

# Learning How to Vote With Principles

## Axiomatic Insights Into the Collective Decisions of Neural Networks

Levin Hornischer<sup>1</sup> and Zoi Terzopoulou<sup>2</sup>

<sup>1</sup>Munich Center for Mathematical Philosophy, LMU Munich  
<sup>2</sup>GATE, Saint-Etienne School of Economics

**Abstract** Can neural networks be applied in voting theory, while satisfying the need for transparency in collective decisions? We propose *axiomatic deep voting*: a framework to build and evaluate neural networks that aggregate preferences, using the well-established axiomatic method of voting theory. Our findings are: (1) Neural networks, despite being highly accurate, often fail to align with the core axioms of voting rules, revealing a disconnect between mimicking outcomes and reasoning. (2) Training with axiom-specific data does not enhance alignment with those axioms. (3) By solely optimizing axiom satisfaction, neural networks can synthesize new voting rules that often surpass and substantially differ from existing ones. This offers insights for both fields: For AI, important concepts like bias and value-alignment are studied in a mathematically rigorous way; for voting theory, new areas of the space of voting rules are explored.\*

**Keywords** Computational social choice, machine learning, neural networks, voting theory, axiomatic method, semantic loss function, data augmentation.

### 1. Introduction

Artificial intelligence (AI) is increasingly applied in many domains, including not just scientific and technological but also societal domains. This poses a dilemma when it comes to *social choice*, i.e., voting, preference aggregation, and other processes of collective decisions. On the one hand voting systems should be transparent, but the neural networks on which modern AI is built are notoriously opaque. On the other hand neural networks could unearth novel and tailor-made collective decision procedures. Already, state-of-the-art techniques for alignment of Large Language Models (LLMs) with human values—like RLHF (Bai et al., 2022) or DPO (Rafailov et al., 2024)—rely on the aggregation of human preferences about the generated outputs to fine-tune LLMs. This triggered recent research in guiding such AI alignment using social choice (Conitzer et al., 2024).

In this paper, we study how neural networks aggregate votes and preferences. When they form such collective decisions, do they adhere to the normative principles that social choice theory formulated as axioms? This is fundamental both for a discussion of the dilemma and for using social choice for AI alignment. Moreover, it offers new insights for both AI and voting theory. For AI, this provides a rich testing ground to study pressing machine learning concepts like bias, value-alignment and interpretability in a mathematically rigorous way. For example, a network is not biased towards specific individuals if it aggregates their preferences in accordance with the axiom of anonymity; the so-called Pareto principle requires the neural network to align with any preference shared among all individuals; and the well-known axiom of independence entails a certain compositional interpretability of the network. For voting theory, axiomatic deep voting provides a new method for the central quest of exploring the space of voting rules.

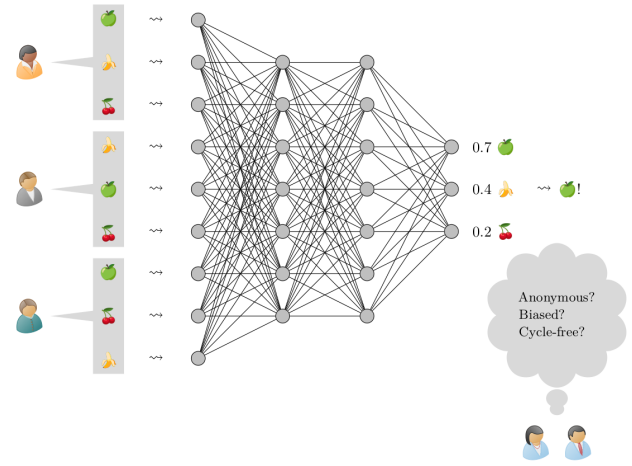


Figure 1: Can neural networks learn to vote with principles?

**Social choice.** How are individual preferences best turned into a collective decision? This question is studied by *social choice theory* (Brandt et al., 2016; List, 2022) and, specifically, *voting theory* (Zwicker, 2016). A *voting rule* is a function that takes as input a *profile*—i.e., a list of each individual’s preferences among a given set of alternatives—and produces as output a *collective decision*, i.e., the alternative(s) that the rule takes to be most preferred for the group as a whole (see Section 3 for the formal definitions). The most straightforward rule is *Plurality* (which picks the alternative that is considered best by the most individuals); other classic rules include *Borda* and *Copeland*, while a more recent suggestion is *Split Cycle*.

**Axiomatic deep voting.** To study the collective choices of neural networks, we develop the *axiomatic deep voting* framework (sketched in Figure 1). Deep neural networks are (parametrized) functions that map vectors (typically of a high dimension) to vectors (typically of a low dimension). So, after suitably *encoding* profiles and collective decisions as vectors, neural networks realize voting rules, i.e., functions from profiles to collective decisions. Discovering a voting rule can then be seen as an *optimization problem*: updating the neural network parameters until a given desired property is fulfilled. We *evaluate* a trained neural network in terms of accuracy and axiom satisfaction. While the former is standard in machine learning, the latter is specific to voting theory and its *axiomatic method* (Thomson, 2001; List, 2011). Different axioms describe different desirable properties of voting rules. An example is the already mentioned anonymity axiom which requires that the names of the voters should *not* influence the collective decision.

**Research questions.** With this framework, we investigate three specific questions.

\*The source code will eventually be made available here: <https://github.com/LevinHornischer/AxiomaticDeepVoting>.

- (1) *Correct for the right reasons?* Neural networks can accurately learn standard voting rules, but do they adhere to the normative principles expressed by voting-theoretic axioms?

We observe eminent violations of the axioms, despite high accuracy in mimicking voting rules. So we focus on teaching neural networks the expert knowledge expressed by axioms. There are two common ways to do this. The first is via *dataset augmentation* (Xia, 2013):

- (2) *Learning principles by example?* Can neural networks be trained to adhere to voting-theoretic axioms by training with data exemplifying the axioms?

The second way is via *semantic loss functions* (Xu et al., 2018). For this, we develop a translation of the axioms into loss functions; so, by optimizing this loss during training, the network increases the corresponding axiom satisfaction. Importantly, though, perfect axiom satisfaction is impossible according to the infamous theorem by Arrow (1951). So we search for the best possible axiom satisfaction:

- (3) *Rule synthesis guided by principles?* When neural networks optimize axiom satisfaction, can they develop new voting rules that surpass existing ones?

We compare the discovered rules to a wide range of known voting rules, to test if neural networks can advance the current state of the art in voting theory.

**Key findings.** We test three paradigmatic neural network architectures: multi-layer perceptrons, convolutional neural networks, and word embedding based classifiers. We also check four standard distributions of voter preferences. We find the following:

- (1) Our employed architectures demonstrate similar behavior both regarding accuracy and axiom satisfaction. Importantly, despite high accuracy, they markedly violate critical axioms like anonymity—yet, the news is not as bad for other axioms.
- (2) Data augmentation does not seem to boost the principled learning of neural networks. However, it drastically decreases the amount of required training data.
- (3) Neural networks that perform the unsupervised learning task of optimizing axiom satisfaction discover voting rules that are substantially different from existing ones and are comparable—and often better—in axiom satisfaction.

Thus, we fruitfully combine two approaches to studying the space of voting rules: Drawing on machine learning, we use neural networks qua universal function approximators to *explore* that space; and drawing on voting theory, we *evaluate* points in that space—i.e., voting rules—by their axiom satisfaction, thus guiding the exploration.

## 2. Related Work

We identify three main streams of relevant literature.

### 2.1. Axiomatic Evaluation of Voting Rules

Social choice theory has extensively quantified the axiom satisfaction of various voting rules, with a significant focus on the concept of *manipulability*, i.e., the propensity of voters to be untruthful in order to sway the outcome in their favor (Favardin et al., 2002; Favardin and Lepelley, 2006; Nitzan, 1985). Numerous studies (Fishburn and Gehrlein, 1982; Merrill, 1984; Nurmi, 1988) examine how often voting rules elect the *Condorcet winner* (that is, the alternative representing a majoritarian consensus) for relatively small elections all having the same probability of materializing (that is, assuming the Impartial Culture distribution). In line with our findings, the Borda rule is found to elect the Condorcet winner more often than the Plurality

rule (Nurmi, 1988). When considering the axiom of independence—the main trigger of Arrow’s impossibility theorem—the Borda rule fulfills it more frequently than Copeland, which in turn satisfies it more than Plurality (Dougherty and Heckelman, 2020). For the special case of 3 voters and 3 alternatives, an anonymous voting rule satisfies independence between 1.3% and 25.5% of the time (Powers, 2007).

Overall, our work aligns with the traditional concept of evaluating voting rules based on axioms. However, we also consider learning voting rules and not just evaluating existing ones.

### 2.2. Neural Networks and Voting

The synergy between voting and machine learning has recently garnered more and more attention. Kujawska et al. (2020) use, among others, multi-layer perceptrons (MLPs) on elections of 20 alternatives and 25 voters to predict the winners of different voting rules. The study’s primary aim is to identify an effective computational technique on top of the classical ones of the voting literature. The authors find that the Borda rule is predicted by the neural networks with high accuracy (up to 99%), but more complex rules are predicted with lower accuracy (up to 85% for Kemeny and 89% for Dodgson). Burka et al. (2022) employ MLPs to investigate the relation between sample size and accuracy when learning different voting rules, including Plurality, Borda, and Copeland. In that work, up to 3000 data points are used based on the Impartial Culture assumption, with at most 5 alternatives and 11 voters. The MLP is found to mimic more closely Borda, no matter on which rule it is trained: e.g., for 3 alternatives and 7 voters, trained on Plurality, the MLP mimicked Borda with 95% accuracy and Plurality with 86% accuracy. However, the size of the training data exhibits an impact on the results: e.g., when trained on elections with a Condorcet winner, the MLP mimics more closely Borda in sample-size up to 1000, and Copeland in larger samples. Increasing the size of the MLP by adding layers does not seem important. Anil and Bao (2021) study more complex neural network architectures (such as Set Transformers and DeepSets), improving the accuracy of MLPs by up to 4% in learning Plurality and Copeland. With sufficiently many data points, those networks are shown to match almost perfectly each voting rule, and to also generalize to elections with an unseen number of voters.

Similarly to all these works, our first experiment considers precisely the problem of using neural networks to learn existing rules from voting theory. However, we systematically study this with axioms: instead of only targeting the right outcomes, we test whether they are obtained via the right principles.

