DeepVoting: Learning and Improving Voting Rules with Fine-Tuning

Leonardo Matone
Tulane University
lmatone@tulane.edu

Ben Abramowitz
Tulane University
babramow@tulane.edu

Ben Armstrong
Tulane University
barmstrong@tulane.edu

Avinash Balakrishnan
IBM Research
avinash.bala@us.ibm.com

Nicholas Mattei
Tulane University
nmattei@tulane.edu

Abstract

Recently, social choice theory has been found broadly useful for AI alignment and evaluation, often based on the axiomatic properties for combining preferences provided by social choice functions. However, for many sets of axioms, functions which *never* violate the axioms are known to not exist, e.g., Arrow's Impossibility Theorem. Recent work on learning novel rules to *minimize* axiom violations shows the effectiveness of machine learning but explores a limited class of axioms. In this work we show the effectiveness of standard social choice data structures as features for learning existing voting rules, resulting in networks that have better accuracy than past approaches while using an order of magnitude fewer parameters. Subsequently, we develop a process for *adding axiomatic properties* to existing voting rules. By building novel axiomatic loss functions we are able to use transfer learning to fine-tune existing voting rules in a way that quantitatively preserves their original behaviour while also exhibiting strong adherence to previously absent axiomatic properties, resulting in novel voting rules that are *closer* to the impossibility frontier than those in the machine learning or theoretical literature.

1 Introduction

Preference aggregation is an important and widespread problem across many areas of computer science including information retrieval, recommender systems, reinforcement learning with human feedback, and agent evaluation. A common goal when choosing a preference aggregation function is that of axiom satisfaction: finding functions which *guarantee* certain mathematical properties. Unfortunately, dozens of "impossibility theorems" have shown that many sets of axioms cannot be mutually satisfied [11]. In this work we provide the foundational theoretical results, previously absent, required for a scientifically rigorous approach to learning voting rules. We demonstrate that novel rules, learned from base models trained to replicate existing well-understood rules, can be imbued with axiomatic properties not originally present. Our results highlight that in the learning of social choice axioms *some impossibilities are harder than others*.

Recently, a growing body of work has shown significant applicability of social choice theory to AI training [18], alignment [49, 29, 39] and evaluation [32, 33]. Learning social choice mechanisms can allow us to efficiently search for novel functions which provide desirable properties. Past work often shows that no single voting rule can perfectly align AI systems [41, 55], suggesting a demand for novel, tailored social choice functions. Other work has shown that probabilistic functions, i.e., lotteries, can provide stronger axiomatic performance for alignment settings [39].

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

This work provides a principled and rigorous methodology for learning both existing and novel voting rules. By using carefully chosen input embeddings we overcome the limitations of much past work, which has been limited by large networks with complex architectures [3], limited data [14], and the number of voters that can be considered simultaneously [3, 30]. We demonstrate that, across a wide range of voting rules, *we are able to train models which replicate existing rules better than the previous state-of-the-art with significantly fewer parameters*.

The majority of our results relate to axiomatic social choice. Axioms in social choice theory fall into one of two categories: intra-profile axioms (e.g., the majority winner property) which can be said to have been violated, or not, based on a social choice function and a single preference profile [24, 1] and inter-profile axioms which require more information. For example, the Participation property is violated if a voter can improve their outcome by not participating. To determine whether this occurs it is necessary to consider both the original profile and the profile in which the voter abstains [54].

While many axioms are intra-profile, many important axioms require evaluation of multiple alternative profiles, such as Participation and Strategyproofness. Little past work has explored the ability of machine learning to develop learned rules that minimize violations of inter-profile axioms [30]. Building upon base models trained to replicate existing rules, **our primary contribution is a fine-tuning approach which effectively adds new axiomatic properties into existing voting rules** by developing custom loss functions based on the principles of stochastic dominance. The end result of the process is a voting rule (in the form of a neural network) that behaves quantitatively similarly to an existing voting rule while exhibiting a novel axiomatic property.

