halves with different properties, whereas in antagonism (for the case where half of the voters rank the candidates in the same order) the voters are partitioned. However, the nature of the partitioning is, naturally, quite different.

We chose the above matrices because they capture natural, intuitive phenomena and seem to occupy very different areas of the space of elections. To see that the latter holds, let us calculate their positionwise distances (for further arguments see also Appendix B).

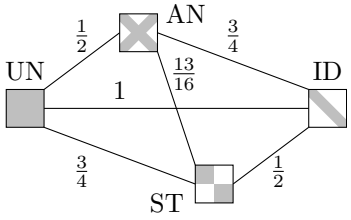
**Proposition 3** (★). *If $m$ is divisible by 4, then it holds that:*

1. $\mathrm{POS}(\mathrm{ID}_m, \mathrm{UN}_m) = \frac{1}{3}(m^2 - 1)$,
2. $\mathrm{POS}(\mathrm{ID}_m, \mathrm{AN}_m) = \mathrm{POS}(\mathrm{UN}_m, \mathrm{ST}_m) = \frac{m^2}{4}$,
3. $\mathrm{POS}(\mathrm{ID}_m, \mathrm{ST}_m) = \mathrm{POS}(\mathrm{UN}_m, \mathrm{AN}_m) = \frac{2}{3}(\frac{m^2}{4} - 1)$,

*4.* $\mathrm{POS}(\mathrm{AN}_m, \mathrm{ST}_m) = \frac{13}{48}m^2 - \frac{1}{3}$.

To normalize these distances, we divide them by $D(m) = \mathrm{POS}(\mathrm{ID}_m, \mathrm{UN}_m)$, which we suspect to be the largest positionwise distance between two matrices from $\mathcal{F}(m)$ (as argued below). For each two matrices $X$ and $Y$ among our four, we let $d(X,Y) := \lim_{m\to\infty} {}^{POS(X_m,Y_m)}/_{D(m)}$. A simple computation shows the following (see also the drawing on the right; we sometimes omit the subscript $m$ for simplicity):

$d(\mathrm{ID}, \mathrm{UN}) = 1,$
$d(\mathrm{ID}, \mathrm{AN}) = d(\mathrm{UN}, \mathrm{ST}) = {}^{3}/_{4},$
$d(\mathrm{AN}, \mathrm{ST}) = {}^{13}/_{16},$
$d(\mathrm{ID}, \mathrm{ST}) = d(\mathrm{UN}, \mathrm{AN}) = {}^{1}/_{2}.$




For small $m$, using ILPs, we verified that each compass matrix is almost as far away as possible from the others. Further, we believe that ID and UN are the two most distant frequency matrices, i.e., they form the diameter of our space. While showing this formally seems to be challenging, for each $m \in \{3, \ldots, 7\}$, using an ILP, we have verified that, indeed, $\mathrm{ID}_m$ and $\mathrm{UN}_m$ are the two most distant matrices under the positionwise distance.

### 4.2 Paths between Election Matrices

Next, we consider convex combinations of frequency matrices. Given two such matrices, $X$ and $Y$, and $\alpha \in [0,1]$, one might expect that matrix $Z = \alpha X + (1-\alpha)Y$ would lie at distance $(1-\alpha)\mathrm{POS}(X,Y)$ from $X$ and at distance $\alpha\mathrm{POS}(X,Y)$ from $Y$, so that we would have:

$$\mathrm{POS}(X,Y) = \mathrm{POS}(X,Z) + \mathrm{POS}(Z,Y).$$

However, without further assumptions this is not necessarily the case. Indeed, if we take $X = \mathrm{ID}_m$ and $Y = \mathrm{rID}_m$, then $0.5X + 0.5Y = \mathrm{AN}_m$ and $\mathrm{POS}(X,Y) = 0$, but $\mathrm{POS}(X, 0.5X + 0.5Y) = \mathrm{POS}(\mathrm{ID}, \mathrm{AN}) > 0$. Yet, if we arrange the two matrices $X$ and $Y$ so that their positionwise distance is achieved by the identity permutation of their column vectors, then their convex combination lies exactly between them.

**Proposition 4** (★). *Let $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots y_m)$ be two $m \times m$ frequency matrices such that $\mathrm{POS}(X,Y) = \sum_{i=1}^{m} \mathrm{EMD}(x_i, y_i)$. Then, for each $\alpha \in [0,1]$ it holds that $\mathrm{POS}(X,Y) = \mathrm{POS}(X, \alpha X + (1-\alpha)Y) + \mathrm{POS}(\alpha X + (1-\alpha)Y, Y)$.*

Using Proposition 4, for each two compass matrices, we can generate a sequence of matrices that form a path between them. For example, matrix $0.5\mathrm{ID} + 0.5\mathrm{UN}$ is exactly at the same distance from ID and from UN. In Figure 2a we show a map of elections that (in addition to the dataset of Szufa et al. [21]) contains our four compass matrices and for each two of them, i.e., for each two $X, Y \in \{\mathrm{ID}, \mathrm{UN}, \mathrm{AN}, \mathrm{ST}\}$, a set of $\lceil 50 \cdot d(X,Y) \rceil$ matrices obtained as their convex combinations with values of $\alpha$ uniformly distributed in $[0,1]$. Note that by the proof of Proposition 3, it holds that the positionwise distance between any two of our four matrices is achieved for the identity permuation, as required by Proposition 4.

## 5 Applying the Compass

In this section, we apply our compass to gain a better understanding of the map of elections created by Szufa et al. [21] and to place some real-life elections on the map. We also determine where Mallows and urn elections land.

## 5.1 A Map of Statistical Cultures with a Compass

In Figure 2a, we show a map of the 800 elections provided by Szufa et al. [21] in their 10x100 dataset, together with the compass. As expected, the uniformity matrix is close to the impartial culture elections, but still at some distance from them. Similarly, the identity matrix is very close to the Mallows elections with close-to-zero values of $\phi$. Indeed, such elections consist of nearly identical votes.

The red path, linking AN and ST, roughly partitions the elections into those closer to UN and those closer to ID. The latter group consists mostly of Mallows and urn elections (with low $\phi$ or high $\alpha$, respectively), but single-crossing and some single-peaked elections also make an appearance.

Analyzing the distances of elections to AN and ST, it is striking that 1D Interval elections lie closer to AN, while other hypercube elections lie closer to ST, even though, formally, they are similar.[2] Moreover, it is intriguing that single-peaked elections generated according to the Walsh model [22] are closer to ST, whereas those from the Conitzer model [5] (which are very similar to the 1D Interval ones) are closer to AN. To understand this phenomenon, let us look at single-peakedness and the models of Conitzer and Walsh more closely.

**Definition 1.** *Consider a set $C = \{c_1, \ldots, c_m\}$ of candidates and a linear order $c_1 \lhd c_2 \lhd \cdots \lhd c_m$, referred to as the* societal axis*. A vote $v$ is single-peaked with respect to $\lhd$ if for each $t \in [m]$ it holds that the $t$ top-ranked candidates in $v$ form an interval in $\lhd$. An election $E = (C, V)$ is single-peaked if there is a societal axis $\lhd$ such that each vote in $V$ is single-peaked with respect to $\lhd$.*

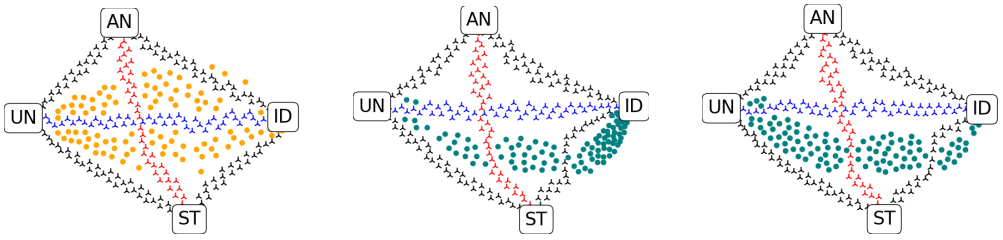
Intuitively, the societal axis orders the candidates with respect to some common, one-dimensional issue, such as, e.g., their position on the political left-to-right spectrum. In both the Conitzer and the Walsh model, we start by choosing the axis uniformly at random. Then, in the Conitzer model, we generate each vote as follows: We choose the top-ranked candidate uniformly at random and we keep on extending the vote with candidates to the left and to the right of the already-selected ones, deciding which one to pick with a coin toss, until the vote is complete. Thus, by choosing close-to-extreme candidates from different sides of the axis as top-ranked, we generate close-to-opposite preference orders with fairly high probability. As a consequence, the Conitzer model generates elections that have common features with the antagonism ones. Under the Walsh model, we choose each single-peaked preference order uniformly at random. There are few such preference orders with extreme candidates (with respect to the axis) ranked highly, but many with the center candidates on top and the extreme candidates ranked around the bottom. This leads to stratification.

## 5.2 Urn and Mallows Elections

Our next goal is to place "paths" of urn and Mallows elections on the map. In both cases it requires some care. Recall that the urn model has parameter $\alpha$, which takes values between 0 and $\infty$. To generate an urn election, we choose $\alpha$ according to the Gamma distribution with shape parameter $k = 0.8$ and scale parameter $\theta = 1$ (this ensures that about half of the urn elections are closer to UN than to ID, and the opposite for the other half; see Figure 3a).