In an initial exploration towards the same direction, Armstrong and Larson (2019) use a single axiom—prescribing the election of a Condorcet winner when one exists—to train normatively appealing neural networks. This work relies on real data from Canadian federal elections, while ours builds on extensive synthetic data. Additionally, our third experiment illustrates original interactions between sets of different axioms that have not been explored in the literature yet. However, one of the more intricate voting principles not tackled in our paper is fairness. After observing a theoretical trade-off between fairness and certain notions of economic efficiency (some related to the Condorcet winner), Mohsin et al. (2022) train two machine learning models on synthetic data and discover new voting rules that compete well against both Plurality and Borda. Although rule synthesis and axiomatic analysis is not a main focus of that work, the obtained results enforce the idea that machine learning methods can beat existing ones from economic theory when optimized for principled learning.

Other promising lines of research target learning an abstract voting rule given examples about its choices (Procaccia et al., 2009) and designing a voting rule that maximizes some notion of social welfare (Anil and Bao, 2021). Holliday et al. (2024) explore the strategic manipulation of voting rules by MLPs of different sizes, generating elections of up to 6 alternatives and 21 voters. They find that

1	2	3	4	5	6
c	d	d	c	a	d
a	b	b	b	b	a
b	c	a	d	c	b
d	a	c	a	d	c

Figure 2: A voting profile, with voters  $N = \{1, \dots, 6\}$  and alternatives  $A = \{a, b, c, d\}$ . Each column depicts the preference of the individual voter; e.g., voter 1 prefers alternative c most, followed by alternative a, etc.

sufficiently large MLPs learn to profitably manipulate all examined voting rules only with information about the pairwise majority victories between alternatives. But some rules like Split Cycle seem more resistant than other rules (e.g., Plurality and Borda).

A different approach, rather orthogonal to ours, is to consider AI models as the individuals who vote, instead of using them as the aggregation mechanisms. In this vein, Yang et al. (2024) consider a human voting experiment with 180 participants to establish a baseline for human preferences and conducted a corresponding experiment with LLM (e.g., GPT-4) agents. The voting behavior of the networks seems to be affected by the presentation order of the alternatives, as well as the numerical ID assigned to each LLM representing a voter. Some voting rules such as Borda show that LLMs may lead to less diverse collective outcomes. Importantly GPT-4 seems to over-rely on stereotypical demographics of the voters it is supposed to mimic. Similarly, using data from Brazil’s 2022 presidential election, Gudiño Rosero et al. (2024) tests the accuracy with which LLMs predict an individual’s vote. They find that LLMs are more accurate than a naive rule guessing that individuals simply vote for the proposals of the candidate most aligned with their political orientation.

### 2.3. Social Choice for AI Alignment

A growing research area studies how social choice theory can be used to guide the alignment of modern AI methods with human values and moral judgments. Conitzer et al. (2024) highlight a series of technical connections—for example, the alternatives in a voting context could be treated as all possible parameterizations of a network, or as all its possible answers. As an indication, in a popular work about a controversial topic, Noothigattu et al. (2018) use data from the online ‘moral machine experiment’ to build a model of aggregated moral preferences aimed at guiding the decision making of autonomous vehicles. On a more theoretical level, Mishra (2023) utilize Arrow’s theorem to prove that there does not exist any AI system that can treat all its users and human supervisors equally. We do not directly engage with the ethical dimension of this research area; still, we participate in the related foundational discussion by studying whether neural networks can learn to vote with principles.

### 3. Preliminaries on Voting Theory

We work in the standard setting of voting theory, where a finite set  $N$  of voters  $N$  have preferences, which are linear orders (also called *rankings*) over a finite set  $A$  of *alternatives* (Zwicker, 2016). Set  $m := |A|$  and  $n := |N|$ . We denote by  $\mathbf{P} = (P_1, \dots, P_n)$  a preference *profile*, i.e., a vector with the preference  $P_i$  for every voter  $i \in N$ . This is illustrated in Figure 2.

For a permutation of the alternatives  $\sigma : A \rightarrow A$ , the ranking  $\sigma(\mathbf{P})$  is obtained by applying  $\sigma$  elementwise to the ranking  $\mathbf{P}$ , and  $\sigma(\mathbf{P}) = (\sigma(P_1), \dots, \sigma(P_n))$ . For a permutation of the voters  $\pi : N \rightarrow N$ , we define  $\pi(\mathbf{P}) = (P_{\pi(1)}, \dots, P_{\pi(n)})$ . A *voting rule* is a function  $F$  that determines the winning alternatives for each such profile. Formally,  $F : \mathbf{P} \mapsto S$ , where  $\emptyset \neq S \subseteq A$ .<sup>1</sup>

<sup>1</sup>We use the Python package `pref-voting` in all our experiments.

### 3.1. Voting Rules

Voting rules usually fit into one of two categories: scoring rules and tournament solutions. *Scoring rules* assign a score to each alternative depending on its position in the linear preference of each voter and declare as winners those alternatives with the highest score across all voters. The two primary scoring rules are *Plurality* (assigning score 1 to an alternative each time it is ranked first by a voter, and score 0 otherwise) and *Borda* (assigning score  $m - 1$  to an alternative ranked first by a voter, score  $m - 2$  to an alternative ranked second, and so on, until score 0 is assigned to an alternative ranked last by a voter).<sup>2</sup>

*Tournament solutions* on the other hand are based on tournaments that capture pairwise comparisons between the alternatives, induced by the voters’ preferences. For  $x, y \in A$ , let  $N_{x>y}^{\mathbf{P}}$  be the set of voters  $i$  in the profile  $\mathbf{P}$  that consider  $x$  better than  $y$  in  $P_i$ , and  $n_{x>y}^{\mathbf{P}} := |N_{x>y}^{\mathbf{P}}|$ . A characteristic tournament solution is the *Copeland* rule, which selects as winners the alternatives that beat the most other alternatives in a pairwise majority contest:  $\arg \max_{x \in A} |\{y \in A : n_{x>y}^{\mathbf{P}} \geq n_{y>x}^{\mathbf{P}}\}|$ . The weighted tournament of a profile is a weighted directed graph the nodes of which are alternatives with an edge from  $x$  to  $y$  of weight  $n_{x>y}^{\mathbf{P}}$ . Suppose that in each cycle of the graph, we simultaneously delete the edges with minimal weight. Then the alternatives with no incoming edges are the winners of *Split Cycle* (Holliday and Pacuit, 2023a). If there is only one Split Cycle winner in a profile  $\mathbf{P}$ , then this also is the winner of *Stable Voting*; otherwise  $x$  is a winner of Stable Voting if for some alternative  $y$  it holds that  $x$  is a Split Cycle winner with the maximal margin  $n_{x>y}^{\mathbf{P}}$  such that  $x$  is a Stable Voting winner in the profile  $\mathbf{P}_{-y}$  obtained from  $\mathbf{P}$  after deleting alternative  $y$  (Holliday and Pacuit, 2023b).

Two prominent rules that do not fit into the two above categories are *Blacks* and *Weak Nanson*. Black returns the Condorcet winner (i.e., the alternative beating every other alternative in a pairwise strict majority contest) if one exists, otherwise it returns the Borda winners. Weak Nanson is defined iteratively on voting profiles of various sizes. In each round, all alternatives with Borda score at most as high as the average Borda score are removed. Whenever all alternatives have the same Borda score, they all win; otherwise the alternative that remains in the last round wins. In our third experiment, we use even more common voting rules for detailed comparisons, though they are not essential to the paper; hence we refer to the introductory chapter of Zwicker (2016) and to the `pref-voting` documentation for the relevant definitions.

### 3.2. Axioms

We define axioms as functions that map a voting rule and a preference profile to a value in  $\{-1, 1, 0\}$ , where 0 means that the axiom is not applicable,  $-1$  means that the desideratum is violated, and 1 that it is satisfied. The *satisfaction degree* of an axiom is the ratio of the number of sampled profiles in which the axiom is satisfied to the number of sampled profiles in which it is applicable. We focus on axioms that capture basic and diverse normative properties of a voting rule  $F$ .

- *Anonymity* is always applicable; it is satisfied in  $\mathbf{P}$  if for all permutations of voters  $\pi : N \rightarrow N$ ,  $F(\pi(\mathbf{P})) = F(\mathbf{P})$ . In words, the winners should be invariant under permutations of the voters.
- *Neutrality* is always applicable; it is satisfied in  $\mathbf{P}$  if for all permutations of alternatives  $\sigma : A \rightarrow A$ ,  $F(\sigma(\mathbf{P})) = \sigma(F(\mathbf{P}))$ . In words, under permutations of the alternatives, the winners should be permuted respectively.
- *Condorcet principle* is applicable in  $\mathbf{P}$  if some  $x \in A$  is such that  $n_{x>y}^{\mathbf{P}} > n/2$  for all  $y \in A \setminus \{x\}$ ; it is satisfied if  $F(\mathbf{P}) = \{x\}$ . In words, if a Condorcet winner exists, then it should be the unique winner of the voting rule.

<sup>2</sup>Plurality and Borda are often contrasted in voting (Hatzivellkos, 2018; Terzopoulou, 2023).

- *Pareto principle* is applicable in  $\mathbf{P}$  if there exist two alternatives  $x, y \in A$  such that  $n_{x>y}^{\mathbf{P}} = n$ ; it is satisfied if  $y \notin F(\mathbf{P})$ . In words, if an alternative is considered inferior to a certain other alternative by all voters, then it should not win.
- *Independence* is applicable in  $\mathbf{P}$  if  $F(\mathbf{P}) \neq A$ ; it is satisfied if for all  $x \in F(\mathbf{P})$ ,  $y \notin F(\mathbf{P})$ , and  $\mathbf{P}'$  such that  $N_{x>y}^{\mathbf{P}} = N_{x>y}^{\mathbf{P}'}$ , it holds that  $y \notin F(\mathbf{P}')$ . In words, if the relative ranking between a winning alternative and a losing alternative remains the same for all voters, then the losing alternative should not win.

All voting rules defined above satisfy anonymity and neutrality, as well as the Pareto principle, for all preference profiles. They all violate independence for some preference profile. Copeland, Split Cycle, Stable voting, Blacks rule and Weak Nanson always satisfy the Condorcet principle.

### 3.3. Distributions of Preference Profiles

Specifying the distribution of preference data is essential to studying the voting behavior of a society. To ensure that our results are independent of the specific choice of the distribution, we employ four different ones. (For a detailed comparison of the various voting distributions, see Boehmer et al. (2024).)

*Impartial Culture (IC)* assumes that all preference profiles have the same probability of appearing. Each preference of a voter in a profile is sampled uniformly at random. The *Mallows* distribution (Mallows, 1957) fixes a reference ranking  $\mathbf{P}$  and assumes that each voter’s preference is close to that ranking. Closeness to the reference ranking is defined using the Kendall-tau distance, parameterized by a dispersion parameter  $\phi \in (0, 1]$ . This distribution reduces to IC when  $\phi = 1$  and concentrates all mass on  $\mathbf{P}$  as  $\phi$  tends to 0.