Experimental studies of learned voting rules face two primary critiques: the relative complexity of using a neural network in place of a simple function, and reduced interpretability from the black-box nature of machine learning models. Both of these concerns are valid but tangential to our work; we aim first to provide a novel rigorous exploration of the data best suited for learning existing voting rules and then to develop novel voting rules with superior axiomatic properties to those known in the literature. A lack of interpretability is acceptable for us as our focus is on understanding the possibilities of learning, rather than deploying a rule for widespread use in human settings. However, in non-political settings that utilize social choice such as AI alignment or recommender systems, where deployed systems often aggregate orders using bandits or other learned, "black box" procedures [2, 45], the lack of simple, explainable rules is often a secondary concern to system efficiency.

Contribution. We (1) demonstrate, for the first time, that using common embeddings from social choice dramatically reduces the complexity and increases the efficiency of learning voting rules, and (2) use transfer learning (fine-tuning) to add axiomatic properties to learned rules. Our results are broadly useful for work using machine learning to develop voting rules and specifically useful as a learned probabilistic social choice function (i.e., a lottery) used in many modern AI evaluation schemes and recommendation systems where the increased efficiency from randomization is desirable.

2 Related Work

Machine learning has been applied to many tasks adjacent to voting, including auction design, matching, allocation mechanisms, and learning complex preference structures [19, 46, 35, 17, 47, 50].

While formal proposals to learn voting rules date back over a decade [48, 59], attention to learning voting rules has increased considerably in recent years. Kujawska et al. [31] and Burka et al. [14] used several common machine learning methods to mimic existing voting rules. However, both of these works overlooked the importance of the choice of embedding in the role of learning, finding that certain rules were "easier" to learn but not theoretically characterizing why certain embeddings maintain properties. Subsequently, Anil and Bao [3] showed that permutation-invariant architectures offer better generalization to larger numbers of voters. Hornischer and Terzopoulou [30] consider CNNs and additional architectures, and Li et al. [34] use graph neural networks. These papers continue to have a primary focus on learning existing voting rules and do not consider the ability of well-chosen embeddings to improve outcomes. While Armstrong [4] considered the impact of embeddings on rule learnability across a wide range of preference distributions the results were limited to only 5 candidates and provided little subsequent analysis.

Procaccia et al. [48] showed that positional scoring rules are efficiently PAC learnable, but learning pairwise comparison-based voting rules requires an exponential number of samples. Firebanks-

Quevedo [22] uses one axiomatic measure of optimality (Condorcet consistency) and strategyproofness for learning, but as we show, their choice of embedding does not preserve the information necessary to implement Condorcet-consistent rules, hindering performance. Wilson [58] focused on learning a voting rule given pair-wise relations and properties that must hold for the optimization criteria. However, they examine only the possibility of learning and do not employ any ML techniques.

The loss function chosen by Armstrong and Larson [5] was a function of the profile and outcome, and thus could learn a rule but not inter-profile axioms such as Participation. Recently, Mohsin et al. [42] focused on the problem of designing and/or learning fair and private rules using random forests and a subset of embeddings we study, proving that under differential privacy there is an upper bound on the trade-off between group fairness and efficiency. Caiata et al. [15] consider a wide range of intra-profile axioms in a multi-winner setting. They learn novel rules which violate axioms less than existing rules, but their methodology does not transfer to inter-profile axioms or finetuning of existing rules. Learning voting rules bears some similarity to the well studied area of learning to rank (L2R) from the machine learning literature [16]. L2R is concerned with accurate recovery of the *population preference* and not the axioms or properties of the aggregation method itself (e.g., fairness).

3 Preliminaries

Agents and Preference Profiles Let V be a set of n voters and C a set of m candidates. Each voter $i \in V$ reports a strict order x_i over all candidates in C as their ballot, i.e. a permutation of the candidates in C. We denote that i strictly prefers a over b by $a \succ_i b$ for $a, b \in C$. There are $m!$ possible ballots, or ways to strictly order (permute) the candidates in C. A list of n ballots, one for each voter, constitutes a *profile* $X = (x_i)_{i \in V}$. Voter i ranks candidate a at position $1 \leq x_i(a) \leq m$, and we use $x_i[l]$ to denote the lth element of x_i. Let $\mathcal{X}$ be the set of all possible profiles.