Regarding the Mallows model, we have a parameter $\phi$ that takes values between 0 and 1, where 0 leads to generating ID elections and 1 leads to generating IC ones. It is thus intuitive to choose $\phi$ uniformly at random from the $[0, 1]$ interval. Yet, as seen in Figure 3b, doing so places elections quite unevenly on the map. Similarly, for different numbers of candidates

[2] Elections in a $t$-dimensional hypercube model are generated by drawing, for each candidate and each voter, a point from $[0, 1]^t$ uniformly at random. A voter then ranks the candidates with respect to the increasing distance of their points from his or her. For $t \in \{1, 2, 3\}$, $t$-dimensional hypercube elections are called 1D Interval, 2D Square, and 3D Cube, respectively. The others are called $t$D H-Cube.




(a) Urn elections (orange); $\alpha$ follows the Gamma distribution.

(b) Mallows elections (teal); $\phi$ follows the uniform distribution.

(c) Mallows elections (teal); rel-$\phi$ follows the uniform distribution.

Figure 3: Maps showing the four compass matrices, their connecting paths, and, respectively, urn elections and Mallows elections (for two distributions of their parameter). These visualizations are for 20 candidates and 100 voters.

the same value of $\phi$ leads to choosing elections at different distances from ID, which thereby also end up in (quite) different places on the map (see the top-left part of Figure 4). This imbalance, in essence, follows from the fact that making, say, five swaps among 10 candidates has a stronger effect than making five swaps among 20 candidates.

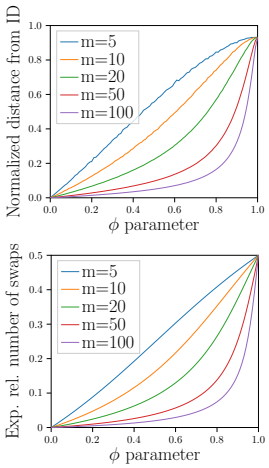
**Normalizing Mallows.** Let us consider a setting with $m$ candidates. For a $\phi \in [0, 1]$, let expswaps$(m, \phi)$ be the expected swap distance between an $m$-candidate vote generated using the Mallows model with parameter $\phi$ and the center vote. We define the relative expected number of swaps as:

$$\text{relswaps}(m, \phi) = \frac{\text{expswaps}(m,\phi)}{m(m-1)/2}$$

(see the bottom-left part of Figure 4 for plots of this value, depending on $\phi$ and $m$). In our approach, we choose a value rel-$\phi \in [0, 1]$ as a parameter, find a $\phi$ such that relswaps$(m, \phi)$ = rel-$\phi$, and draw an election from the Mallows model using this $\phi$ (see Appendix C.1 for details). Working on rel-$\phi$ instead of $\phi$ not only allows for an intuitive and natural interpretation of the parameter as the relative expected number of swaps in each vote (or the normalized distance from ID), but also for obtaining comparable elections for different numbers of candidates. In Figures 3b and 3c, we visualize Mallows elections generated with $\phi \in [0, 1]$ and rel-$\phi \in [0, 0.5]$ chosen uniformly at random, respectively (we use rel-$\phi \leq 0.5$ because for larger values one obtains analogous elections, but reversed; e.g., both rel-$\phi = 0$ and rel-$\phi = 1$ lead to identity elections). The latter figure shows a far more balanced distribution of points.

As computing the $\phi$ values based on rel-$\phi$ and the number of candidates requires some effort, we provide sample mappings in the table on the right side of Figure 4.

**Importance of the New Normalization.** The new parameterization of Mallows seems to be important. The Mallows model is often used in experiments and—in light of our findings—using a fixed $\phi$ for different numbers of candidates or drawing $\phi$ from a distribution independent of the number of candidates, may be questionable. Yet, this is not uncommon, as witnessed, e.g., in the works of Bachrach et al. [1], Betzler et al. [3], Garg et al. [10], Goldsmith et al. [11], Skowron et al. [20], and in a number of other papers. We mention these works as examples only; their authors designed their experiments as best practice suggested at the time and we do not challenge their high-level conclusions. Our point is that given the current evidence, they might prefer to design their experiments a bit differently.




| rel-$\phi\backslash m$ | 5 | 10 | 20 | 50 | 100 |
|---|---|---|---|---|---|
| 0 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 0.05 | 0.118 | 0.209 | 0.345 | 0.567 | 0.724 |
| 0.1 | 0.224 | 0.360 | 0.527 | 0.734 | 0.846 |
| 0.15 | 0.321 | 0.477 | 0.641 | 0.815 | 0.898 |
| 0.2 | 0.414 | 0.572 | 0.722 | 0.864 | 0.927 |
| 0.25 | 0.504 | 0.653 | 0.784 | 0.899 | 0.946 |
| 0.3 | 0.594 | 0.727 | 0.835 | 0.925 | 0.960 |
| 0.35 | 0.687 | 0.796 | 0.880 | 0.946 | 0.972 |
| 0.4 | 0.783 | 0.863 | 0.921 | 0.965 | 0.982 |
| 0.45 | 0.886 | 0.930 | 0.961 | 0.983 | 0.991 |
| 0.5 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |

Figure 4: Average normalized positionwise distances of Mallows elections from ID (plot on the top-left), relative expected number of swaps in votes drawn from the Mallows model (plot on the bottom-left), both depending on $\phi$ and for different numbers $m$ of candidates, and—in the table—the values of $\phi$ such that relswaps$(m, \phi)$ = rel-$\phi$ for $m \in \{5, 10, 20, 50, 100\}$ and rel-$\phi \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$.

## 5.3 Real-Life Elections on the Map

Let us now consider where real-life elections appear on the map. We start by describing the datasets that we use (mostly from PrefLib [15]). Whenever we speak of real-life elections in this section, we mean elections from our datasets.

We chose eleven real-life datasets, where each belongs to one of three groups. The first group contains *political* elections: city council elections from Glasgow and Aspen [18], elections from North Dublin, Meath (Irish), and elections held by non-profit organizations, trade unions, and professional organizations (ERS). The second group consists of *sport* elections: Tour de France (TDF), Giro d'Italia (GDI), speed skating, and figure skating (the former three dataset are due to us). The last group consists of *surveys*: preferences over Sushi, T-Shirt designs, and costs of living and population in different cities [4]. For TDF and GDI, each race is a vote, and each season is an election. For speed skating, each lap is a vote, and each competition is an election. For figure skating, each judge's opinion is a vote, and each competition is an election.

**Preprocessing the Data.** Each of our datasets consists of a number of preference profiles, where each profile consists of (possibly) partial preference orders. Since for our map we need elections with complete preference orders, we preprocess the data as follows.[3] First, for our new sport-related datasets (i.e., for TDG, GDI, and speed skating) we delete candidates and voters until each remaining candidate is ranked by at least 70% of the voters, and each voter ranks at least 70% of the candidates. Second, for all the datasets we disregard those profiles that contain fewer than ten candidates. Third, we extend each partial preference order in each remaining profile as follows (our approach is a much-simplified variant of the technique introduced by Doucette [6]):

1. If some $t$ top candidates are ranked and the remaining ones are not (except that they are reported to be below the top ones), then we fill $v$ iteratively: (1) We draw

[3]We speak of *elections* when we mean the collections of preference orders that we used in the map. We speak of *preference profiles* when we mean collections of preference orders at various stages of preprocessing.

uniformly at random one of the original votes from the same profile that ranks at least the top $t+1$ candidates and that agrees with $v$ on the top $t$ positions, and (2) we extend $v$ with whoever is ranked on the $(t+1)$-st position in this drawn vote (if there are no votes to sample from, then we extend $v$ with a candidate chosen uniformly at random). We repeat this process until $v$ is complete.

2. If a vote contains a tie of a different type than described above, then we break it uniformly at random.

After applying these preprocessing steps, each dataset contains profiles with ten or more candidates, and with complete votes. For each such profile, we select the ten candidates with the highest Borda score and remove the other ones. Finally, we delete some of the profiles based on the number of voters they contain (as compared to the other profiles in a given dataset; see Appendix C.2 for details). We refer to the resulting datasets as *intermediate*.

We treat each of the intermediate real-life datasets as a separate election model, from which we sample fifteen elections. We sample each election as follows. First, we randomly select one of the profiles. Second, we sample 100 votes from the profile uniformly at random (this implies that for profiles with less than 100 votes, we select some of the votes multiple times, and for profiles with more than 100 votes, we do not select some votes at all). After executing this procedure, we arrive at 11 datasets, each containing 15 elections consisting of 100 complete and strict votes over 10 candidates, which we use for our experiments. For a more detailed description of our data and its processing, see Appendix C.2.

**The Map.** In Figure 2b, we show a map of our real-life elections along with the compass, Mallows, and Urn elections. For readability we present Mallows and Urn elections as large, pale-colored areas. Not all real-life elections form clear clusters, hence the labels refer to the largest compact groupings.

While the map is not a perfect representation of distances among elections, analyzing it nevertheless leads to many conclusions. Most strikingly, real-life elections occupy a very limited area of the map; this is especially true for political elections and surveys. Except for several sport elections, all elections are closer to UN than to ID, and none of the real-life elections falls in the top-right part of the map. Another observation is that Mallows elections go right through the real-life elections, while Urn elections are on average far away. This means that for most real-life elections there exists a parameter $\phi$ such that elections generated according to the Mallows model with that parameter are relatively close (see the next section for specific recommendations).