The IC and Mallows distributions are complementary: IC is simplistic and widely employed in theoretical works on voting rules; it captures an extreme case with no correlation between preferences of voters. Mallows is often employed in numerical studies of voting rules that use artificial data but wish to capture more realistic voting scenarios (Caragiannis and Micha, 2017; Lee et al., 2014).

The next two distributions also capture more intricate relationships between the preferences in a profile. According to the *2D-Euclidean* distribution, voters and alternatives are distributed randomly in 2-dimensional Euclidean space, and the closer an alternative is to a voter the more the voter prefers that alternative. Finally, the *Urn* distribution (Eggenberger and Pólya, 1923) generates a profile given a parameter  $\alpha \in [0, \infty)$ . Voters randomly draw their ranking from an urn. Initially, the urn includes all possible rankings over the alternatives. After a voter randomly draws from the urn, we add to the urn  $\alpha n!$  copies of that ranking. When  $\alpha = 0$ , this reduces to IC.

## 4. Method

To answer our research questions, we develop the *axiomatic deep voting* framework, visualized in Figure 3. It is built around a neural network, which is a function  $f_w : \mathbb{R}^i \rightarrow \mathbb{R}^j$  parametrized by weights  $w \in \mathbb{R}^k$ . We will instantiate this with three different neural network architectures (see Section 4.1). Every profile  $\mathbf{P}$  is mapped, via an *encoding* function  $e$  (see Section 4.2), to a vector  $x = e(\mathbf{P}) \in \mathbb{R}^i$ , for which the neural network produces an output  $\hat{y} \in \mathbb{R}^j$ .<sup>3</sup> The *decoding* function  $d$  (see Section 4.3) turns this output into a winning set  $S = d(\hat{y})$ . Thus, this setup realizes the voting rule:

$$F_w(\mathbf{P}) := d\left(f_w(e(\mathbf{P}))\right).$$

The network is trained, as usual, using backpropagation with respect to a *loss function* (in Section 4.4), which relies on training data. Finally, we *evaluate* (in Section 4.5) the trained network not only with respect

<sup>3</sup>For our third architecture, the encoding function is part of the neural network, while for the first two it is independent (see Section 4.2); hence we treat  $e$  as a separate entity here.

to its accuracy (how well it fits the test dataset), but, crucially, also by how much it satisfies the various voting axioms.

### 4.1. Architectures

We use three paradigmatic neural network architectures from modern machine learning.

First, *multi-layer perceptrons* (MLPs)—also known as feed-forward neural network—are the classic deep neural net (see, e.g., Goodfellow et al., 2016, ch. 6). They consist of an input layer of neurons, one or more hidden layers, and an output layer.

Second, *convolutional neural networks* (CNNs) are a standard architecture to process grid-like input data such as images (see, e.g., Goodfellow et al., 2016, ch. 9), and in our case profiles. Compared to MLPs, they additionally use so-called convolutional layers to capture local, invariant patterns in the input.

Third, we devise an architecture that satisfies the anonymity axiom by design: We view profiles as sentences whose words are the rankings. We use the *word embedding* algorithm *Word2vec* (Mikolov et al., 2013) to map each ranking to a high-dimensional embedding vector. These vectors are averaged—hence we get anonymity—and an MLP then classifies this average into a winning set. This combined architecture we call here *word embedding classifiers* (WECs).

### 4.2. Encoding

To ensure our neural networks learn general patterns, we do not work with a fixed number of voters and alternatives, but only with a maximal number of voters  $n_{\max}$  and a maximal number of alternatives  $m_{\max}$ . So the model should allow as input any profile  $\mathbf{P}$  over the set of voters  $N = \{0, \dots, n-1\}$  with  $n \leq n_{\max}$  and set of alternatives  $M = \{0, \dots, m-1\}$  with  $m \leq m_{\max}$ . (For readability, we also write  $a, b, c, \dots$  for the alternatives.) We write  $\alpha_r^s$  for the  $r$ -th most preferred alternative of voter  $s$ , so the profile  $\mathbf{P}$  is represented as the matrix  $(\alpha_r^s)_{r,s}$ , whose columns are the rankings as in Figure 2. We write  $\tilde{\mathbf{P}} = (\tilde{\alpha}_r^s)_{r,s}$  for the result of padding the  $m \times n$  matrix  $\mathbf{P}$  with the symbol  $\sim$  to the maximal input dimensions  $m_{\max} \times n_{\max}$ . (So  $\tilde{\alpha}_r^s$  is  $\alpha_r^s$  if  $r \leq m$  and  $s \leq n$ , and otherwise it is  $\sim$ .)

How should we encode  $\tilde{\mathbf{P}}$  so it can be inputted to a neural network? The most straightforward way is to read each alternative  $\alpha_r^s \in M$  as the number that it is and the padding symbol  $\sim$  as, say,  $-1$ . Then the matrix  $\tilde{\mathbf{P}}$  is regarded as a vector of dimension  $m_{\max} n_{\max}$ . However, this does not perform well, so, following Anil and Bao (2021), we represent an alternative not as a number but as a one-hot vector. For  $a \in \{0, \dots, m_{\max}-1\}$ , let  $\bar{a}$  be the vector of length  $m_{\max}$  that is 1 at position  $a$  and 0 everywhere else. For the padding symbol, let  $\bar{\sim}$  be the vector of length  $m_{\max}$  that is 0 everywhere. We write  $\bar{\mathbf{P}} = (\bar{\alpha}_r^s)_{r,s}$ .

The *encoding function for MLPs*,  $e_{\text{MLP}}$ , maps profile  $\mathbf{P}$  to the vector  $x$  obtained by casting the matrix  $\bar{\mathbf{P}}$  column by column into a flattened vector (of dimension  $m_{\max}^2 n_{\max}$ ). This vector  $x$  can then be inputted into the MLP.

The *encoding function for CNNs* regards the matrix  $\bar{\mathbf{P}}$  as a pixel image: the ‘pixel’ at position  $(r, s)$  has the ‘color value’  $\bar{\alpha}_r^s$ . Thus,  $e_{\text{CNN}}$  maps profile  $\mathbf{P}$  to the matrix  $\bar{\mathbf{P}}$  recast as a tensor with dimensions  $(\text{channel}, \text{height}, \text{width}) = (m_{\max}, m_{\max}, n_{\max})$ . This tensor can then be inputted into the CNN.

The *encoding function for WECs* regards the profile  $\mathbf{P} = (P_1, \dots, P_n)$  as a sentence with words  $P_i$ . We train it to embed these words into vectors of a fixed high dimension. Thus, unlike the previous encoding functions, this one is not separate from the neural network but rather forms the first layer of the WEC, with the remaining layers processing the embedding vectors. More precisely, we first pre-train the embeddings as follows. For a given corpus size  $c$ , we sample  $c$ -many profiles from a given distribution of profiles (e.g., IC) to form our corpus (i.e., a set of sentences). The rankings occurring in the profiles form the vocabulary of this corpus, to which we add the unk token (to later represent *unknown* rankings,

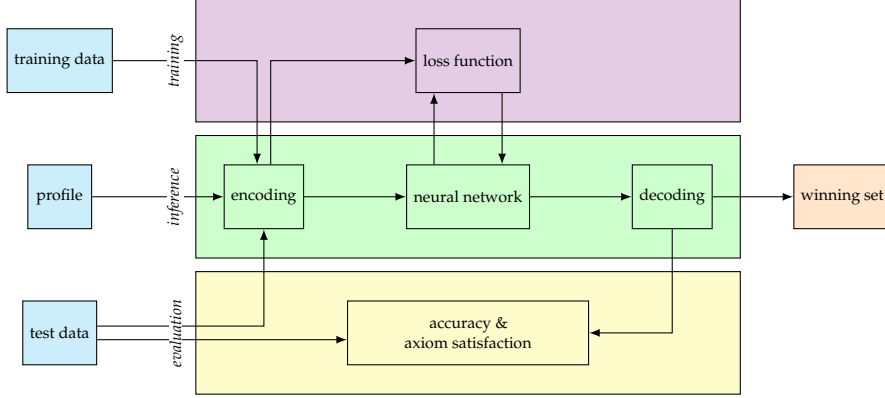


Figure 3: The axiomatic deep voting architecture.

i.e., rankings that are not in the vocabulary) and the *pad* token (to *pad* a profile to length  $n_{\max}$ ). Using Word2vec, we train embeddings which represent words in the vocabulary as vectors. When instantiating the WEC architecture, these embeddings form the first layer: it maps the profile  $(P_1, \dots, P_n)$  to the corresponding embedding vectors  $(v_1, \dots, v_n)$ . The next layer averages these vectors into a single vector  $v$ , followed by several linear layers ending with the output layer.

#### 4.3. Decoding

Given a profile  $\mathbf{P}$  as input, all neural network architectures produce as output the logits  $\hat{y} = (\hat{y}_0, \dots, \hat{y}_{m_{\max}})$  in  $\mathbb{R}^{m_{\max}}$ . We apply the sigmoid function  $\sigma$  elementwise to obtain the probability that alternative  $r$  is in the winning set. With  $m$  the number of alternatives in profile  $\mathbf{P}$ , we define the decoding function

$$d_m(\hat{y}) := \{r \in \{0, \dots, m\} : \sigma(\hat{y}_r) > 0.5\}^4.$$

In experiment 3, we will consider further versions of this decoding function (see Section 6.3).

#### 4.4. Loss Functions

Since multiple alternatives can win, we cast the task of finding a voting rule as a *multi-label classification* problem. Each input profile  $\mathbf{P}$  is associated with  $m$  binary labels (where  $m$  is the number of alternatives in  $\mathbf{P}$ ), and the  $r$ -th label is 1 if and only if the  $r$ -th alternative is in the winning set associated with  $\mathbf{P}$ . Hence we use *binary cross entropy* as loss function.

For each axiom, we also design a loss function that enforces satisfaction of that axiom. Concretely, for the anonymity axiom, this is done as follows. Given the network  $f_w$  and profile  $\mathbf{P}$ , uniformly sample  $N$ -many permutations  $\pi_1, \dots, \pi_N$  of the set of voters of  $\mathbf{P}$  and define

$$L_A(f_w, \mathbf{P}) := \frac{1}{N} \sum_{\tau=1}^N \text{KL}(f_w(e(\mathbf{P})), f_w(e(\pi_\tau(\mathbf{P})))),$$

where KL is Kullback–Leibler divergence. For the other axioms, we proceed similarly (see Appendix A, where we also discuss differentiability).