Probabilistic Social Choice Functions A probabilistic social choice function (PSCF) is a function $f : \mathcal{X} \to \Delta(C)$ that takes a profile $X \in \mathcal{X}$ as input and returns a *lottery*, or probability distribution $f(X) \in \Delta(C)$ over the set of candidates in the profile, where $\Delta(C)$ is the set of all lotteries over C. Let $\mathcal{F}$ be the set of all such PSCFs. We denote by $f(X)(a)$ the probability mass that f places on candidate a given profile X. Any PSCF can be used to construct a non-deterministic voting rule by sampling a winner from the lottery. The PSCFs we consider return a lottery that is a uniform distribution over a non-empty subset of the candidates, i.e., there are multiple potential winners (ties) that we choose from to construct a single-winner voting rule. Therefore, we let $U(Y)$ denote the uniform distribution over any finite set Y. We will abuse notation and let $f(X, c)$ denote the probability mass that lottery $f(X)$ allocates to candidate $c \in C$.

3.1 Embedding Social Preferences

Traditional feed-forward neural networks require a fixed-size input for learning and inference, corresponding to the size of their input layer. If we learn voting rules using networks that take the entire profile as input, then the input layer must be large $(m \times n)$ and networks cannot scale to work with arbitrarily large numbers of voters. Similarly, if the number of voters shrinks, then the network must be retrained or, as demonstrated by Hornischer and Terzopoulou [30], the profile may be padded carefully to preserve performance. To learn rules that are agnostic to the number of voters we need embeddings of a fixed-size that retain relevant information for profiles with any number of voters. Naturally, different embeddings preserve different information from the original profile, leading to differing efficacy when learning different rules and axioms. Most rules and axioms in the literature are defined for any positive number of voters, so we would like a learned rule to be similarly agnostic.

An embedding T is a function $T : \mathcal{X} \to \mathcal{X}'$ mapping profiles to some codomain $\mathcal{X}'$. The embeddings we are concerned with are many-to-one mappings. This means that multiple different profiles may have the same embedding and consequently, some information about the original profile is lost in the embedding. The following embeddings, illustrated in Figure 1, are drawn from the voting literature but are not commonly discussed as embeddings in the machine learning literature.

Definition 1 (Tournament Embedding) *Embedding T_T yields an $m \times m$ matrix M where $M[j, k] = 1$ if a majority of voters prefer j over k, $M[j, k] = \frac{1}{2}$ if an equal number of voters prefer each candidate, and $M[j, k] = 0$ otherwise.*

$v1 : A > B > C$
$v2 : A > B > C$
$v3 : B > C > A$
$v4 : C > B > A$

(a) Ballot Profile

	1st	2nd	3rd
A	2	0	2
B	1	3	0
C	1	1	2

(b) Rank Frequency Matrix

	A	B	C
A	0	½	½
B	½	0	1
C	½	0	0

(c) Tournament Matrix

	A	B	C
A	0	2	2
B	2	0	3
C	2	1	0

(d) Weighted Tournament Matrix

Figure 1: Three embeddings derived from the ballot profile. The size of the profile grows in $O(mn)$, while each embedding grows with $O(m^2)$, which is far smaller in the most common case, when $m \ll n$. However, these embeddings do not preserve all of the information in the original profile.

Definition 2 (Weighted Tournament Embedding) *Embedding T_{WT} yields an $m \times m$ matrix M where $M[j,k] = |\{i \in V : j \succ_i k\}|$ for $j, k \in C$.*

Definition 3 (Rank Frequency Embedding) *Embedding T_{RF} yields an $m \times m$ matrix M of how many voters rank each candidate $c \in C$ in each position $k \leq m$ where $M[c,k] = |\{i \in V \text{ s.t. } x_i(c) = k\}|$.*