Most of the political elections lie close to one another and are located next to Mallows elections and high-dimensional hypercube ones. At the same time, sport elections are spread over a larger part of the map and, with the exception of GDI, are shifted toward ID. As to the surveys, the City survey is basically equivalent to a sample from IC. The Sushi survey is surprisingly similar to political elections. The T-Shirt survey is shifted toward stratification (apparently, people often agree which designs are better and which are worse).

## 5.4 Capturing Real-Life Elections

In this section we analyze how to choose the rel-$\phi$ parameter so that elections generated using the Mallows model with our normalization resemble our real-life ones. We consider four different datasets each consisting of elections with 10 candidates and 100 voters (created as descibed in Section 5.3): the set of all political elections, the set of all sport elections, the set of all survey elections, and the combined set of all our real-life elections. For each of these four datasets, to find the value of rel-$\phi$ that produces elections that are as similar as possible to the respective real-life elections, we conduct the following experiment. For each

| Type of elections | Value of rel-$\phi$ | Average Normalized Distance | Norm. Std. Dev. | Num. of elections |
|---|---|---|---|---|
| Political elections | 0.375 | 0.15 | 0.036 | 60 |
| Sport elections | 0.267 | 0.27 | 0.080 | 60 |
| Survey elections | 0.365 | 0.20 | 0.034 | 45 |
| All real-life elections | 0.350 | 0.22 | 0.106 | 165 |

Table 1: Values of rel-$\phi$ such that elections generated with Mallows model for $m = 10$ are, on average, as close as possible to elections from the respective dataset. We include the average distance of elections generated with Mallows model for this parameter rel-$\phi$ from the elections from the dataset as well as the standard deviation, both normalized by $D(10) = 33$. The last column gives the number of elections in the respective real-life dataset.

rel-$\phi \in \{0, 0.001, 0.002, ..., 0.499, 0.5\}$, we generate 100 elections with 10 candidates and 100 voters from the Mallows model with the given rel-$\phi$ parameter. Subsequently, we compute the average distance between these elections and the elections from the respective dataset. Finally, we select the value of rel-$\phi$ that minimizes this distance. We present the results of this experiment in Table 1.

Recall that in the previous section we have observed that a majority of real-life elections are close to some elections generated from the Mallows model with a certain dispersion parameter. However, we have also seen that the real-life datasets consist of elections that differ to a certain extent from one another (in particular, this is very visible for the sports elections). Thus, it is to be expected that elections drawn from the Mallows model for a fixed dispersion parameter are at some non-zero (average) distance from the real-life ones. Indeed, this is the case here. Nevertheless, the more homogeneous political elections and survey elections can be quite well captured using the Mallows model with parameter rel-$\phi = 0.375$ and rel-$\phi = 0.365$, respectively. Generally speaking, if one wants to generate elections that should be particularly close to elections from the real world, then choosing a rel-$\phi$ value between 0.35 and 0.39 is a good strategy. If, however, one wants to capture the full spectrum of real-life elections, then we recommend using the Mallows model with different values of rel-$\phi$ from the interval $[0.25, 0.4]$. In Appendix C.1.2, we provide some evidence that these recommendations are also applicable for different numbers of candidates.

# 6 Conclusions and Future Work

Perhaps the most important conclusion from our work is that the Mallows model is very good at generating elections similar, under positionwise distance, to those that appear in the real world, but to achieve this effect one needs to choose its dispersion parameter carefully. Specifically, the parameter should depend both on the type of elections that we are interested in (such as, e.g., political ones or sports ones) and on the number of candidates in the election. We have provided some recommendations for its choice.

Our work leads to a number of open problems. On the practical side, all our experiments regarded elections with exactly 10 candidates. It is important to extend them to different numbers of candidates and validate that our conclusions still hold (see Appendix C.1.2 for an initial discussion regarding this issue). On the mathematical side, one of the most intriguing question is whether, indeed, the identity and uniformity election are the two farthest ones in the positionwise metric.

**Acknowledgments.** NB was supported by the DFG project MaMu (NI 369/19). PF conducted this research based on his ERC project PRAGMA.

# References


[1] Y. Bachrach, O. Lev, Y. Lewenberg, and Y. Zick. Misrepresentation in district voting. In *Proceedings of IJCAI-2016*, pages 81–87, 2016.

[2] S. Berg. Paradox of voting under an urn model: The effect of homogeneity. *Public Choice*, 47(2):377–387, 1985.

[3] N. Betzler, R. Bredereck, and R. Niedermeier. Theoretical and empirical evaluation of data for exact Kemeny rank aggregation. *Autonomous Agents and Multiagent Systems*, 28(5):721–748, 2014.

[4] I. Caragiannis, X. Chatzigeorgiou, G. Krimpas, and A. Voudouris. Optimizing positional scoring rules for rank aggregation. *Artificial Intelligence*, 267:58–77, 2019.

[5] V. Conitzer. Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research*, 35:161–191, 2009.

[6] J. Doucette. *Social Choice for Partial Preferences Using Imputation.* PhD thesis, University of Waterloo, 2016.

[7] P. Faliszewski, P. Skowron, A. Slinko, S. Szufa, and N. Talmon. How similar are two elections? In *Proceedings of AAAI-2019*, pages 1909–1916, 2019.

[8] T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[9] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3):324–360, 2006.

[10] N. Garg, L. Gelauff, S. Sakshuwong, and A. Goel. Who is in your top three? Optimizing learning in elections with many candidates. In *Proceedings of HCOMP-2019*, pages 22–31, 2019.

[11] J. Goldsmith, J. Lang, N. Mattei, and P. Perny. Voting with rank dependent scoring rules. In *Proceedings of AAAI-14*, pages 698–704, 2014.

[12] R. Horn and C. Johnson. *Matrix Analysis, 2nd Ed.* Cambridge University Press, 2012.

[13] D. Leep and G. Myerson. Marriage, magic, and solitaire. *The American Mathematical Monthly*, 106(5):419–429, 1999.

[14] C. Mallows. Non-null ranking models. *Biometrica*, 44:114–130, 1957.

[15] N. Mattei and T. Walsh. Preflib: A library for preferences. In *Proceedings of ADT-2013*, pages 259–270, 2013.

[16] J. McCabe-Dansted and A. Slinko. Exploratory analysis of similarities between social choice rules. *Group Decision and Negotiation*, 15:77–107, 2006.

[17] OEIS Foundation Inc. The on-line encyclopedia of integer sequences, 2020. URL http://oeis.org/A008302.

[18] J. O'Neill. Open STV, www.openstv.org. 2013.

[19] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical recipes: The art of scientific computing, 3rd Edition.* Cambridge University Press, 2007.

[20] P. Skowron, P. Faliszewski, and A. Slinko. Achieving fully proportional representation: Approximability result. *Artificial Intelligence*, 222:67–103, 2015.

[21] S. Szufa, P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. Drawing a map of elections in the space of statistical cultures. In *Proceedings of AAMAS-20*, pages 1341–1349, 2020.

[22] T. Walsh. Generating single peaked votes. Technical Report arXiv:1503.02766 [cs.GT], arXiv.org, March 2015.

Niclas Boehmer
Technische Universität Berlin
Berlin, Germany
Email: niclas.boehmer@tu-berlin.de

Robert Bredereck
Humboldt-Universität zu Berlin
Berlin, Germany
Email: robert.bredereck@hu-berlin.de

Piotr Faliszewski
AGH University
Krakow, Poland
Email: faliszew@agh.edu.pl

Rolf Niedermeier
Technische Universität Berlin
Berlin, Germany
Email: rolf.niedermeier@tu-berlin.de

Stanisław Szufa
Jagiellonian University
Krakow, Poland
Email: stanislaw.szufa@uj.edu.pl

# Appendix

## A Missing Material from Section 3

### A.1 Missing Proofs and Discussion

**Proposition 1 (★).** *Given a position matrix $X \in \mathcal{P}(m)$, one can compute in $O(m^{4.5})$ time an election $E$ that contains at most $m^2 - 2m + 2$ different votes such that $\#\mathrm{pos}(E) = X$.*

*Proof.* Let $X$ be our input $m \times m$ matrix and let $C = \{c_1, \ldots, c_m\}$ be a set of candidates. Our algorithm creates an election $E = (C, V)$ iteratively, as follows. In each iteration we first create a bipartite graph $G$ with vertex sets $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_m\}$. For each $i, j \in [m]$, if $x_{i,j}$ is nonzero, then we put an edge between $a_i$ and $b_j$ (vertices in $A$ correspond to rows of $X$ and vertices in $B$ correspond to the columns). Next, we compute a perfect matching $M$ in $G$ (we will see later that it is guaranteed to exist). Let $v$ be the vote that ranks $c_j$ on position $i$ exactly if $M(a_i) = b_j$ ($v$ is well-defined because $M$ is a perfect matching). Let $P$ be the position matrix corresponding to vote $v$, i.e., to election $(C, (v))$, and let $z$ be the largest integer such that $X - zP$ contains only non-negative entries. Then, we add $z$ copies of $v$ to $V$ and set $X := X - zP$. We proceed to the next iteration until $X$ becomes the zero matrix.