#### 4.5. Evaluation Metrics

We have two ways of evaluating the model: accuracy and axioms. First, we calculate the accuracy of the trained neural network on a

<sup>4</sup>A priori, it can happen that the neural network does not assign any winner, in contrast to our definition of a voting rule. We check (and train) that this happens, if at all, only with a negligible probability.

given test set in two ways: *Identity* (or *hard*) *accuracy* is the percentage of pairs  $(\mathbf{P}, S)$  in the test set for which  $F_w(\mathbf{P}) = S$ . *Subset* (or *soft*) *accuracy* is defined in the same way but replacing the identity with  $F_w(\mathbf{P}) \subseteq S$ . Second, we calculate the satisfaction degrees for the various axioms of the voting rule  $F_w$  that the trained neural network realizes (see Section 5.3 for the details).

## 5. Experimental Setup

### 5.1. Voting-Theoretic Parameters

We work with  $n_{\max} = 77$  and  $m_{\max} = 7$  and all four profile distributions in the first experiment and with  $n_{\max} = 55$  and  $m_{\max} = 5$  and with IC and Mallows in the other experiments. The first experiment does not show a qualitative difference between these settings, but the latter is computationally more efficient.

We use the Mallows distribution with a parameter  $\text{rel-}\phi$  (randomly generated) that, together with the number of alternatives, determines the value of the dispersion parameter  $\phi$ . According to Boehmer et al. (2021) and Boehmer et al. (2023), this methodology generates data that resemble more closely those of real elections. We use the Urn-R distribution (Boehmer et al., 2021), where, for each generated profile,  $\alpha$  is chosen according to a Gamma distribution with shape parameter  $k = 0.8$  and scale parameter  $\theta = 1$ . The other distributions do not need further parameters.

### 5.2. Synthetic Data Generation

We can sample profiles in a controlled and realistic manner and produce their corresponding winning sets with existing voting rules (see Section 3). So we generate synthetic data: Given a profile distribution  $\mu$  and a voting rule  $F$ , we randomly pick integers  $n \in [1, n_{\max}]$  and  $m \in [1, m_{\max}]$  and  $\mu$ -sample a profile  $\mathbf{P}$  with  $n$  voters and  $m$  alternatives and compute  $S = F(\mathbf{P})$ . Thus, we generate a dataset  $D = \{(\mathbf{P}_1, S_1), \dots, (\mathbf{P}_k, S_k)\}$ .

### 5.3. Evaluating Axiom Satisfaction

To evaluate the axiom satisfaction of a voting rule (be it realized by a neural network or an existing one), we sample 400 profiles on which the axioms are applicable. We use the same profile distribution  $\mu$  as was used for training the neural network, and we again randomly choose integers  $n \in [1, n_{\max}]$  and  $m \in [1, m_{\max}]$  before  $\mu$ -sampling a profile with  $n$  voters and  $m$  alternatives. To compute whether an axiom is satisfied for a profile, the axioms of anonymity, neutrality, and independence require sampling of permutations. We sample, per profile, 50, 50, and  $4^n$  (with  $n$  the number of voters in the considered profile) permutations, respectively.<sup>5</sup>

<sup>5</sup>Independence requires both a permutation of voters and of alternatives, hence we sample more permuted versions of the given profile.

## 5.4. Hyperparameters

All models use ReLU as activation function. Our MLP has three hidden layers with 128 neurons each. The CNN has two convolution layers with kernel size (5, 1) and (1, 5), respectively (and 32 or 64 channels), followed by three linear layers with 128 neurons. The WEC has the word embedding layer, then the averaging layer, and then three linear layers with 128 neurons. For pre-training the embedding layer with word2vec, we use a corpus size of  $10^5$ , an embedding dimension of 200, and a window size of 7.<sup>6</sup> This results in the following numbers of parameters in the setting  $n_{\max} = 77$  and  $m_{\max} = 7$ : the MLP has 500.487 parameters, the CNN has 1.834.439 parameters, and the WEC has 1.226.143 parameters. Thus, the models are roughly comparable in size.

For training, we use the *AdamW* algorithm (Loshchilov and Hutter, 2019). We use a batch size of 200. Since we have synthetic data, we do not use epochs and hence only specify the number of gradient steps. In experiment 1, 2, and 3, these are 15,000, 5,000, and 15,000, respectively. Similar to Anil and Bao (2021), we use as a learning rate scheduler cosine annealing with warm restarts (Loshchilov and Hutter, 2017). All results are reported for one fixed seed. (In the Appendix, tables 4 and 5 report averaged results across different seeds.) All experiments have been run on a laptop without GPU.

## 6. Results and Analysis

Within our axiomatic deep voting framework, we answer our three research questions: (1) Are preferences-aggregating neural networks correct for the right reasons? No. (2) Can they learn voting-theoretic principles by example? No. (3) Can they synthesize new rules guided by the principles? Yes.

### 6.1. Correct for the Right Reasons?

Recent work in computer science has studied the capabilities of neural networks to learn voting rules (Anil and Bao, 2021; Burka et al., 2022), but without asking whether “the system performs well for the right reasons” (Bender and Koller, 2020, p. 5192). Here we use voting-theoretic axioms to shed light on the learning behavior of neural networks, specifically aiming to distinguish solely accurate versus principled learning.

**Design.** We train each one of the three neural network architectures (MLP, CNN, and WEC) on data from each one of the three basic voting rules (Plurality, Borda, and Copeland) using four different sampling distributions (IC, Urn, Mallows, and Euclidean). We report the results as *relative* accuracy and axiom satisfaction, i.e.,

$$\langle \text{relative evaluation} \rangle = \langle \text{rule evaluation} \rangle - \langle \text{model evaluation} \rangle.$$

For example, if the model has 95% identity accuracy, then, since the rule trivially has 100% accuracy, the relative identity accuracy is  $100\% - 95\% = 5\%$  (i.e., the error). If the model has 35% satisfaction of the independence axiom and the rule only 30%, then the relative independence satisfaction is  $30\% - 35\% = -5\%$ .

**Results.** The relative accuracy and axiom satisfaction when sampling with the IC distribution are given in Figure 4. (Section B in the Appendix shows similar results for the other distributions.) The three architectures do not differ much in accuracy. The best accuracy is achieved for the simple Plurality rule, while the complex Copeland rule decreases accuracy.

Notably, across all voting rules, architectures, and distributions, we see large violations of neutrality despite high accuracy (e.g., 4.6% identity-accuracy error for the WEC architecture when trained on

<sup>6</sup>That is in the setting  $n_{\max} = 77$  and  $m_{\max} = 7$ . When  $n_{\max} = 55$  and  $m_{\max} = 5$ , we reduce this to a corpus size of  $2 \times 10^4$ , an embedding dimension of 100, and a window size of 5.

the Plurality rule but still 19.5% neutrality error). Large violations of anonymity are also observed under the MLP and CNN architectures (the WEC is anonymous by design). This is particularly noteworthy since anonymity and neutrality are always satisfied by the given voting rules. The MLP and CNN models regularly violate anonymity more than neutrality (with the models trained on Plurality demonstrating the smallest such difference).

Regarding the other axioms, all models adhere perfectly to Pareto, in accordance with the voting rules on which they are trained. The MLP and WEC models trained on Plurality seem to satisfy the Condorcet principle more than Plurality does, but the opposite holds for the CNN model. Along a similar line, the MLP model trained on Borda seems to satisfy the Condorcet principle more than Borda does, but this is not the case for the CNN and WEC models. Since Copeland always satisfies the Condorcet principle, the corresponding principled error of all models is positive—yet, it is rather small. The MLP and WEC models trained on Plurality and Borda, as well as the CNN model trained on Plurality, satisfy independence to a similar degree as the rules do on which they are trained. All models trained on Copeland satisfy independence more than Copeland does, and the same holds for the CNN model trained on Borda.

**Discussion.** Regarding the learnability of different voting rules, the simplicity of Plurality is probably the reason behind the high accuracy with which all models learn it. However, this simplicity also renders Plurality problematic in other contexts (Laslier, 2011).

The models take a stance on the well-documented tension between anonymity and neutrality.<sup>7</sup> They tend to favor outcomes that align more closely with the former than with the latter. This inclination exposes an inherent bias within neural networks when navigating fundamental democratic axioms.

As the architectures are not invariant to permutations of the input data, the severe violations neutrality (and of anonymity for MLPs and CNNs) are not *a priori* surprising. What is surprising is that this violation persists even for high accuracy with respect to rules that are perfectly neutral and anonymous.

Overall, our experiment on accurate versus principled learning within voting contexts highlights precisely the importance of the *reasons* behind automated decision-making. Outcomes that are mimicking well-defined voting rules are arguably still unsafe to rely on, since they do not come with a guarantee of respecting the principles on which those rules are built.

### 6.2. Learning Principles by Example?

Can we teach neural networks voting-theoretic principles, beyond merely presenting data from various voting rules? A natural approach for integrating expert knowledge in neural networks is data augmentation. In the voting-context, this was proposed by Xia (2013) but has not been tested in practice, to the best of our knowledge. We focus on the neutrality axiom, since it was violated most, and we also test the effects of data augmentation on the model’s accuracy.

**Design.** We train each one of the three neural network architectures (MLP, CNN, and WEC) on data from each one of the three basic voting rules (Plurality, Borda, and Copeland); but we vary the ratio  $p$  of sampled data and augmented data, while keeping the total number of data points fixed. Thereby, any improvement comes from the quality of the augmented data points and not from merely a higher quantity of data points.

Specifically, given a percentage  $1 \leq p \leq 100$ , we first sample  $p$  (times the total data size) many profiles and compute the corresponding winning sets according to the considered rule; call these the *sampled* data points. Then we generate the remaining  $100 - p$  many data points as follows: we randomly pick one of the sampled data

<sup>7</sup>No voting rule that always elects a single winner can simultaneously uphold both anonymity and neutrality.

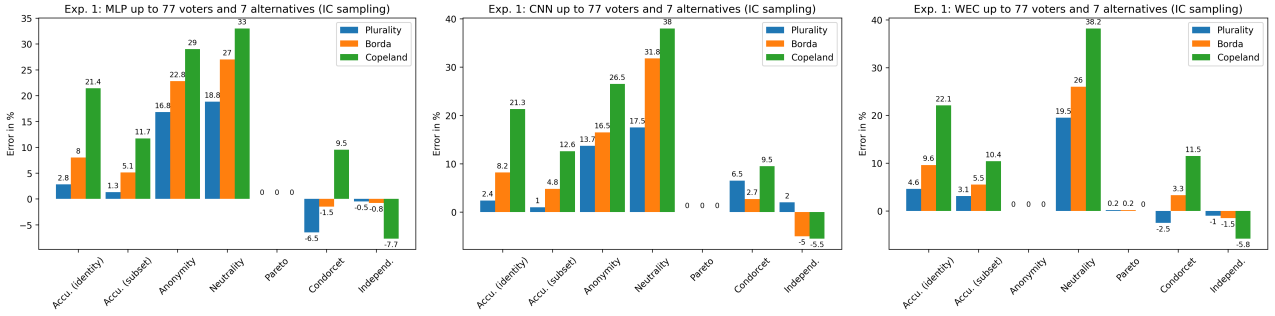


Figure 4: Training the three architectures (MLP, CNN, and WEC) on data from Plurality, Borda, and Copeland (the three bars in each plot) with IC samples and comparing the errors in both accuracy and axiom satisfaction.

points  $(\mathbf{P}, S)$  and a permutation  $\sigma$  of the alternatives, and then add the data point  $(\sigma(\mathbf{P}), \sigma(S))$ ; call these the *augmented* data points. Thus, the augmented data points are “neutrality variations” of the sampled data points. For different choices of  $p$ , we then test the models’ neutrality satisfaction and accuracy.