Observe that T_{WT} contains strictly more information about the original profile than T_T as the tournament can be computed from the weighted tournament. Additionally, we normalize T_{RF} and T_{WT} by dividing all their elements by n. There is a tension in the literature between rules that use *positional information* and those that rely on *pairwise comparison* information [11]. Note T_{RF} maintains positional information while T_{WT} and T_T are majoritarian. At times we also refer to a concatenation of all three embeddings which we refer to as T_{CO}:

Definition 4 (Combined Embedding) *The combined embedding T_{CO} is the $3m^2$ concatenation of $[T_{RF}, T_T, T_{WT}]$.*

3.2 Probabilistic Social Choice Functions

We consider 8 unique PSCFs through the paper, listed in Table 1. Complete definitions of each rule can be found in Appendix A. We discuss here the information required to compute each PSCF. There exist numerous ways to categorize the informational requirements of voting rules; Fishburn [23] described three categories: C1 rules are those which can be computed using only information found in the tournament matrix T_T, C2 rules are those which can be computed using information found in the *weighted* tournament matrix T_{WT}, and all other rules are C3 rules. An alternative categorization is that of *positional scoring rules*, rules which can be computed using exactly the information of T_{RF}, the rank frequency embedding [11]. Rules can also be discussed in terms of the axioms they satisfy. We consider four axioms in this work, discussed in Section 3.3, and list in Table 1 which rules are known to satisfy or violate each axiom [11, 43].

Table 1: Informational requirements of PSCFs and which axioms each function satisfies.

	Fishburn	Scoring	Participation	Monotonicity	Strategyproof	Top Cycle
Plurality	C3	✓	✓	✓	×	×
Borda	C2	✓	✓	✓	×	×
Copeland	C1	×	×	✓	×	✓
Schulze	C2	×	×	✓	×	✓
Simpson-Kramer	C2	×	×	✓	×	×
IRV	C3	×	×	×	×	×
Black's Rule	C2	×	×	✓	×	×
Ranked Pairs	C2	×	×	✓	×	✓

3.3 Axiom Loss Functions

We consider four well-known social choice axioms – Top Cycle, Participation, Monotonicity, and Strategyproofness – and define novel loss functions that quantify how "close" a PSCF is to satisfying

each axiom. In Section 5 we demonstrate how to fine-tune pre-trained base models by training them on a combination of axiom loss function and rule loss that move them towards satisfaction of each individual axiom while maintaining rule performance.

It is useful to first introduce adjacent swaps and stochastic dominance. An *adjacent swap* refers to swapping the places of any two adjacent candidates in an ordering. With m candidates there are $m-1$ possible adjacent swaps for any ordering, e.g., given (a, b, c) the orderings (b, a, c) and (a, c, b) are the only two adjacent swaps. We denote by $s(X, i, j)$ the profile obtained by swapping candidates ranked at positions j and $j+1$ in x_i. *Stochastic dominance* (SD) is a relation between two lotteries with respect to a given ordering of the candidates. We use stochastic dominance to characterize voters' preferences over lotteries based on their preferences over candidates.

Definition 5 (Stochastic Dominance) *Given a strict ordering x over C and two lotteries P and Q, we say P stochastically dominates Q with respect to x if for all $k \leq m$, $\sum_{l \leq k} P(x[l]) \geq \sum_{l \leq k} Q(x[l])$.*

Definition 6 (Stochastic Dominance Loss) *Given ordering x over C, a reference lottery P, and an alternative lottery Q, we say that the stochastic dominance loss is zero if P stochastically dominates Q. If P does not stochastically dominate the alternative lottery Q, then the loss is equal to $L^{sd}(P|x, Q) = \max_{k \leq m}(\sum_{l \leq k} Q(x[l]) - \sum_{l \leq k} P(x[l]))$, i.e. the largest difference between the sums of prefixes of the lotteries over all prefixes when the distributions' supports are ordered by x.*

We are now ready to introduce our four axioms and define the four loss functions for measuring how close a PSCF is to satisfying them with respect to a given profile.

3.3.1 Axiom 1: Top Cycle

The Top Cycle of a profile is the smallest set of candidates such that all candidates inside the set preferred by a weak majority of voters to all candidates outside the set [11]. The Top Cycle axiom says that all probability mass must be on candidates in the Top Cycle. We translate this to a loss function by measuring the total probability mass allocated to candidates outside of the Top Cycle.