To prove the correctness of the algorithm, we show that at each iteration the constructed graph $G$ has a perfect matching. Let us assume that this is not the case. Note that each row and each column in the current $X$ sums up to the same integer, say $n'$. Since there is no perfect matching, by Hall's theorem, there is a subset of vertices $A' \subseteq A$ such that the neighborhood $B' \subseteq B$ of $A'$ in $G$ contains fewer than $|A'|$ vertices. Yet, we have that $\sum_{a_i \in A', b_j \in B'} x_{i,j} = n'|A'|$, as we sum up all the nonzero entries of each row corresponding to a vertex from $A'$. However, this implies that $|B'| \geq |A'|$ because each column corresponding to a vertex from $B'$ sums up to $n'$, but we do not necessarily include all its nonzero entries. This is a contradiction.

The algorithm terminates after at most $m^2 - m + 1$ steps (in each step at least one more entry of $X$ becomes zero, and in the last step, $m$ entries become zero). Each step requires $\mathcal{O}(m^{2.5})$ time to compute the matching, so the overall running time is $\mathcal{O}(m^{4.5})$. This implies that $V$ contains at most $m^2 - m + 1$ different votes; indeed, using a similar argument as in Leep and Myerson [13] it can be shown that the algorithm always terminates after at most $m^2 - 2m + 2$ steps. □

The above proof shows that if all the entries of a given position matrix are at least $t > 0$, then we can create an election that induces this matrix by first choosing $t$ votes completely arbitrarily, and only then resorting to matching in a bipartite graph. In consequence, position matrices where all entries are large correspond to large and varied sets of elections. While in some situations this is unavoidable (e.g., when one considers impartial culture elections with many more voters than candidates), it may mean that the features observed for one election corresponding to a given matrix are not shared by many of the others.

Fortunately, the matrices in the datasets of Szufa et al. [21] either contain zero entries or have a very small smallest entry (in the $10 \times 100$ dataset, 69% of the matrices have zero entries, and among the remaining ones, the average value of the smallest entry is 2.61 with standard deviation 1.15).

Naturally, if a matrix does have some zero entries it still may correspond to a large and varied set of elections; we simply claim that if all the entries are large then this is, in essence, unavoidable.

**Proposition 2.** *Given an $m \times m$ bistochastic matrix $X$ and an integer $n$, one can compute in polynomial time an election $E$ with $n$ voters whose position matrix $P$ satisfies $|nx_{i,j} - p_{i,j}| \leq 1$ for each $i, j \in [m]$ and, under this condition, minimizes the value $\sum_{1 \leq i,j \leq m} |nx_{i,j} - p_{i,j}|$.*

*Proof (continued from Section 3).* Here we provide a fully deterministic algorithm for computing the matrix $D$ from the proof presented in the main body, which does not invoke dependent rounding, and which minimizes the value $\sum_{1 \leq i,j \leq m} |nx_{i,j} - p_{i,j}|$.

Consider matrix $Y$ from the first part of the proof and let $B = \sum_{1 \leq i,j \leq m} y_{i,j}$.

We form a flow network with source $s$, nodes $v_{i,j}$ for each $i, j \in [m]$, "pre-sink" nodes $t_1, \ldots, t_m$, and sink node $t$. For each $i \in [m]$, we have a directed path which starts at the source node $s$, then goes to $v_{i,1}$, next to $v_{i,2}$, and so on, until $v_{i,m}$. Each edge on this path has capacity equal to $\sum_{j=1}^{m} y_{i,j}$ (recall that this value is an integer). For each $i, j \in [m]$, we have an edge from $v_{i,j}$ to $t_j$, with capacity one and with cost $1 - 2y_{i,j}$ (all the other types of edges have cost 0). Finally, for each $j \in [m]$, we have an edge from $t_j$ to $t$ with capacity $\sum_{i=1}^{m} y_{i,j}$ (this value is an integer). Next, we compute in polynomial time (using some classic algorithm) an integral flow that moves $\sum_{i,j \in [m]} y_{i,j}$ units of flow from $s$ to $t$ at the lowest possible cost (which we denote $B_f$). If this flow existed, then we could compute matrix $D$ by setting, for each $i, j \in [m]$, $d_{i,j}$ to be the amount of flow (0 or 1) going from $v_{i,j}$ to $t_j$. Indeed, by definition of our flow network, for each $i$ we would have that $\sum_{j=1}^{m} d_{i,j} = \sum_{j=1}^{m} y_{i,j}$ (because when a flow enters some node $v_{i,j}$, then it can either go to $t_j$ or to $v_{i,j+1}$). Due to the capacities on the edges from the pre-sink nodes to the sink, for each $j \in [m]$ we would also have that $\sum_{i=1}^{m} d_{i,j} = \sum_{i=1}^{m} y_{i,j}$. These two properties are equivalent to ensuring that the degrees of the vertices in $G'$ are equal to the fractional degrees in $G$, as is done by dependent rounding. Further, we have that:

$$\sum_{1 \leq i,j \leq m} |nx_{i,j} - (\lfloor nx_{i,j} \rfloor + d_{i,j})| = B + B_f.$$

To see why this is the case, note that if all the values $d_{i,j}$ where 0, then the left-hand sum would be $B$, and on the right-hand size $B_f$ would be 0. Each $d_{i,j} = 1$ increases both sides of the sum by the same amount.

It remains to see that the desired flow indeed always exists. However, this follows from the previous algorithm: Since dependent rounding always produces a desired matching, which induces matrix $D$, $D$ induces a flow of our desired value. □

# B Missing Material from Section 4

## B.1 The Four Matrices

We start this subsection by proving Proposition 3. Afterwards, we provide some evidence that our four compass elections are, indeed, almost as far away from each other as possible.

**Proposition 3 (★).** *If $m$ is divisible by 4, then it holds that:*

1. $\mathrm{POS}(\mathrm{ID}_m, \mathrm{UN}_m) = \frac{1}{3}(m^2 - 1)$,
2. $\mathrm{POS}(\mathrm{ID}_m, \mathrm{AN}_m) = \mathrm{POS}(\mathrm{UN}_m, \mathrm{ST}_m) = \frac{m^2}{4}$,
3. $\mathrm{POS}(\mathrm{ID}_m, \mathrm{ST}_m) = \mathrm{POS}(\mathrm{UN}_m, \mathrm{AN}_m) = \frac{2}{3}(\frac{m^2}{4} - 1)$,
4. $\mathrm{POS}(\mathrm{AN}_m, \mathrm{ST}_m) = \frac{13}{48}m^2 - \frac{1}{3}$.

*Proof.* **$\mathrm{ID_m}$ and $\mathrm{UN_m}$:** We start by computing the distance between $\mathrm{ID}_m$ and $\mathrm{UN}_m$. Note that $\mathrm{UN}_m$ always remains the same matrix independent of how its columns are ordered. Thus, we can compute the distance between these two matrices using the identity permutation between the columns of the two matrices: $\mathrm{POS}(\mathrm{ID}_m, \mathrm{UN}_m) = \sum_{i=1}^{m} \mathrm{EMD}((\mathrm{ID}_m)_i, (\mathrm{UN}_m)_i) = \sum_{i=1}^{m}(\sum_{j=1}^{i-1} \frac{j}{m} + \sum_{j=1}^{m-i} \frac{j}{m})$
$= \frac{1}{m}\sum_{i=1}^{m}(\frac{1+(i-1)}{2}(i-1) + \frac{1+(m-i)}{2}(m-i))$
$= \frac{1}{2m}\sum_{i=1}^{m}(2i^2 - 2i - 2mi + m^2 + m)$
$= \frac{1}{2m}(2\frac{m(m+1)(2m+1)}{6} - m(m+1) - m^2(m+1) + m(m^2+m))$
$= \frac{1}{2m}(\frac{(m^2+m)(2m+1)}{3} - (m+1)(m+m^2) + m(m^2+m))$
$= \frac{m+1}{2}(\frac{(2m+1)}{3} - (m+1) + m) = \frac{(m+1)(m-1)}{3} = \frac{1}{3}(m^2-1).$

In the following, we use $(*)$ when we omit some calculations analogous to the calculations for $\mathrm{POS}(\mathrm{ID}_m, \mathrm{UN}_m)$.

**$\mathrm{UN_m}$ and $\mathrm{ST_m}$:** Similarly, we can also directly compute the distance between $\mathrm{UN}_m$ and $\mathrm{ST}_m$ using the identity permutation between the columns of the two matrices. In this case, all column vectors of the two matrices have indeed the same EMD distance to each other: $\mathrm{POS}(\mathrm{UN}_m, \mathrm{ST}_m) = m \cdot (\frac{1}{2} + 2 \cdot \sum_{i=1}^{\frac{m}{2}-1} \frac{i}{m}) = \frac{m}{2} + \frac{m}{2}(\frac{m}{2} - 1) = \frac{m^2}{4}$.

**$\mathrm{UN_m}$ and $\mathrm{AN_m}$:** Next, we compute the distance between $\mathrm{UN}_m$ and $\mathrm{AN}_m$ using the identity permutation between the columns of the two matrices. Recall that $\mathrm{AN}_m$ can be written as:

$$\mathrm{AN}_m = 0.5 \begin{bmatrix} \mathrm{ID}_{m/2} & \mathrm{rID}_{m/2} \\ \mathrm{rID}_{m/2} & \mathrm{ID}_{m/2} \end{bmatrix}.$$

Thus, it is possible to reuse our ideas from computing the distance between identity and uniformity:
$\mathrm{POS}(\mathrm{UN}_m, \mathrm{AN}_m) = 4\sum_{i=1}^{\frac{m}{2}}(\sum_{j=1}^{i-1} \frac{j}{m} + \sum_{j=1}^{\frac{m}{2}-i} \frac{j}{m}) = (*) = \frac{2}{3}(\frac{m^2}{4} - 1)$.