**Results.** Results for IC sampling are exhibited in Figure 5 (and for the Mallows model in Figure 8 in the Appendix). We find that data augmentation does not improve adherence to neutrality: the ratio  $p$  between sampled and augmented data does not seem to correlate with neutrality satisfaction. For  $p < 10\%$ , i.e., with almost only augmented data, both accuracy and neutrality satisfaction are unsatisfactory, so data augmentation only becomes relevant for  $p \geq 10\%$ . Here accuracy is stable: it does not vary by more than 5%. In some cases, neutrality is equally stable: for the CNN on all rules and the MLP on Borda (certainly for  $p \geq 25\%$ , with slightly worse neutrality satisfaction for smaller  $p$ ). In the remaining non-stable cases, the best neutrality satisfaction is achieved for  $p = 100\%$ , i.e., without augmented data—with only negligible exceptions.<sup>8</sup> Thus, neither in the stable nor the unstable cases can we see reliable comparative improvements in neutrality satisfaction with more neutrality-augmented data.

**Discussion.** Learning voting-theoretic principles by examples—augmented to the training data—does not seem to work for neural networks: Comparatively more neutrality-augmented data does not lead to higher neutrality satisfaction. However, an advantage of data augmentation is a drastic increase in data efficiency when we only aim for accuracy. Sampling only 10% of the total data set (and using neutrality augmented data for the remaining 90%) does not substantially decrease the MLP’s or WEC’s accuracy in comparison to sampling the whole data set. This is crucial if we use real and not sampled election data, where having access to a vast amount of data points is practically impossible. Even when more data is needed to increase the accuracy of network, we could build an appropriate data set based on a limited amount of real data points and then augment it via the neutrality axiom.

### 6.3. Rule Synthesis Guided by Principles?

We saw that neural networks, when trained on data from established voting rules, struggle to vote with principles. This raises the question: can we directly train neural networks to form principled collective decisions, without relying on any pre-existing voting rules? This will be limited by Arrow’s Impossibility Theorem (Arrow, 1951): a voting

rule cannot simultaneously satisfy anonymity, Pareto, and independence. Neural network-based approaches also face this impossibility. However, how close can we get to full axiom satisfaction? We design an optimization task, using custom loss functions, to guide neural networks in learning novel and principled voting rules.

**Design.** We train each one of the three neural network architectures (MLP, CNN, and WEC) on the loss functions defined in Section 4.4 which represent the axioms anonymity, neutrality, Condorcet, Pareto, and independence. Since neural networks could attempt to vacuously satisfy the axioms by proposing no winner, we also consider the “No-winner” loss function, which demands the winning sets to be nonempty. Moreover, by Arrow’s Theorem, the axioms cannot be jointly satisfied and will, hence, negatively influence each other. So optimizing for all axioms is not necessarily the best. Instead, we pick, for each architecture, a set  $\mathcal{O}$  of objectives that we optimize for. For WEC, we choose: no winner, Condorcet, and Pareto. For MLP and CNN, we add: anonymity and independence. Then the optimization problem is:

$$\operatorname{argmin}_w \sum_{\mathbf{O} \in \mathcal{O}} \mathbb{E}_{\mathbf{P} \sim \mathcal{D}} [L_{\mathbf{O}}(f_w, \mathbf{P})],$$

where the loss functions  $L_{\mathbf{O}}$  are described in Section 4.4 and  $\mathcal{D}$  is the chosen distribution of profiles  $\mathbf{P}$  (IC or Mallows). Note that, unlike the previous experiments, this is an unsupervised learning task.

In order to have an architecture that also is neutral by design (not just anonymous by design like the WEC), we design a further decoding function in addition to the one used so far (Section 4.3). This *neutrality-averaged* decoding works as follows (cf. Burka et al., 2022). Given an input profile, we first generate all alternative-permuted versions of the profile, then compute the logits-predictions of the model on each of those permuted profiles (in one batch), next de-permute the predictions again and average all of them, and finally we turn those average logits into a winning set with the decoding function used so far.

Thus, WEC with neutrality-averaged decoding is anonymous and neutral by design. For the other architectures, we also test a decoding method that is *neutrality-and-anonymity-averaged*. For that, given an input profile, we first randomly generate 12 alternative-permuted versions of it, and, for each of those, we also randomly generate 10 voter-permuted versions and, as before, compute the averaged logits and from those the winning set. The numbers are explained as follows: Neutrality-averaging requires, with at most 5 alternatives, considering at most  $5! = 120$  permutations; hence neutrality-and-anonymity-averaging also considers  $12 \times 10 = 120$  permutations (checking all  $5! \approx 10^{73}$  voter permutations would be infeasible).

**Results.** Table 1 shows the axiom satisfaction of different neural networks (bottom) and, for comparison, of several known rules from voting theory (top), all using IC sampling (for Mallows see Table 6 in the Appendix). The best neural-network based rule in terms of axiom

<sup>8</sup>The only two exceptions are the CNN on Plurality (where neutrality is most satisfied at  $p = 75\%$  but to a very similar degree as for  $p = 100\%$ ) and the CNN on Copeland (where neutrality is minimized at  $p = 25\%$ ). Moreover, CNN on Borda and MLP on Copeland have a local—albeit not global—minimum at  $p = 25\%$ . Thus, while there might be some special cases where neutrality is improved in the highly augmented scenario, this is not enough to consider data augmentation as a successful strategy to improve neutrality satisfaction (which is what we are concerned with here).

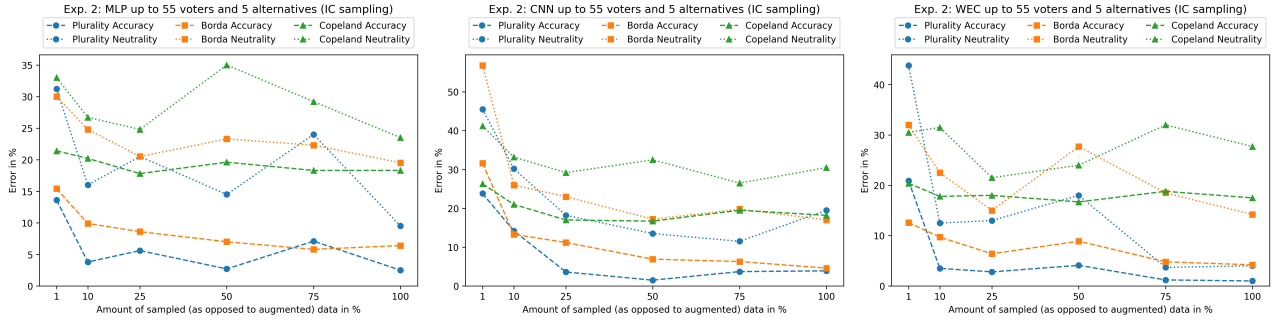


Figure 5: For each architecture, the accuracy and neutrality error across different ratios of augmented data. For example, 10% in the x axis means that the dataset consists of 10% sampled data and 90% augmented data.

satisfaction is the neutrality-averaged WEC, with close contestants the neutrality-averaged CNN and MLP. The neutrality-averaged WEC clearly outperforms the classic Plurality, Borda, and Copeland rules in every single axiom. Even when we consider more modern rules in voting theory, the neutrality-averaged WEC is competitive: the existing rule with highest axiom satisfaction is Stable Voting and its edge is marginal, with its average axiom satisfaction being less than 1% higher than that of the neutrality-averaged WEC.<sup>9</sup> (In the Mallows case, the neutrality-averaged WEC even just beats all other voting rules, see table 6 in the Appendix.<sup>10</sup>)

In addition to examining axiom satisfaction, we should also consider how often the examined rules produce the same outcomes: because similar axiom satisfaction does not imply similarity of outcomes.<sup>11</sup> Table 2 describes similarity in outcome between the five rules which excelled in axiom satisfaction, using IC sampling (for Mallows, see Table 7 in the Appendix). In particular, we see that the rule discovered by the neutrality-averaged WEC model is substantially different from the existing voting rules: it proposes different outcomes than each one of them at least 9.3% of the time. In comparison, Stable Voting that was found best in Table 1 disagrees with Borda and Copeland 8.9% of the time and with Weak Nanson and Blacks only 6.6% of the time. Thus, the discovered rule not only is competitive in axiom satisfaction, it also is novel, i.e., substantially different from existing voting rules.

To illustrate the difference between the discovered and the existing rules, Figure 6 shows an example of a profile where the winning set provided by the neutrality-averaged WEC model is different to all the winning sets provided by the considered existing voting rules. The choice of the WEC also has intuitive plausibility: it chooses alternative a which, among the eight voters, is three times the most preferred option and two times the second-most preferred option.

**Discussion.** The reason why the WEC outperforms the other two architectures is that, because it is anonymous by design, it is enough to use the neutrality-averaged decoding to get a model that is anonymous and neutral. Since the MLP and CNN are not anonymous, they need neutrality-and-anonymity-averaged decoding to become anonymous and neutral by design. This, however, needs infeasibly many permutations, so it can only be approximated via sampling permutations. Here, however, the tension between the axioms of anonymity and neutrality resurfaces: sampled neutrality-and-anonymity-averaging can result in negative interference with the other axioms yielding worse performance than just neutrality-averaging (e.g., the CNN rules in Table 1). Mere neutrality-averaging

	1	2	3	4	5	6	7	8
a	e	d	a	e	b	e	a	a
b	b	b	c	b	a	a	a	b
e	d	c	b	c	e	c	d	d
d	a	e	e	a	c	d	e	e
c	c	a	d	d	d	b	c	c

- {a} neutrality-averaged WEC
- {b} Blacks, Stable Voting, Borda, Weak Nanson, Copeland
- {a, e} Plurality, PluralityWRunoff PUT
- {e} Instant Runoff TB, Anti-Plurality
- {a, b} Llull, Uncovered Set, Banks, Coombs, Baldwin, and Kemeny-Young
- {a, b, e} Top Cycle

Figure 6: Profile where the ‘WEC n’ model disagrees with existing voting rules. The winning sets for each rule are mentioned below the table.

also influences the satisfaction of the other axioms, but in this case not in a negative way.<sup>12</sup>

Moreover, for the WEC just three optimization objectives were enough to obtain the above competitive results. Since the MLP and CNN are not anonymous by design, they needed to optimize for anonymity as well. The MLP and CNN also needed to optimize for independence, while the WEC interestingly had enough *implicit* inductive bias toward satisfying independence—again highlighting non-trivial interference of the axioms and the network architecture.