Definition 7 (Top Cycle Loss) *Let $TC(X)$ refer to the set of candidates in the Top Cycle for a profile X and f be a PSCF, then $L^{TC}(X, f) = 1 - \sum_{c \in TC(X)} f(X, c)$.*

3.3.2 Axiom 2: Monotonicity

A PSCF is (weakly) monotonic if swapping the positions of any two candidates in the ordering of any single voter never decreases the probability mass on the candidate swapped upward. A PSCF is adjacent-swap monotonic (ASM) if swapping any two adjacent candidates in a single ordering within the profile will never decrease the probability mass on the candidate swapped upward. These properties are equivalent: Monotonicity clearly implies ASM, and ASM implies monotonicity as we can create a series of adjacent swaps that move one candidate up and the candidate formerly in that position down to the other candidates' original position. We characterize our loss function in terms of ASM as the space of all $n(m-1)$ possible adjacent swaps in a profile is much smaller than the space of all $\frac{nm(m-1)}{2}$ possible swaps, and we therefore expect to learn ASM more easily. We generalize ASM to consider settings where only the k top candidates in a voter's ordering can be swapped. When $k = m-1$ this is equivalent to full ASM. A PSCF f satisfies Adjacent-Swap Monotonicity if for all possible profiles $X \in \mathcal{X}$, for all voters $i \in V$, if we take any two adjacent candidates in x_i where $c \succ_i c'$ and swap them, this does not decrease the probability mass allocated to c'.

Definition 8 (Adjacent-Swap Monotonicity Loss) *Given a profile X and an index k, $L^{mono}(f, X, k) = \max\big(0, \max_{i \in V, j \leq k} f(X, x_i[j]) - f(s(X, i, j), x_i[j-1])\big)$.*

3.3.3 Axiom 3: Participation

A PSCF satisfies Participation if no voter prefers the outcome when abstaining over casting a truthful ballot. Violation of Participation is referred to as the No Show Paradox.

We denote by X^{-i} the profile obtained when voter i's vote is removed from X, representing an abstention. We say that voter i prefers voting truthfully to abstaining when $f(X)$ stochastically dominates $f(X^{-i})$ with respect to x_i. We want our PSCF to be immune to strategic abstentions. A PSCF f obeys Participation if, $\forall\ X \in \mathcal{X}$, and $\forall\ i \in V$, $f(X) \succ_i f(X^{-i})$, so every voter prefers truthful voting to abstention because $f(X)$ stochastically dominates $f(X^{-i})$ according to x_i.

Definition 9 (Participation Loss) *Given a profile X and PSCF f, $L^{part}(f, X) = \max_{i \in V} L^{sd}(f(X)|x_i, f(X^{-i}))$, the maximum SD loss for any single abstaining voter.*

3.3.4 Axiom 4: Strategyproofness

Adjacent-Swap Strategyproofness (ASSP) was originally described as a means of capturing whether minor changes to a voter's preference order ("small lies") may provide useful nonmanipulability results but has been shown to be equivalent to general forms of strategyproofness on universal preference domains [52, 26].

A PSCF satisfies ASSP if there is no profile for which a single voter i can make an adjacent swap in their preferences to create an outcome that is not stochastically dominated by the truthful outcome with respect to x_i. Formally, a PSCF is adjacent-swap strategyproof if $\forall\ X \in \mathcal{X}, \forall\ i \in V, \forall\ j \in 1 \ldots m-1$, $L^{\text{sd}}(f(X)|x_i, f(s(X, i, j))) = 0$. We generalize ASSP to consider strategyproofness in which only the k top candidates in a voter's ordering can be swapped, and define a corresponding loss function. For experimental tractability we limit k, but when $k = m - 1$ this is equivalent to full ASSP.