**$\mathrm{ID_m}$ and $\mathrm{ST_m}$:** There exist only two different types of column vectors in $\mathrm{ST}_m$, i.e., $\frac{m}{2}$ columns starting with $\frac{m}{2}$ entries of value $\frac{2}{m}$ followed by $\frac{m}{2}$ zero-entries and $\frac{m}{2}$ columns starting with $\frac{m}{2}$ zero entries followed by $\frac{m}{2}$ entries of value $\frac{2}{m}$. In $\mathrm{ID}_m$, $\frac{m}{2}$ columns have a one entry in the first $\frac{m}{2}$ rows and $\frac{m}{2}$ columns have a one entry in the last $\frac{m}{2}$ rows. Thus, again the identity permutation between the columns of the two matrices minimizes the EMD distance:
$\mathrm{POS}(\mathrm{ID}_m, \mathrm{ST}_m) = 2 \cdot \mathrm{POS}(\mathrm{ID}_{\frac{m}{2}}, \mathrm{UN}_{\frac{m}{2}}) = \frac{2}{3}(\frac{m^2}{4} - 1)$

**$\mathrm{AN_m}$ and $\mathrm{ST_m}$:** We now turn to computing the distance between $\mathrm{AN}_m = (\mathrm{an}_1, \ldots, \mathrm{an}_m)$ and $\mathrm{ST}_m = (\mathrm{st}_1, \ldots, \mathrm{st}_m)$. As all column vectors of $\mathrm{AN}_m$ are palindromes, each column vector of $\mathrm{AN}_m$ has the same EMD distance to all column vectors of $\mathrm{ST}_m$, i.e., for $i \in [m]$ it holds that $\mathrm{EMD}(\mathrm{an}_i, \mathrm{st}_j) = \mathrm{EMD}(\mathrm{an}_i, \mathrm{st}_{j'})$ for all $j, j' \in [m]$. Thus, the distance between $\mathrm{AN}_m$ and $\mathrm{ST}_m$ is the same for all permutation between the columns of the two matrices. Thus, we again use the identity permutation. We start by computing $\mathrm{EMD}(\mathrm{an}_i, \mathrm{st}_i)$ for different $i \in [m]$ separately distinguishing two cases. Let $i \in [\frac{m}{4}]$. Recall that $\mathrm{an}_i$ has a 0.5 at position $i$ and position $m-i+1$ and that $\mathrm{st}_i$ has a $\frac{2}{m}$ at entries $j \in [\frac{m}{2}]$. We now analyze how to transform $\mathrm{an}_i$ to $\mathrm{st}_i$. For all $j \in [i-1]$, it is clear that it is optimal that the value $\frac{2}{m}$ moved to position $j$ comes from position $i$. The overall cost of this is $\sum_{j=1}^{i-1} \frac{2j}{m}$. Moreover, the remaining surplus value at position $i$ (that is, $\frac{1}{2} - \frac{2i}{m}$) needs to be moved toward the end. Thus, for $j \in [i+1, \frac{m}{4}]$, we move value $\frac{2}{m}$ from position $i$ to position $j$. The overall cost of this is $\sum_{j=1}^{\frac{m}{4}-i} \frac{2j}{m}$. Lastly, we need to move value $\frac{2}{m}$ to positions $j \in [\frac{m}{4}+1, \frac{m}{2}]$. This needs to come from position $m-i+1$. Thus, for each $j \in [\frac{m}{4}+1, \frac{m}{2}]$, we move value $\frac{2}{m}$ from position $m-i+1$ to position $j$. The overall cost of this is $\frac{1}{2} \cdot (\frac{m}{2} - i) + \sum_{j=1}^{\frac{m}{4}} \frac{2j}{m} = \frac{1}{2}(\frac{m}{2} - i) + \frac{m}{16} + \frac{1}{4}$

Now, let $i \in [\frac{m}{4}+1, \frac{m}{2}]$. For $j \in [\frac{m}{4}]$, we need to move value $\frac{2}{m}$ from position $i$ to position $j$. The overall cost of this is $\frac{1}{2} \cdot (i - \frac{m}{4} - 1) + \sum_{j=1}^{\frac{m}{4}} \frac{2j}{m} = \frac{1}{2} \cdot (i - \frac{m}{4} - 1) + \frac{m}{16} + \frac{1}{4}$. For $j \in [\frac{m}{4}+1, \frac{m}{2}]$, we need to move value $\frac{2}{m}$ from position $m - i + 1$ to position $j$. The overall cost of this is $\frac{1}{2} \cdot (\frac{m}{2} - i) + \sum_{j=1}^{\frac{m}{4}} \frac{2j}{m} = \frac{1}{2} \cdot (\frac{m}{2} - i) + \frac{m}{16} + \frac{1}{4}$.

Observing that the case $i \in [\frac{3m}{4}+1, m]$ is symmetric to $i \in [\frac{m}{4}]$ and the case $i \in [\frac{m}{2}+1, \frac{3m}{4}]$ is symmetric to $i \in [\frac{m}{4}+1, \frac{m}{2}]$ the EMD distance between $\mathrm{AN}_m$ and $\mathrm{ST}_m$ can be computed as follows:
$\mathrm{POS}(\mathrm{AN}_m, \mathrm{ST}_m) = 2 \cdot (A + \frac{1}{2} \cdot (\sum_{i=1}^{\frac{m}{4}} \frac{m}{2} - i) + \frac{m}{4} \cdot (\frac{m}{16} + \frac{1}{4}) + \frac{1}{2} \cdot (\sum_{i=\frac{m}{4}+1}^{\frac{m}{2}} \cdot (i - \frac{m}{4} - 1)) +$
$\frac{m}{4} \cdot (\frac{m}{16} + \frac{1}{4}) + \frac{1}{2} \cdot (\sum_{i=\frac{m}{4}+1}^{\frac{m}{2}} \frac{m}{2} - i) + \frac{m}{4} \cdot (\frac{m}{16} + \frac{1}{4}))$
$= \frac{m^2}{48} - \frac{1}{3} + \frac{3m^2-4m}{32} + \frac{m}{2} \cdot (\frac{m}{16} + \frac{1}{4}) + \frac{m^2-4m}{32} + \frac{m}{2} \cdot (\frac{m}{16} + \frac{1}{4}) + \frac{m^2-4m}{32} + \frac{m}{2} \cdot (\frac{m}{16} + \frac{1}{4}))$
$= \frac{m^2}{48} - \frac{1}{3} + \frac{3m^2-4m}{32} + \frac{3m}{2}(\frac{m}{16} + \frac{1}{4}) + \frac{m^2-4m}{16} = (\frac{1}{48} + \frac{3}{32} + \frac{3}{32} + \frac{1}{16})m^2 + (-\frac{4}{32} + \frac{3}{8} - \frac{4}{16})m\frac{1}{3} = \frac{13}{48}m^2 - \frac{1}{3}$
with
$A = \sum_{i=1}^{\frac{m}{4}} (\sum_{j=1}^{i-1} \frac{2j}{m} + \sum_{j=1}^{\frac{m}{4}-i} \frac{2j}{m}) = (*) = \frac{1}{6}(\frac{m^2}{16} - 1) = \frac{1}{2}(\frac{m^2}{48} - \frac{1}{3})$

**$\mathrm{ID}_\mathbf{m}$ and $\mathrm{AN}_\mathbf{m}$:** Lastly, we consider $\mathrm{ID}_m = (\mathrm{id}_1, \ldots, \mathrm{id}_m)$ and $\mathrm{AN}_m = (\mathrm{an}_1, \ldots, \mathrm{an}_m)$. Note that, for $i \in [m]$, $\mathrm{id}_i$ contains a 1 at position $i$ and $\mathrm{an}_i$ contains a 0.5 at position $i$ and position $m - i$. Note further that for $i \in [\frac{m}{2}]$ it holds that $\mathrm{an}_i = \mathrm{an}_{m-i+1}$. Fix some $i \in [\frac{m}{2}]$. For all $j \in [i, m - i + 1]$ it holds that $\mathrm{EMD}(\mathrm{an}_i, \mathrm{id}_j) = \frac{m-2i+1}{2}$ and for all $j \in [1, i-1] \cup [m - i + 2, m]$ it holds that $\mathrm{EMD}(\mathrm{an}_i, \mathrm{id}_j) > \frac{m-2i+1}{2}$. That is, for every $i \in [m]$, $\mathrm{an}_i$ has the same distance to all column vectors of $\mathrm{ID}_m$ where the one entry lies in between the two 0.5 entries of $\mathrm{an}_i$ but a larger distance to all column vectors of $\mathrm{ID}_m$ where the one entry is above the top 0.5 entry of $\mathrm{an}_i$ or below the bottom 0.5 entry of $\mathrm{an}_i$. Thus, it is optimal to choose a mapping of the column vectors such that for all $i \in [m]$ it holds that $\mathrm{an}_i$ is mapped to a vector $\mathrm{id}_j$ where the one entry of $\mathrm{id}_j$ lies between the two 0.5 in $\mathrm{an}_i$. This is, among others, achieved by the identity permutation, which we use to compute:
$\mathrm{POS}(\mathrm{ID}_m, \mathrm{AN}_m) = 2 \sum_{i=1}^{\frac{m}{2}} (\frac{1}{2}(m - 2i + 1))$
$= \frac{m}{2}m - \frac{m}{2}(\frac{m}{2} + 1) + \frac{m}{2} = \frac{m^2}{4}$ □

Focusing on $6 \times 6$ matrices, we now argue that our four compass matrices are quite far away from each other. We focus on $m = 6$, as this is the largest dimension for which the ILPs we will use in this section were able to compute optimal solutions within several hours. While it could be the case that for larger $m$ the results are significantly different, we do not expect that this is the case (see e.g. Section 4.1). Moreover, the case $m = 6$ is already interesting on its own, as elections with only six candidates also regularly appear in the real world.