The neural networks beat the classic voting rules in terms of axiom satisfaction while being comparable to the best voting rules known today. This may be taken to suggest that existing rules may already be close to optimal axiom satisfaction. In other words, they are in the (approximate) *Pareto front* of axiom satisfaction. At the same time, even if the novel rules derived from axiom optimization inherit the opacity of neural networks, they assure high adherence to key normative principles in collective decisions. Since these newly discovered rules were substantially different from existing rules, they extend the boundaries of what is so far explored in voting theory.

## 7. Discussion

With our axiomatic deep voting framework, we investigated the space of all voting rules by fruitfully combing voting theory and machine learning. The neural network explores the space and the voting-theoretic axioms evaluate the network, thus guiding the exploration. The universal approximation theorems (Hornik et al., 1989; Cybenko, 1989) ensure that the neural networks are dense in the space of all voting rules, so all areas of that space can be explored with axiomatic deep voting. Arrow’s Impossibility Theorem (Arrow, 1951) establishes insurmountable divisions of that space: e.g., the

<sup>9</sup>Table 3 in the Appendix suggests that more gradient steps do not further improve the results.

<sup>10</sup>Tables 4 and 5 in the Appendix suggest the statistical robustness of these results by reporting the average results across 5 runs of this experiment with different seeds.

<sup>11</sup>For example, the Blacks and Weak Nanson rules are close in average axiom satisfaction (less than 1% difference), but Table 2 shows that more than 8% of the time they propose a different set of winners.

<sup>12</sup>We did not use neutrality-averaging in the previous experiments because it would not directly correspond to the binary cross entropy loss and the interference with the other axioms blurs the axiomatic evaluation of the neural network.



	Anon.	Neut.	Condorcet	Pareto	Indep.	Average
Plurality	100	100	80.2	100	28.5	81.8
Borda	100	100	95.5	100	37.2	86.5
Anti-Plurality	100	100	74.2	100	24.8	79.8
Copeland	100	100	100	100	28.0	85.6
Llull	100	100	100	100	26.8	85.4
Uncovered Set	100	100	100	100	27.8	85.5
Top Cycle	100	100	100	100	29.0	85.8
Banks	100	100	100	100	27.8	85.5
Stable Voting	100	100	100	100	43.0	88.6
Blacks	100	100	100	100	35.2	87.1
Instant Runoff TB	100	100	94.8	100	28.2	84.6
PluralityWRunoff PUT	100	100	95.0	100	25.5	84.1
Coombs	100	100	96.2	100	34.5	86.2
Baldwin	100	100	100	100	39.2	87.9
Weak Nanson	100	100	100	100	40.0	88.0
Kemeny-Young	100	100	100	100	39.2	87.9
MLP p (NW, A, C, P, I)	77.8	75.8	92.5	100	39.5	77.1
MLP n (NW, A, C, P, I)	89.2	100	95.0	100	42.2	85.3
MLP na (NW, A, C, P, I)	89.8	86.8	95.5	100	36.5	81.7
CNN p (NW, A, C, P, I)	85.2	67.2	92.0	100	39.5	76.8
CNN n (NW, A, C, P, I)	92.2	100	94.5	100	40.0	85.4
CNN na (NW, A, C, P, I)	86.0	86.5	94.8	100	34.0	80.2
WEC p (NW, C, P)	100	72.5	94.2	100	41.8	81.7
WEC n (NW, C, P)	100	100	96.8	100	41.2	87.6

Table 1: Axiom satisfaction of different rules (top part of the table) and models (bottom part of the table), for IC sampling. Rounded to one decimal. The names of the models are explained as follows: The letters after the architecture type indicate how the voting rule is computed from the model: p–plain (i.e., no averaging), n–neutrality-averaged, na–neutrality-and-anonymity-averaged. The letters in the brackets indicate which axioms the model optimized for during training: NW–No winner, A–Anonymity, C–Condorcet, P–Pareto, I–Independence. All models have been trained for 15k gradient steps with batch size 200 and the same seed. The WEC by far was the fastest.

	WEC n	Blacks	Stable Voting	Borda	Weak Nanson	Copeland
WEC n	100	90.7	90.2	89.6	88.1	87.4
Blacks		100	95.65	95.45	91.82	90.56
Stable Voting			100	91.1	93.38	91.99
Borda				100	87.27	86.01
Weak Nanson					100	92.26
Copeland						100

Table 2: Similarities between the rules. Computed on 10,000 IC-sampled profiles. For example, the entry 90.2 in row ‘WEC n’ and column ‘Stable Voting’ means that in 90.2% of the sampled profiles Stable Voting outputs the same winning set as the neutrality-averaged WEC.

area of rules satisfying anonymity and Pareto does not intersect the area of rules satisfying independence.

The importance of our results for AI is twofold. First, the axiomatic evaluation offers another cautionary tale that accuracy is not everything: Neural networks can have high accuracy (descriptively good) without following the right reasons (normatively bad). Second, this changes, however, when we move from the supervised setting of learning rules from examples to the unsupervised setting of directly optimizing axiom satisfaction. We were able to do this by translating the voting-theoretic axioms into corresponding loss functions. Having a way to optimize the axioms is important because the axioms can be seen as *mathematical formalizations* of important normative notions in modern machine learning. For example:

- *Bias*: anonymity says that the neural network is not biased towards particular individuals.
- *Fairness*: neutrality demands that the neural network treats all alternatives equally.
- *Value-alignment*: the Pareto principle requires that if all individuals value one alternative more than another, then the neural

network aligns with this; and similarly for the Condorcet principle.

- *Interpretability*: independence provides a sense of ‘compositionality’ when interpreting the network—to understand its choice for two given alternatives, we can ignore all other alternatives.

Hence, our axiomatic optimization provides a way of improving the neural network—in a mathematically precise sense—regarding bias, fairness, value-alignment, and interpretability.

Moreover, qua interdisciplinary project, our results are also relevant for voting theory. Axiomatic deep voting offers a new tool for the field’s central goal of exploring the space of all voting rules. While existing voting rules are crafted by human insight, we could find—in a completely automated process—novel voting rules that are comparable in terms of axiom satisfaction to the best rules known today. This provides a promising starting point for an analytic exploration of new axiom-optimal voting rules and the influence the axioms exert on each other.

**Limitations.** We tested a wide range of standard neural network architectures. However, future work could also investigate further architectures like Set Transformers, Graph Isomorphism Networks, or Deep Sets (which were used by Anil and Bao (2021)) and, more generally, the transformer architecture (as a refinement our word embedding architecture). We also covered the most important voting-theoretic axioms, but yet more can be considered, e.g., monotonicity and transitivity (the latter then requires architectures that are not transitive by design). Finally, the large number of permutations causes a high statistical variance in testing the satisfaction of the independence axiom.

**Future work.** First, more options in generating the dataset can be explored. For example, we can consider the *extrapolation* task in which the model has to find a general rule after only observing the rule on a small part of the input space, namely the profiles where some given voting rules agree or satisfy a given axiom. Or we can consider the *interpolation* task in which the model sees data of different rules and has to find a compromise between their outputs.

Second, we can implement further social choice theory frameworks. Since we already output logits corresponding to the alternatives, we could, instead of winning sets, also consider preference rankings or welfare functions. It would also be interesting to consider judgment aggregation, which includes reasoning about logical implications between the alternatives.

Third, it seems promising to bridge notions of explainability in voting theory (Cailloux and Endriss, 2016; Nardi et al., 2022; Boixel et al., 2022) and notions of explainability in AI (Adadi and Berrada, 2018). In particular, is it possible to extract out a symbolic representation (e.g., in logic programming) of the rule that the model learned?

Fourth, studying the voting-theoretic concept of manipulability via neural networks (Holliday et al., 2024) can be further connected to machine learning notions like *adversarial attacks* (Goodfellow et al., 2015) or *performativity* (Perdomo et al., 2020).

Fifth, from the point of view of *geometric deep learning* (Bronstein et al., 2021), axioms represent *symmetries* that the neural networks should learn. For example, anonymity says that the neural network should be invariant under the group action of the voter-permutation group on profiles; and neutrality says that the neural network should be equivariant under the group action of the alternative-permutation group on profiles and winning sets, respectively. It seems worth exploring this connection to geometric deep learning.

## 8. Conclusion

We introduced the axiomatic deep voting framework to study how neural networks aggregate preferences. We found that neural networks *do not* learn to vote with principles, despite achieving high accuracy, when trained on data from existing voting rules—even when augmented with axiom-specific data. However, they *do* learn to vote with principles when they directly optimize for axiom satisfaction, which we achieved by translating axioms into custom loss functions. The axiomatic deep voting framework promises fruitful further investigation both in voting theory (new ways of exploring the space of voting rules) and AI (a mathematically precise testing ground for normative notions like bias and value-alignment).

**Acknowledgments** For very helpful comments and discussions, we would like to thank Ben Armstrong, Balder ten Cate, Timo Freiesleben, Ronald de Haan, Alina Leidinger, and Christian List.