Definition 10 (Adjacent-Swap Strategyproofness Loss) *Given a profile X and an index k, $L^{assp}(f, X, k) = \max_{i \in V,\ j \leq k} L^{sd}(f(X)|x_i, f(s(X, i, j)))$.*

4 Social Choice Embeddings Enable SOTA Rule Learning

To achieve our primary goal of learning *novel* rules which behave in a quantitatively similar manner to existing rules, we first show that we are able to achieve state-of-the-art performance at learning existing rules by using the input embeddings defined in Section 3. Using networks with a simpler structure and fewer parameters we surpass or match the performance of all past work. These networks serve as base models for our subsequent fine-tuning task.

Our embeddings provide three key advantages: (1) They dramatically reduce the size of our input layer, therefore greatly reducing the number of model weights. (2) When a rule is paired with an appropriate embedding, the embedding preserves all the information necessary to learn the rule. (3) Input size no longer depends on the number of voters; as a result training costs increase only with the number of candidates and a trained model can be used for profiles containing any number of voters.

Experimental Setup and Network Details We train our base models on profiles with $n = 44$ voters and $m = 11$ candidates. For all experiments, profiles are sampled from the *Impartial Culture* distribution – i.e., rankings are generated uniformly at random [6]. We use the Whalrus package for voting rule implementations.[1]

We use an MLP architecture similar to that used by Anil and Bao [3], with 5 fully-connected layers, the first hidden layer with 200 nodes, then four with 120 nodes, layer normalization between each hidden layer, TanH activation functions, a Softmax output layer.[2] The key difference is that our network takes in embedded profiles so the size of our input layer is m^2 (T_T, T_{RF}, T_{WT}) or $3m^2$ (T_{CO}) compared to their nm^2. This reduces the total number of parameters in our model from 10 million to ≈150K. We train our models on a set of 100,000 randomly sampled profiles in batches of size 32 for 1000 epochs. Models are trained 5 times using different seeds. We train each model on an NVIDIA A100 GPU with each model requiring ≈4 hours. We used the Adam optimizer with an initial learning rate of 0.001, reducing on plateau (patience = 50, factor = 0.5, min_lr = $1e^{-5}$).

Models receive as input one or three flattened embedding matrices. e.g., models trained on T_{CO} receive: the rank frequency, tournament, and weighted tournament matrices. For each PSCF we train a base model on **L1 Loss**, the L1 distance between the output of the model and the target PSCF.

[1] https://pypi.org/project/whalrus/

[2] Discussion of other setups is in Appendix E.

Table 2: Single winner accuracy of our base models compared with prior work on rules considered in past work. Using fewer parameters we are able to nearly match, or exceed, all past results.

	# Parameters	Plurality	Borda	Copeland	SK
DeepVoting (ours)	**150K**	0.99	0.95	**1.00**	**0.80**
Anil and Bao [3]	10M	**1.00**	0.99	0.83	**0.80**
Hornischer and Terzopoulou [30]	1.2M - 1.8M	0.99	0.95	0.90	N/A
Li et al. [34]	100K-1M	**1.00**	**1.00**	0.76	0.73

Table 3: Accuracy of our models on each embedding using test data from the Impartial Culture ($m = 11, n = 44$), on profiles that may contain ties. A prediction counts as correct if the alternative receiving the most predicted probability mass is a (potentially tied) winner. Cells show mean $\pm$ standard error over 5 independently trained models. Most accurate embeddings are shown in bold. Shaded cells indicate the embedding contains sufficient information to learn the rule perfectly.