We present two different justifications that our compass matrices cover very different areas on the map. First, we explain how we created the compass by adding the four matrices one after another, trying to maximize the distances between them. Second, we argue that each of the matrices is almost as far away from the other three as possible.

### B.1.1 Iteratively Building the Compass

We built the compass in an iterative fashion, i.e., we added the matrices one after each other. We first wanted to find the two matrices that are furthest away from each other. As discussed in Section 4.1, we believe that these two matrices are the identity and uniformity matrices. For $m = 6$, using an ILP, we were able to verify that, indeed, $\mathrm{ID}_6$ and $\mathrm{UN}_6$ have the highest distance among all pairs of $6 \times 6$ frequency matrices.

| | $\mathrm{POS}(\mathrm{ID}_6,\cdot)$ | $\mathrm{POS}(\mathrm{UN}_6,\cdot)$ | $\mathrm{POS}(\mathrm{UN}_6,\cdot)$ $+\mathrm{POS}(\mathrm{UN}_6,\cdot)$ |
|---|---|---|---|
| $M$ | 7.18 | 7.18 | 14.36 |
| $S$ | 10 | 4.66 | 14.66 |
| $\mathrm{AN}_6$ | 9 | 5.33 | 14.33 |
| $\mathrm{ST}_6$ | 9 | 5.33 | 14.33 |

Table 2: Distance of four $6 \times 6$ matrices to $\mathrm{ID}_6$ and $\mathrm{UN}_6$. $M$ is the matrix with maximum possible minimum distance and $S$ the matrix with maximum possible summed distance to $\mathrm{ID}_6$ and $\mathrm{UN}_6$.

| | $\mathrm{POS}(\mathrm{ID}_6,\cdot)$ | $\mathrm{POS}(\mathrm{UN}_6,\cdot)$ | $\mathrm{POS}(\mathrm{AN}_6,\cdot)$ | $\mathrm{POS}(\mathrm{ID}_6,\cdot)$ $+\mathrm{POS}(\mathrm{UN}_6,\cdot)$ $+\mathrm{POS}(\mathrm{AN}_6,\cdot)$ |
|---|---|---|---|---|
| $M'$ | 7.17 | 7.16 | 7.82 | 22.15 |
| $\mathrm{ST}_6$ | 9 | 5.33 | 9.66 | 23.99 |

Table 3: Distance of two $6 \times 6$ matrices to $\mathrm{ID}_6$, $\mathrm{UN}_6$, and $\mathrm{AN}_6$. $M'$ is the matrix with maximum possible minimum distance to the three matrices.

Next, we wanted to find a frequency matrix that is as far away from $\mathrm{ID}_6$ and $\mathrm{UN}_6$ as possible. However, in this context, it is not entirely clear what "as far as possible" means, as one might, e.g., maximize the sum or the minimum of the distances to $\mathrm{ID}_6$ and $\mathrm{UN}_6$. Using again an ILP, we found two (quite unstructured) matrices $M$ (with maximum minimum distance) and $S$ (with maximum summed distance):

$$M = \begin{bmatrix} 0.55 & 0.45 & 0 & 0 & 0 & 0 \\ 0.05 & 0.25 & 0.7 & 0 & 0 & 0 \\ 0 & 0.16 & 0 & 0.6 & 0.24 & 0 \\ 0 & 0.14 & 0.3 & 0 & 0.16 & 0.4 \\ 0 & 0 & 0 & 0 & 0.4 & 0.6 \\ 0.4 & 0 & 0 & 0.4 & 0.2 & 0 \end{bmatrix}$$

$$S = \begin{bmatrix} 0.33 & 0.33 & 0.33 & 0 & 0 & 0 \\ 0.08 & 0.08 & 0.08 & 0.17 & 0.17 & 0.42 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0.08 & 0.08 & 0.8 & 0.33 & 0.33 & 0.08 \\ 0.17 & 0.17 & 0.5 & 0 & 0 & 0.17 \\ 0.33 & 0.33 & 0 & 0 & 0 & 0.33 \end{bmatrix}.$$

The distances of $M$ and $S$ to $\mathrm{ID}_6$ and $\mathrm{UN}_6$ can be found in Table 2. As both $M$ and $S$ are quite unstructured, it is unclear how to generalize them to higher dimensions, and it is not intuitively clear what kind of elections they resemble, we started thinking about canonical matrices distant from identity and uniformity and came up with AN and ST. As displayed in Table 2, both $\mathrm{AN}_6$ and $\mathrm{ST}_6$ are quite far away from both $\mathrm{ID}_6$ and $\mathrm{UN}_6$. Overall, the summed distance of these two matrices to $\mathrm{ID}_6$ and $\mathrm{UN}_6$ is close to the maximum achievable distance. We decided to first add antagonism to the compass.

In the next step, we again computed two matrices $M'$ with maximum minimum distance from $\mathrm{ID}_6$, $\mathrm{UN}_6$, and $\mathrm{AN}_6$, and $S'$ with maximum summed distance from $\mathrm{ID}_6$, $\mathrm{UN}_6$, and

| | Minimum distance | Summed distance | Maxmimum minimum distance | Maxmimum summed distance |
|---|---|---|---|---|
| ID | 5.33 | 26 | 7.75 | 26 |
| UN | 5.33 | 26 | 7.63 | 26.44 |
| AN | 5.33 | 23.99 | 7.05 | 24.17 |
| ST | 5.33 | 23.99 | 7.16 | 23.99 |

Table 4: For each of the four compass matrices (with m=6), the first column shows its minimum distance to the other three matrices and the second column its summed distance to the other three matrices. The third columns shows the maximum minimum distance of any matrix to the other three matrices and the fourth column the maximum summed distance of any matrix to the other three matrices.

$\mathrm{AN}_6$. It turned out that $S' = \mathrm{ST}_6$ and

$$M' = \begin{bmatrix} 0.33 & 0.33 & 0.33 & 0 & 0 & 0 \\ 0.33 & 0.33 & 0.33 & 0 & 0 & 0 \\ 0 & 0 & 0.08 & 0.25 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0.25 & 0.08 & 0 & 0 \\ 0 & 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0 & 0 & 0 & 0.33 & 0.33 & 0.33 \end{bmatrix}.$$

As depicted in Table 3, $\mathrm{ST}_6$ is quite far away from each of the three already fixed matrices from our compass. As, at the same time, $\mathrm{ST}_6$ maximizes the summed distance among all matrices, we selected stratification as our fourth compass matrix.

#### B.1.2 Distance of One Compass Matrix from the Other Three

We now describe a second experiment to verify that our compass matrices are indeed nearly as far away from each other as possible. For each of the four matrices $X \in \{\mathrm{ID}_6, \mathrm{UN}_6, \mathrm{AN}_6, \mathrm{ST}_6\}$, using an ILP, we compute the best matrix to replace $X$ in order to maximize the diversity of the resulting set of four matrices. That is, we compute the $6\times6$ matrix with maximum summed distance to the matrices in $\{\mathrm{ID}_6, \mathrm{UN}_6, \mathrm{AN}_6, \mathrm{ST}_6\} \setminus \{X\}$ and the $6 \times 6$ matrix with maximum minimum distance to a matrix in $\{\mathrm{ID}_6, \mathrm{UN}_6, \mathrm{AN}_6, \mathrm{ST}_6\} \setminus \{X\}$. The results of this experiment along with the minimum/summed distance of $X$ to the matrices in $\{\mathrm{ID}_6, \mathrm{UN}_6, \mathrm{AN}_6, \mathrm{ST}_6\} \setminus \{X\}$ are shown in Table 4. The results show that each compass matrix has the highest possible or close to the highest possible summed distance to the other three matrices. This implies that none of our matrices can be replaced by another matrix such that the resulting set covers a significantly larger area of the map. Concerning the minimum distance of each compass matrix to the other three, the compass matrices are no longer very close to being optimal. Nevertheless, each pair of matrices is quite far away from each other. Moreover, maximizing the minimum and summed distance are (at least partly) conflicting optimization goals and, for our purpose of putting a compass on the map, the summed distance is of greater importance.