## References

- Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160.
- Anil, C. and Bao, X. (2021). Learning to elect. *Advances in Neural Information Processing Systems*, 34:8006–8017.
- Armstrong, B. and Larson, K. (2019). Machine learning to strengthen democracy. In *NeurIPS Joint Workshop on AI for Social Good*.
- Arrow, K. J. (1951). *Social Choice and Individual Values*. John Wiley & Sons.
- Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. (2022). Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv*. arXiv:2204.05862.
- Bender, E. M. and Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198.
- Boehmer, N., Brederbeck, R., Faliszewski, P., Niedermeier, R., and Szufa, S. (2021). Putting a compass on the map of elections. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*.
- Boehmer, N., Faliszewski, P., Janeczko, Ł., Kaczmarczyk, A., Lisowski, G., Pierczyński, G., Rey, S., Stolicki, D., Szufa, S., and Waś, T. (2024). Guide to numerical experiments on elections in computational social choice. *arXiv*. arXiv:2402.11765.
- Boehmer, N., Faliszewski, P., and Kraicz, S. (2023). Properties of the mallows model depending on the number of alternatives: A warning for an experimentalist. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.
- Boixel, A., Endriss, U., and de Haan, R. (2022). A calculus for computing structured justifications for election outcomes. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*.
- Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. D., editors (2016). *Handbook of Computational Social Choice*. Cambridge University Press.
- Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. (2021). Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv*. arXiv:2104.13478.
- Burka, D., Puppe, C., Szepesváry, L., and Tasnádi, A. (2022). Voting: A machine learning approach. *European Journal of Operational Research*, 299(3):1003–1017.
- Cailloux, O. and Endriss, U. (2016). Arguing about voting rules. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
- Caragiannis, I. and Micha, E. (2017). Learning a ground truth ranking using noisy approval votes. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.
- Conitzer, V., Freedman, R., Heitzig, J., Holliday, W., Jacobs, B., Lambert, N., Mossé, M., Pacuit, E., Russell, S., Schoelkopf, H., Tewolde, E., and Zwicker, W. (2024). Position: Social choice should guide ai alignment in dealing with diverse human feedback. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.
- Cybenko, G. (1989). Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems*, 2(4):303–314.
- Dougherty, K. L. and Heckelman, J. C. (2020). The probability of violating arrow’s conditions. *European Journal of Political Economy*, 65:101936.
- Eggenberger, F. and Pólya, G. (1923). Über die statistik verketeter vorgänge. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 3(4):279–289.
- Favardin, P. and Lepelley, D. (2006). Some further results on the manipulability of social choice rules. *Social Choice and Welfare*, pages 485–509.
- Favardin, P., Lepelley, D., and Serais, J. (2002). Borda rule, Copeland method and strategic manipulation. *Review of Economic Design*, 7:213–228.
- Fishburn, P. C. and Gehrlein, W. V. (1982). Majority efficiencies for simple voting procedures: Summary and interpretation. *Theory and Decision*, 14(2):141–153.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. The MIT Press, Cambridge, Massachusetts.

- Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*.
- Gudiño Rosero, J., Grandi, U., and Hidalgo, C. A. (2024). Large language models (LLMs) as agents for augmented democracy. *arXiv*. arXiv:2405.03452.
- Hatzivelkos, A. (2018). Borda and Plurality comparison with regard to compromise as a sorites paradox. *Interdisciplinary Description of Complex Systems: INDECS*, 16(3B):465–484.
- Holliday, W. and Pacuit, E. (2023a). Split cycle: A new Condorcet-consistent voting method independent of clones and immune to spoilers. *Public Choice*, 197(1):1–62.
- Holliday, W. and Pacuit, E. (2023b). Stable voting. *Constitutional Political Economy*, 34(3):421–433.
- Holliday, W. H., Kristoffersen, A., and Pacuit, E. (2024). Learning to manipulate under limited information. *arXiv*. arXiv:2401.16412.
- Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feed-forward networks are universal approximators. *Neural Networks*, 2(5):359–366.
- Kujawska, H., Slavkovik, M., and Rückmann, J.-J. (2020). Predicting the winners of borda, kemeny, and dodgson elections with supervised machine learning. In *Multi-Agent Systems and Agreement Technologies Workshop. At the 17th European Conference on Multi-Agent Systems (EUMAS)*, pages 440–458.
- Laslier, J.-F. (2011). And the loser is... plurality voting. *Electoral Systems*, pages 327–351.
- Lee, D., Goel, A., Aitamurto, T., and Landemore, H. (2014). Crowdsourcing for participatory democracies: Efficient elicitation of social choice functions. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing*.
- List, C. (2011). The logical space of democracy. *Philosophy & public affairs*, 39(3):262–297.
- List, C. (2022). Social Choice Theory. In Zalta, E. N. and Nodelman, U., editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2022 edition.
- Loshchilov, I. and Hutter, F. (2017). SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*.
- Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.
- Mallows, C. L. (1957). Non-null ranking models. *Biometrika*, 44(1/2):114–130.
- Merrill, S. (1984). A comparison of efficiency of multicandidate electoral systems. *American Journal of Political Science*, pages 23–48.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Mishra, A. (2023). AI alignment and social choice: Fundamental limitations and policy implications. *arXiv*. arXiv:2310.16048.
- Mohsin, F., Liu, A., Chen, P.-Y., Rossi, F., and Xia, L. (2022). Learning to design fair and private voting rules. *Journal of Artificial Intelligence Research*, 75:1139–1176.
- Nardi, O., Boixel, A., and Endriss, U. (2022). A graph-based algorithm for the automated justification of collective decisions. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
- Nitzan, S. (1985). The vulnerability of point-voting schemes to preference variation and strategic manipulation. *Public choice*, 47:349–370.
- Noothigattu, R., Gaikwad, S., Awad, E., Dsouza, S., Rahwan, I., Ravikumar, P., and Procaccia, A. (2018). A voting-based system for ethical decision making. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*.
- Nurmi, H. (1988). Discrepancies in the outcomes resulting from different voting schemes. *Theory and Decision*, 25:193–208.
- Perdomo, J., Zrnic, T., Mendler-Dünner, C., and Hardt, M. (2020). Performative prediction. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7599–7609. PMLR.
- Powers, R. C. (2007). The number of times an anonymous rule violates independence in the  $3 \times 3$  case. *Social Choice and Welfare*, 28(3):363–373.
- Procaccia, A. D., Zohar, A., Peleg, Y., and Rosenschein, J. S. (2009). The learnability of voting rules. *Artificial Intelligence*, 173(12-13):1133–1149.
- Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. (2024). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.
- Terzopoulou, Z. (2023). Voting with limited energy: A study of Plurality and Borda. *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
- Thomson, W. (2001). On the axiomatic method and its recent applications to game theory and resource allocation. *Social Choice and Welfare*, 18(2):327–386.
- Xia, L. (2013). Designing social choice mechanisms using machine learning. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
- Xu, J., Zhang, Z., Friedman, T., Liang, Y., and Broeck, G. (2018). A semantic loss function for deep learning with symbolic knowledge. In *International conference on machine learning (ICML)*, pages 5502–5511.
- Yang, J., Dailisan, D., Korecki, M., Hausladen, C., and Helbing, D. (2024). Llm voting: Human choices and AI collective decision-making. *arXiv*. arXiv:2402.01766.
- Zwicker, W. S. (2016). Introduction to the theory of voting. In Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. D., editors, *Handbook of Computational Social Choice*. Cambridge University Press.

## A. List of all loss functions

We continue from section 4.4 and define the other loss functions that we use and discuss their differentiability. Recall that KL refers to Kullback–Leibler divergence.<sup>13</sup>

**Anonymity** For convenience, we repeat the definition for the anonymity loss. Given the network  $f_w$  and profile  $\mathbf{P}$ , uniformly sample  $N$ -many permutations  $\pi_1, \dots, \pi_N$  of the set of voters of  $\mathbf{P}$  and define

$$L_A(f_w, \mathbf{P}) := \frac{1}{N} \sum_{\tau=1}^N \text{KL}(f_w(e(\mathbf{P})), f_w(e(\pi_\tau(\mathbf{P}))).$$

**Condorcet** If  $\mathbf{P}$  has no Condorcet winner,  $L_C(f_w, \mathbf{P}) := 0$ , and otherwise, if that Condorcet winner is alternative  $a$ , define (recall  $\bar{a}$  is the one-hot vector for alternative  $a$ )

$$L_C(f_w, \mathbf{P}) := \text{KL}(f_w(e(\mathbf{P})), \bar{a}).$$

**Pareto** We define (recall that  $\sigma$  is the sigmoid function and  $n_{a>b}^{\mathbf{P}} = n$  means that all voters in  $\mathbf{P}$  rank  $a$  above  $b$ )

$$L_P(f_w, \mathbf{P}) := \sum_{a,b \text{ with } n_{a>b}^{\mathbf{P}} = n} \sigma(f_w(e(\mathbf{P}))_b).$$

**Independence** Define  $L_I(f_w, \mathbf{P}) := 0$  if  $\mathbf{P}$  does not have at least two alternatives. Otherwise, randomly sample  $N$ -many pairs  $(a_\tau, b_\tau)$  of distinct alternatives in  $\mathbf{P}$  and randomly sample, for each ranking  $P_k$  of  $\mathbf{P} = (P_1, \dots, P_n)$ , a shuffling  $P'_k$  of  $P_k$  in which, however, the order of  $a_\tau$  and  $b_\tau$  is the same as in  $P_k$ , and set  $\mathbf{P}_\tau := (P'_1, \dots, P'_n)$ . Write  $\hat{y} := f_w(e(\mathbf{P}))$  and  $\hat{y}^\tau := f_w(e(\mathbf{P}_\tau))$ , and define

$$L_I(f_w, \mathbf{P}) := \sum_{\tau=1}^N \text{KL}((\hat{y}_{a_\tau} \hat{y}_{b_\tau}), (\hat{y}_{a_\tau}^\tau \hat{y}_{b_\tau}^\tau)).$$

**No winner** Writing  $\hat{y} = f_w(e(\mathbf{P}))$  we want that at least one of the numbers in  $\mathbf{p} := (\sigma(\hat{y}_1), \dots, \sigma(\hat{y}_m))$  is above 0.5, i.e., the maximum norm  $\|\mathbf{p}\|_\infty$  should be above 0.5. Hence the more it is below that, the worse the loss:

$$L_{NW}(f_w, \mathbf{P}) := \max(0.5 - \|\mathbf{p}\|_\infty, 0).$$

For almost-everywhere differentiability use the distributivity of the differential operator over sums, the chain rule, and the almost-everywhere differentiability of the involved functions (KL,  $\sigma$ , max,  $\|\cdot\|_\infty$ ).

## B. Experiment 1

We add figure 7.

## C. Experiment 2

We add figure 8.

## D. Experiment 3

We add tables 3, 4, 5, 6, and 7.

<sup>13</sup>Though, in principle, other distance/similarity functions can be considered.