	T_{RF}	T_T	T_{WT}	T_{CO}
Plurality	**0.9887 $\pm$ 0.0254**	0.3284 $\pm$ 0.0049	0.3290 $\pm$ 0.0031	0.9884 $\pm$ 0.0259
Borda	0.8168 $\pm$ 0.0618	0.6945 $\pm$ 0.0021	0.8945 $\pm$ 0.0387	**0.9501 $\pm$ 0.0036**
Copeland	0.6403 $\pm$ 0.0304	0.9999 $\pm$ 0.0001	0.7184 $\pm$ 0.0094	**1.0000 $\pm$ 0.0000**
Schulze	0.6930 $\pm$ 0.0156	0.8115 $\pm$ 0.0032	0.7144 $\pm$ 0.0049	**0.8119 $\pm$ 0.0083**
Simpson-Kramer	0.6924 $\pm$ 0.0125	0.7969 $\pm$ 0.0195	0.7045 $\pm$ 0.0124	**0.8036 $\pm$ 0.0023**
IRV	0.3781 $\pm$ 0.0228	0.3766 $\pm$ 0.0158	0.3622 $\pm$ 0.0139	**0.4061 $\pm$ 0.0220**
Black's Rule	0.7856 $\pm$ 0.0380	0.7099 $\pm$ 0.0021	**0.9077 $\pm$ 0.0353**	0.8654 $\pm$ 0.0109
Ranked Pairs	0.4675 $\pm$ 0.0304	**0.6778 $\pm$ 0.0349**	0.5189 $\pm$ 0.0227	0.6561 $\pm$ 0.0497

Base Model Performance Table 3 gives the final validation accuracy and Figure 2 shows plots of validation losses during training, using 10,000 samples from a held-out validation set, to demonstrate the effectiveness of learning for each rule-embedding pair. Most rules reach their lowest loss quite quickly, with the exception of Borda's and Black's rules. Among settings where the embedding does not provide sufficient information to learn a rule completely we do often see limited loss reduction (e.g., T_T with Ranked Pairs). Rule loss on models trained with T_T is often much lower than random, suggesting that the simple binary structure of the T_T matrix is relatively easy to learn from. On the other hand, while the T_{WT} features contain strictly more information than T_T they often result in much less learning than T_{CO} even on C1/C2 rules which are computable from T_{WT} (e.g., Copeland, Schulze, Simpson-Kramer, Ranked Pairs). In these cases the T_{CO} embedding rarely outperforms T_T, suggesting that the network may only be learning from the simpler T_T features.

An unexpected result in Table 3 is the limited accuracy of the Borda network trained using T_{RF}, despite the fact that T_{RF} contains sufficient information to learn the rule. We hypothesize that this is due to the relative complexity of calculating Borda using T_{RF}. While Borda scores can be calculated from T_{WT} using only the sum or mean of each row, calculating the same scores using T_{RF} requires multiplying by the Borda score vector then summing. That is, computing Borda from T_{RF} requires more mathematical operations than computing Borda from T_{WT}. Recent work exploring the difficulty of learning basic mathematical operations supports this hypothesis [10].

We find that intuitively simpler rules (Plurality, Borda, and Copeland) are easier to learn, while rules based on iterative procedures that require candidate comparisons or other computations in each iteration, like IRV and Ranked Pairs, are hardest to learn. These more complex rules are also the only rules we study which violate the Monotonicity criterion, suggesting a possible link between Monotonicity and learnability. We also see that rules which are efficiently PAC-learnable, such as Plurality and Borda, are learned more easily than others [48].

Single Winner Evaluation While our rules are trained to generate a probability distribution over candidates, we can also treat the goal as a single winner prediction task. This allows us to directly compare with the *subset accuracy* metric of Hornischer and Terzopoulou [30]; the winner chosen by a network for a single profile is the candidate receiving the most probability mass. If this is the winner, or one of the tied winners, the output is considered correct. Using 10,000 test profiles we evaluate the accuracy of each trained model on the corresponding voting rule. Table 3 shows the test accuracy for each rule and each embedding. While we also learn several rules not studied in prior

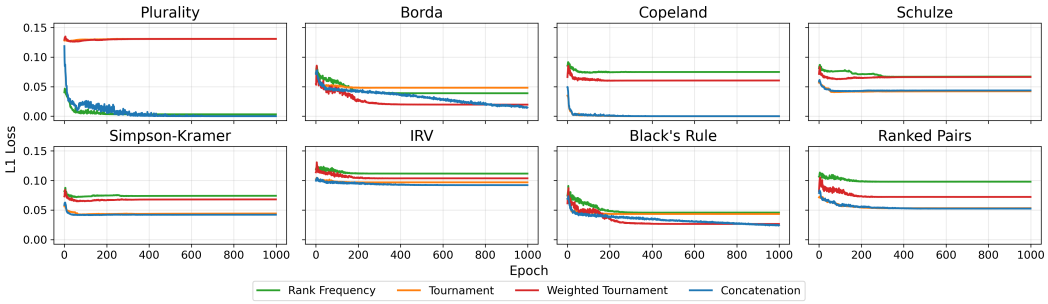

Figure 2: Validation Loss per epoch for each rule and embedding pair.