### B.2 Paths Between Election Matrices

**Proposition 4** (★). *Let $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots y_m)$ be two $m \times m$ frequency matrices such that $\mathrm{POS}(X, Y) = \sum_{i=1}^{m} \mathrm{EMD}(x_i, y_i)$. Then, for each $\alpha \in [0, 1]$ it holds that $\mathrm{POS}(X, Y) = \mathrm{POS}(X, \alpha X + (1-\alpha)Y) + \mathrm{POS}(\alpha X + (1-\alpha)Y, Y)$.*

*Proof.* Let $Z = (z_1, \ldots, z_m) = \alpha X + (1-\alpha)Y$ be our convex combination of $X$ and $Y$. We note two properties of the earth mover's distance. Let $a$, $b$, and $c$ by three vectors that consist of nonnegative numbers, where the entries in $b$ and $c$ sum up to the same value. Then, it holds that $\mathrm{EMD}(a+b, a+c) = \mathrm{EMD}(b,c)$. Further, for a nonnegative number $\lambda$, we have that $\mathrm{EMD}(\lambda b, \lambda c) = \lambda \mathrm{EMD}(b,c)$. Using these observations and the definition of the earth mover's distance, we note that:

$$\begin{aligned}\mathrm{POS}(X,Z) &\leq \textstyle\sum_{i=1}^{m} \mathrm{EMD}(x_i, z_i)\\ &= \textstyle\sum_{i=1}^{m} \mathrm{EMD}(x_i, \alpha x_i + (1-\alpha)y_i)\\ &= \textstyle\sum_{i=1}^{m} \mathrm{EMD}((1-\alpha)x_i, (1-\alpha)y_i)\\ &= (1-\alpha)\textstyle\sum_{i=1}^{m} \mathrm{EMD}(x_i, y_i) = (1-\alpha)\mathrm{POS}(X,Y).\end{aligned}$$

The last equality follows by our assumption regarding $X$ and $Y$. By an analogous reasoning we also have that $\mathrm{POS}(Z,Y) \leq \alpha \mathrm{POS}(X,Y)$. By putting these two inequalities together, we have that:

$$\mathrm{POS}(X,Z) + \mathrm{POS}(Z,Y) \leq \mathrm{POS}(X,Y).$$

By the triangle inequality, we have that $\mathrm{POS}(X,Y) \leq \mathrm{POS}(X,Z) + \mathrm{POS}(Z,Y)$ and, so, we have that $\mathrm{POS}(X,Z) + \mathrm{POS}(Z,Y) = \mathrm{POS}(X,Y)$. □

# C Missing Material from Section 5

## C.1 Further Details on Normalized Mallows

We start this subsection by providing the missing computational details for our new parameterization that we propose in the main body. Afterwards, we look at where Mallows elections for a specific value of rel-$\phi$ end up on the map of elections for different numbers of candidates.

### C.1.1 Computational Details

We start with some definitions. Let $m \in \mathbb{N}$ and $\phi \in [0,1)$. Moreover, recall that we denote by $\mathrm{expswaps}(m,\phi)$ the expected swap distance between a reference vote $v^*$ and a vote sampled from Mallows model with dispersion parameter $\phi$ and reference vote $v^*$ for $m$ candidates. We denote by $\mathrm{relswaps}(m,\phi)$ the relative expected swap distance, that is, the expected swap distance $\mathrm{expswaps}(m,\phi)$ normalized by the maximum possible number $\frac{m(m-1)}{2}$ of swaps in a vote over $m$ candidates.

In the main part, we proposed a new normalization of Mallows model which, for a given rel-$\phi \in [0,1)$, crucially relies on finding a $\phi \in [0,1)$ such that:

$$\mathrm{relswaps}(m,\phi) = \text{rel-}\phi. \tag{1}$$

We now describe how $\phi$ can be computed from rel-$\phi$. As a first step, we compute $\mathrm{expswaps}(m,\phi)$ for some given $m \in \mathbb{N}$ and $\phi \in [0,1)$ . Recall that, given $\phi \in [0,1)$ and a reference vote $v^*$ over $m$ candidates, the probability of sampling a vote $v$ over $m$ candidates under the Mallows model is:

$$\mathbb{P}_{\phi,v^*}(v) = \frac{1}{Z}\phi^{\kappa(v,v^*)} \tag{2}$$

with normalizing constant $Z = 1 \cdot (1+\phi) \cdot (1+\phi+\phi^2) \cdot \cdots \cdot (1+\cdots+\phi^{m-1})$. Next, we need to compute the number of different votes at some swap distance from a given vote over $m$ candidates. For this, we create a table $T$ and let $T[m,i]$ denote the number of different votes at some given swap distance $i \in [\frac{m(m-1)}{2}]$ from some fixed vote over $m$ candidates. Notably,

$T[m,i]$ corresponds to the number of permutations over $m$ elements containing $i$ inversions, a well-studied combinatorial problem ([17]). Unfortunately, no closed form expression to compute this is known. Instead, one needs to resort to dynamic programming ([17]): First, we initialize the table with $T[m,0]=1$. Subsequently, for increasing $m$, we update the table starting from $i=0$ and going to $i=\frac{m(m-1)}{2}$ as:

$$T[m,i]=T[m,i-1]+T[m-1,r]-T[m-1,r-m].$$

After precomputing $T$, we are able to compute expswaps$(\phi,m)$ using Equation (2) as:

$$\text{expswaps}(\phi,m)=\frac{1}{Z}\sum_{i=0}^{\frac{m(m-1)}{2}}T[m,i]\cdot\phi^i. \tag{3}$$

The idea behind this equation is that, for each $i\in[\frac{m(m-1)}{2}]$, there exist $T[m,i]$ different votes at swap distance $i$ from the reference vote, each being sampled with probability $\frac{1}{Z}\phi^i$. Using this, we are now able to reformulate Equation (1): Given some $m\in\mathbb{N}$ and rel-$\phi\in[0,1)$, find $\phi$ such that:

$$-1+\frac{2}{\text{rel-}\phi\cdot Z\cdot m(m-1)}\sum_{i=0}^{\frac{m(m-1)}{2}}T[m,i]\cdot\phi^i=0.$$

Note that the left hand side of this equation is simply a polynomial in $\phi$ of maximum degree $\frac{m(m-1)}{2}$. Thus, solving Equation (1) reduces to finding the root of a higher degree polynomial. While no general formula for finding such a root exists, there exists a whole branch of literature focusing on numerical methods for solving this problem, e.g., Newton's method ([19], Chapter 9).[4]

### C.1.2 Mallows Elections for Different Numbers of Candidates

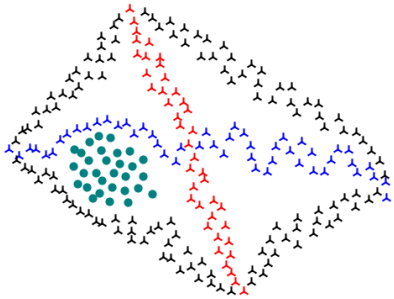
As we only looked at the special case $m=10$, it is, in principle questionable whether our recommendations which dispersion parameter to choose presented in Section 5.4 are also valid for $m>10$ candidates. As mentioned in the main body, we suspect that real-life elections of similar types end up at similar places on the map of elections with our compass on it for different numbers of candidates. Thus, it remains to check whether elections generated from Mallows model with some fixed rel-$\phi$ for different numbers of candidates also end up in similar positions on the respective maps. We confirm this hypothesis using Figure 5, where we depict the map of elections for $m\in\{5,10,20,50\}$ including the four compass matrices, the paths between them, and several elections generated using Mallows model with rel-$\phi=0.375$ (we look at rel-$\phi=0.375$ because this is the best parameter to model the quite homogeneous group of political elections). All maps look quite similar and elections generated using the Mallows model indeed always end up in very similar places.

## C.2 Generation, Selection, and Preprocessing of Real-Life Datasets

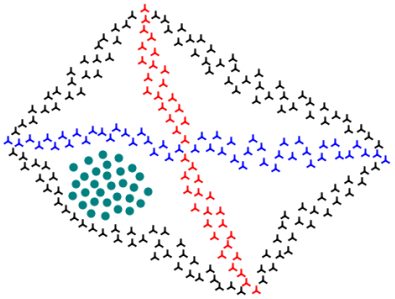
### C.2.1 Generation and Selection of Datasets

As part of our work, we generate three new datasets from sports competitions, that are, Tour de France elections, Giro d'Italia elections, and speed skating elections. As we are interested in elections with complete votes, as a preprocessing step, we always delete votes

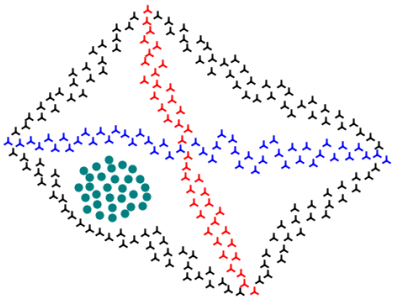
[4] In our python implementation, we used the root() method from the numpy library, which relies on computing the eigenvalues of the companion matrix ([12]).



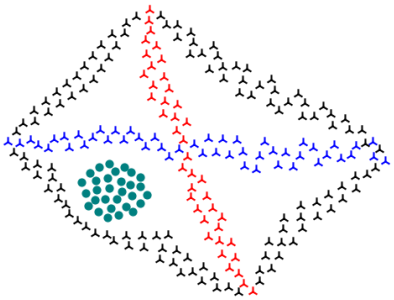
(a) Mallows elections for $m = 5$.



(b) Mallows elections for $m = 10$.