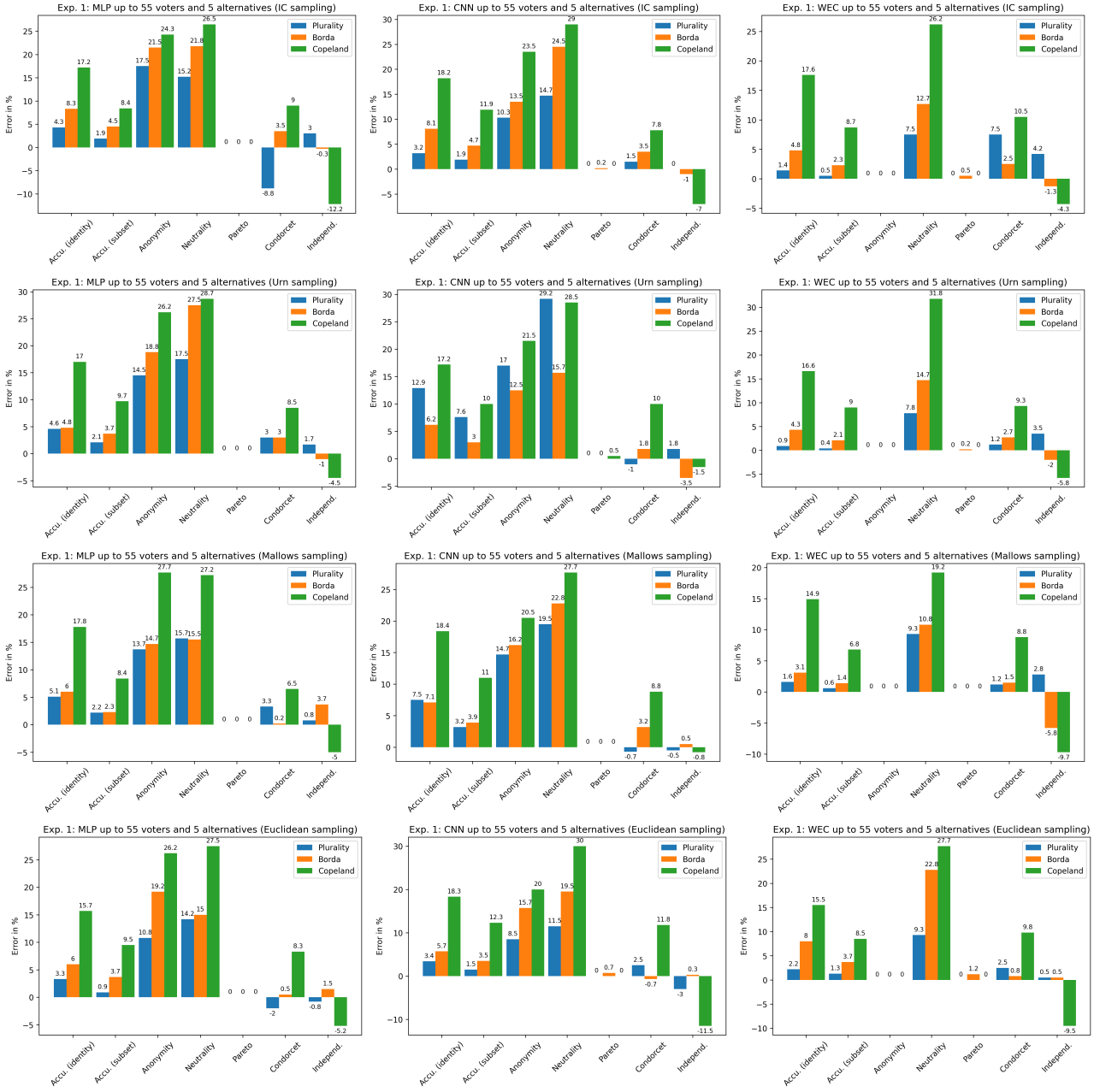


Figure 7: Training the architectures (MLP, CNN, WEC; the three columns) on data from different voting rules (Plurality, Borda, Copeland; the three bars in each plot) using different sampling methods (IC, URN, MALLOWS, Euclidean; the four rows) and comparing the errors in both accuracy and axiom satisfaction.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Average
WEC n (NW, C, P, round 0)	100	100	97.5	100	46	88.7
WEC n (NW, C, P, round 1)	100	100	100	100	38.5	87.7
WEC n (NW, C, P, round 2)	100	100	100	100	34.8	87
WEC n (NW, C, P, I, round 3)	100	100	100	100	31.8	86.3

Table 3: The result of keeping on training a WEC model: each round adds 20k gradient steps to the previous one. Round 1 is with a learning rate of  $10^{-3}$ , round 2 with  $10^{-4}$ , round 3 with  $5 * 10^{-5}$ , and round 4 the same but with added optimization of independence. IC sampling.

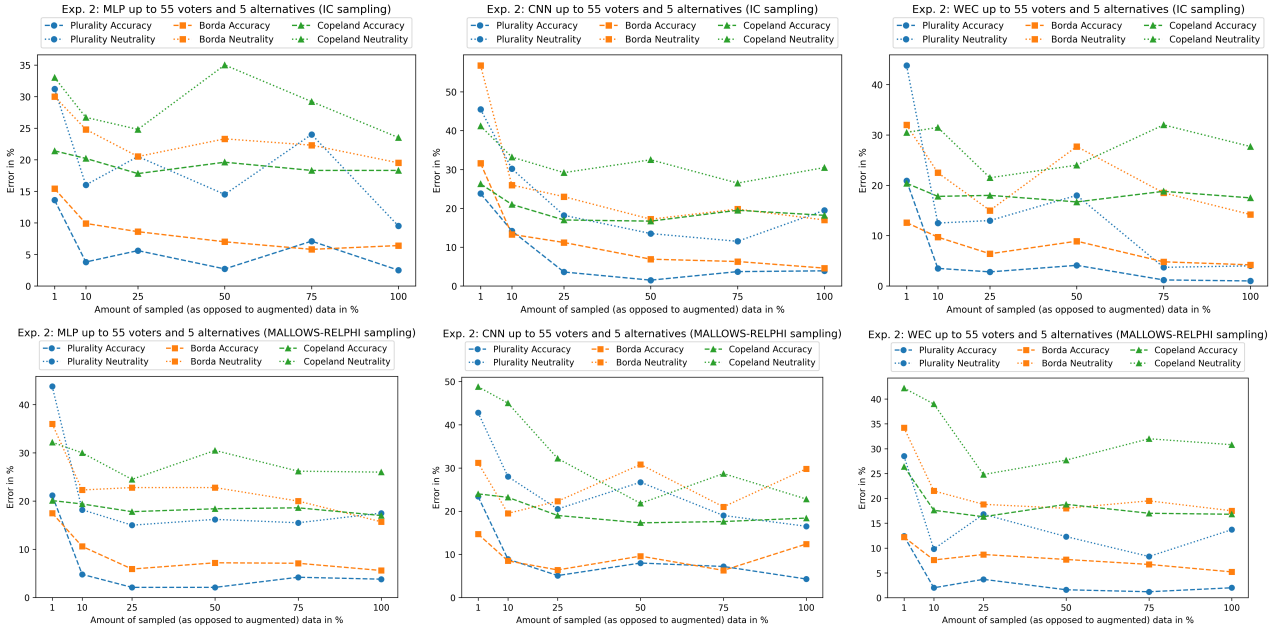


Figure 8: The top row is experiment 2 with IC sampling (repeated for convenience from the main text) and the bottom row is with Mallows sampling. For each architecture (columns from left to right: MLP, CNN, WEC), the accuracy and neutrality error are plotted across different ratios of augmented data. For example, 10% in the x axis means that the dataset consists of 10% sampled data and 90% augmented data.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Average
Blacks	100	100	100	100	36.06	87.24
Stable Voting	100	100	100	100	39.74	87.96
Borda	100	100	94.12	100	35.32	85.9
Weak Nanson	100	100	100	100	39.62	87.92
Copeland	100	100	100	100	27.64	85.54
WEC n (NW, C, P)	100	100	95.4	100	43.88	87.88

Table 4: The average result over 5 runs with different seeds of experiment 3 for the 'WEC n' model and its closest rules. IC sampling.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Average
Blacks	100	100	100	100	35.94	87.18
Stable Voting	100	100	100	100	40.8	88.14
Borda	100	100	94.36	100	35.36	85.94
Weak Nanson	100	100	100	100	40.26	88.06
Copeland	100	100	100	100	27.96	85.6
WEC n (NW, C, P)	100	100	94.66	100	41.94	87.34

Table 5: The average result over 5 runs with different seeds of experiment 3 for the 'WEC n' model and its closest rules. Mallows sampling.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Average
Plurality	100	100	83.0	100	30.2	82.7
Borda	100	100	92.8	100	32.8	85.1
Anti-Plurality	100	100	76.5	100	26.2	80.5
Copeland	100	100	100	100	27.8	85.5
Liull	100	100	100	100	26.2	85.2
Uncovered Set	100	100	100	100	29.5	85.9
Top Cycle	100	100	100	100	25.2	85.0
Banks	100	100	100	100	25.2	85.0
Stable Voting	100	100	100	100	39.0	87.8
Blacks	100	100	100	100	33.8	86.8
Instant Runoff TB	100	100	96.8	100	29.0	85.2
PluralityWRunoff PUT	100	100	94.0	100	27.0	84.2
Coombs	100	100	95.5	100	30.2	85.2
Baldwin	100	100	100	100	39.2	87.9
Weak Nanson	100	100	100	100	33.8	86.8
Kemeny-Young	100	100	100	100	38.2	87.7
MLP p (NW, A, C, P, I)	78.8	76.0	94.0	100	38.8	77.5
MLP n (NW, A, C, P, I)	90.8	100	94.2	100	36.2	84.2
MLP na (NW, A, C, P, I)	92.5	89.5	92.5	100	33.5	81.6
CNN p (NW, A, C, P, I)	80.5	68.8	94.5	100	38.8	76.5
CNN n (NW, A, C, P, I)	91.5	100	95.0	100	42.2	85.8
CNN na (NW, A, C, P, I)	88.0	83.8	94.0	100	36.5	80.5
WEC p (NW, C, P)	100	65.5	91.8	100	37.8	79
WEC n (NW, C, P)	100	100	97.0	100	44.0	88.2

Table 6: Axiom satisfaction of different rules (top part of the table) and models (bottom part of the table). For Mallows. Rounded to one decimal. The names of the models are explained as follows: The letters after the architecture type indicated how the voting rule is computed from the model: p–plain (i.e., no averaging), n–neutrality-averaged, na–neutrality-and-anonymity-averaged. The letters in the brackets indicate which axioms the model optimized for during training: NW–No winner, A–Anonymity, C–Condorcet, P–Pareto, I–Independence. All models have been trained for 15k gradient steps with batch size 200 and the same seed. The WEC by far was the fastest.

	WEC n	Stable Voting	Blacks	Borda	Weak Nanson	Copeland
WEC n	100	90.1	90.1	88.4	88	87.7
Stable Voting		100	96.22	91.57	93.28	92.23
Blacks			100	95.35	91.71	90.83
Borda				100	87.06	86.18
Weak Nanson					100	92.3
Copeland						100

Table 7: Similarities between the rules. Computed on 10.000 Mallows-sampled profiles. For example, the entry 90.1 in row ‘WEC n’ and column ‘Stable Voting’ means that, among the sampled profiles, in 90.1% of the cases the Stable Voting rule outputs the same winning set as the model neutrality-averaged WEC model.