(c) Mallows elections for $m = 20$.



(d) Mallows elections for $m = 50$.

Figure 5: Visualization of our four compass matrices and the connecting paths for different numbers of candidates. Additionally, several elections randomly generated from Mallows model with parameter rel-$\phi = 0.375$ for the respective number of candidates are included.

and candidates until each candidate is contained in at least 70% of all votes and each vote contains 70% of all candidates.

Both the Tour de France and Giro d'Italia are annual cycling competitions consisting of multiple stages. For each of the past 100 editions, we create a separate election with the riders as the candidates and the stages as the voters (where each voter ranks the candidates according to the finish times of the riders in the corresponding stage). To generate the elections, we use publicly available data from https://www.procyclingstats.com. The third new dataset consists of different elections modelling speed skating competitions. For this, we use data from https://results.sporthive.com/ and select 51 speed skating races. Notably, each race consists of multiple laps. For each race, we create a separate election with the speed skaters as the candidates and the laps as the voters (where each voter ranks the candidates according to the lap time of the speed skaters in the corresponding lap).

In addition, our main source of real-life elections is the PrefLib database ([15]). We categorize all election datasets from PrefLib in Table 5. As discussed in the main body, we want to compare elections with ten candidates. Moreover, as our model only allows to consider complete votes without ties, we are interested in datasets where votes are as complete as possible and contain only a few ties. Based on these criteria, our decision which PrefLib datasets to include is displayed in Table 5.

### C.2.2 Preprocessing of Datasets

For all elections from our selected datasets containing incomplete votes (i.e., votes where some of the top candidates are ranked and the remaining candidates are not), we need to

| PrefLib ID | Name | Size | $m$ | Type | Selected | Reason to reject |
|---|---|---|---|---|---|---|
| 1 | Irish | 3 | 9,12,14 | soi | yes | - |
| 2 | Debian | 8 | 4-9 | toc | no | Too few candidates |
| 3 | Mariner | 1 | 32 | toc | no | Too many ties |
| 4 | Netflix | 200 | 3,4 | soc | no | Too few candidates |
| 5 | Burlington | 2 | 6 | toi | no | Too few candidates |
| 6 | Skate | 48 | 14-30 | toc | yes | - |
| 7 | ERS | 87 | 3-29 | soi | yes | - |
| 8 | Glasgow | 21 | 8-13 | soi | yes | - |
| 9 | AGH | 2 | 7,9 | soc | no | Too few candidates |
| 10.1 | Formula | 48 | 22-62 | soi | no | Incomplete and few votes |
| 10.2 | Skiing | 2 | ∼50 | toc | no | Few votes and many ties |
| 11.1 | Webimpact | 3 | 103, 240, 242 | soc | no | Too many candidates and too few votes (∼5) |
| 11.2 | Websearch | 74 | 100-200, ∼2000 | soi | no | Too few votes (∼4) |
| 12 | T-shrit | 1 | 11 | soc | yes | - |
| 13 | Anes | 19 | 3-12 | toc | no | Too many ties |
| 14 | Sushi | 1 | 10 | soc | yes | - |
| 15 | Clean Web | 79 | 10-50, ∼200 | soc | no | Too few votes (∼4) |
| 16 | Aspen | 2 | 5,11 | toc | yes | - |
| 17 | Berkeley | 1 | 4 | toc | no | Too few candidates |
| 18 | Minneapolis | 4 | 7,9,379,477 | soi | no | Incomplete votes |
| 19 | Oakland | 7 | 4-11 | toc | no | Incorrect data (votes like: 1,1,1) |
| 20 | Pierce | 4 | 4,5,7 | toc | no | Too few candidates |
| 21 | San Francisco | 14 | 4-25 | toc | no | Incorrect data (votes like: 1,1,1) |
| 22 | San Leonardo | 3 | 4,5,7 | toc | no | Too few candidates |
| 23 | Takoma | 1 | 4 | toc | no | Too few candidates |
| 24 | MT Dots | 4 | 4 | soc | no | Too few candidates |
| 25 | MT Puzzles | 4 | 4 | soc | no | Too few candidates |
| 26 | Fench Presidential | 6 | 16 | toc | no | Approval ballots |
| 27 | Proto French | 1 | 15 | toc | no | Approval ballots |
| 28 | APA | 12 | 5 | soi | no | Too few candidates |
| 29 | Netflix NCW | 12 | 3,4 | soc | no | Too few candidates |
| 30 | UK labor party | 1 | 5 | soi | no | Too few candidates |
| 31 | Vermont | 15 | 3-6 | toc | no | Approval ballots |
| 32 | Cujae | 7 | 6,32 | soc/soi/toc | no | Many reasons |
| 33 | San Sebastian Poster | 2 | 17 | toc | no | Approval ballots |
| 34 | Cities survey | 2 | 36, 48 | soi | yes | - |

Table 5: Overview of all election datasets that are part of the PrefLib database. "Size" stands for the number of elections in the dataset, "$m$" for the number of candidates and "Type" for the type of the votes in the dataset (soc means that all votes are strict complete orders; soi means that all votes are strict incomplete orders; toc means that all votes are weak incomplete orders).

fill-in the votes. For the decision how to complete each vote, we use the other votes as references assuming that voters that rank the same candidates on top also continue to rank candidates similarly towards the bottom. For each incomplete vote $v$, we proceed as follows. Let us assume that the length of the vote $v$ is $n$. Let $V_P$ be the set of all original votes of which $v$ is a prefix. We uniformly at random select one vote $v_p$ from $V_P$ and then we add candidate $c = \text{pos}(v_p, n+1)$ at the end of vote $v$. We repeat the procedure until vote $v$ is complete. If the set $V_P$ is empty, then we choose $c$ uniformly at random (from those candidates that are not part of $v$ yet). Moreover, if a vote contains ties (i.e., pairs or larger sets of candidates that are reported as equally good, except for the case that would fit the description of an incomplete vote above), we break them randomly.

After applying these preprocessing steps, we arrive at a collection of datasets containing elections with ten or more candidates and complete votes without ties. As we focus on ten candidates, we need to select a subset of ten candidates for each election: We select the ten candidates with the highest Borda score.

In Table 6 we present a detailed description of the resulting selected datasets. In some datasets only parts of the data meets our criteria. For example, in the dataset containing Irish elections we have three different election, but one of the them (an election from West Dublin) contains only 9 candidates. We delete all such elections. After doing so, we finally arrive at eleven real-life datasets containing elections meeting our criteria.

| Source | Category | Name | # Selected Elections | Avg $m$ | Avg $n$ | Description |
|---|---|---|---|---|---|---|
| Preflib | Political | Irish | 2 | 13 | $\sim 54011$ | Elections from North Dublin and Meath |
| Preflib | Political | Glasgow | 13 | $\sim 11$ | $\sim 8758$ | City council elections |
| Preflib | Political | Aspen | 1 | 11 | 2459 | City council elections |
| Preflib | Political | ERS | 13 | $\sim 12$ | $\sim 988$ | Various elections held by non-profit organizations, trade unions, and professional organizations |
| Preflib | Sport | Figure Skating | 40 | $\sim 23$ | 9 | Figure skating |
| This paper | Sport | Speed Skating | 13 | $\sim 14$ | 196 | Speed skating |
| This paper | Sport | TDF | 12 | $\sim 55$ | $\sim 22$ | Tour de France |
| This paper | Sport | GDI | 23 | $\sim 152$ | 20 | Giro d'Italia |
| Preflib | Survey | T-Shirt | 1 | 11 | 30 | Preferences over T-Shirt logo |
| Preflib | Survey | Sushi | 1 | 10 | 5000 | Preferences over Sushi |
| Preflib | Survey | Cities | 2 | 42 | 392 | Preferences over cities |

Table 6: Each row contains a description of one of the real-life datasets we consider. In the column "# Selected Elections", we denote the number of elections we finally select from the respective dataset.

However, as we cannot include all elections from each dataset on the map of elections, we further reduce the number of elections by considering only selected elections. In Table 6, we include in the column "# Selected Elections" the number of elections we selected from each dataset in the end. We based our decision on the selection of elections on the number of voters and candidates. That is, for ERS, we only take election with at least 500 voters, for Speed Skating with at least 80, for TDF with at least 20, and for Figure skating with at least 9. Beside that for TDF, we only select elections with no more than 75 candidates. We refer to the resulting datasets as *intermediate* datasets.

### C.2.3 Sampling Elections from Intermediate Datasets

We treat each of our intermediate real-life datasets as a separate election model from which we sample 15 elections to create the final datasets that we use in the paper. For each intermediate dataset, we sample elections as follows. First, we randomly select one of the elections. Second, we sample 100 votes from the election uniformly at random (this implies that for elections with less than 100 votes, we select some votes multiple times, and for elections with more than 100 votes, we do not select some votes at all). We do so to make full use of elections with far more than 100 votes. For instance, our Sushi intermediate dataset contains only one election consisting of 5000 votes. Sampling an election from the Sushi intermediate dataset thus corresponds to drawing 100 votes uniformly at random from the set of 5000 votes. On the other hand, for intermediate datasets containing a higher number of elections, e.g., the Tour de France intermediate dataset, most of the sampled elections come from a different original election.

After executing this procedure, we arrive at 11 datasets each containing 15 elections consisting of 100 complete and strict votes over 10 candidates, which we use for our experiments.