Table 4: Test accuracy on 10,000 profiles ($m = 11, n = 44$) from each preference distribution, for networks trained using Impartial Culture preferences and the T_{CO} embedding. A prediction is correct when the highest-scoring candidate is the winner; these test sets contain no tied winners. Cells show mean $\pm$ standard error of the mean over 5 independently trained models.

Target Rule	IAC	Urn	Mall.	SP	PrefL
Plurality	1.000 $\pm$ .000	.989 $\pm$ .002	.996 $\pm$ .001	.995 $\pm$ .003	.973 $\pm$ .002
Borda	.975 $\pm$ .001	.570 $\pm$ .006	.477 $\pm$ .006	.329 $\pm$ .138	.381 $\pm$ .005
Copeland	1.000 $\pm$ .000	1.000 $\pm$ .000	1.000 $\pm$ .000	1.000 $\pm$ .000	.980 $\pm$ .000
Schulze	.917 $\pm$ .004	.894 $\pm$ .008	.916 $\pm$ .010	.957 $\pm$ .015	.861 $\pm$ .010
Simpson-Kramer	.919 $\pm$ .001	.898 $\pm$ .005	.921 $\pm$ .007	.804 $\pm$ .166	.866 $\pm$ .009
IRV	.517 $\pm$ .016	.601 $\pm$ .027	.652 $\pm$ .033	.514 $\pm$ .036	.642 $\pm$ .033
Black's Rule	.915 $\pm$ .002	.565 $\pm$ .006	.491 $\pm$ .007	.560 $\pm$ .127	.398 $\pm$ .010
Ranked Pairs	.802 $\pm$ .043	.822 $\pm$ .046	.854 $\pm$ .049	.968 $\pm$ .005	.832 $\pm$ .043

work, Table 2 shows that **well-chosen embeddings leads to state of the art results using a network with 1-2 orders of magnitude fewer parameters** and a simpler architecture than previous results.

Beyond Impartial Culture Table 4 shows the accuracy of networks trained on T_{CO} tested on 10,000 profiles from several preference distributions, and Table 10 shows the loss of these networks. In most cases, rules trained on T_{CO} are able to generalize effectively to new distributions. On highly structured domains, such as Single-Peaked (SP), some rules (e.g., Black's, Ranked Pairs) have *better* performance than on the Impartial Culture.[3] Each preference distribution is defined in Appendix D.1.

Of the "simple" rules, the Copeland rule generalizes almost perfectly across all tested distributions, with Plurality not far behind, while Borda performed surprising poorly outside of the impartial domains (IC and IAC). For all rules other than IRV, either the SP distribution or real-world PrefLib data proved the most challenging, but the shockingly poor generalization of Borda to the SP distribution warrants further investigation. IRV generalizes poorly across the board, which is not surprising considering its performance on IC, but it is surprising that IRV beats Ranked Pairs and Borda on the SP distribution, and its lowest test accuracy was in the IC distribution on which it was trained. Each rule attained similar test accuracy on IAC as it on IC, and performance tended to be similar between Urn and Mallows as well. Of all the theoretical distributions, the Mallows and Urn models showed generalization accuracies closes to the empirical PrefLib data, suggesting that perhaps these model classes correspond more closely to real-world preferences than IC, IAC, or SP.

The fact that model performance generalizes so well beyond impartial culture is quite surprising at first. Impartial culture represents a "uniform random" of preferences, yet appears highly informative about other preference distributions. This may stem from two factors: As every profile has equal probability of being generated under IC we believe networks effectively learn profiles that are like those which would be sampled from other distributions. Additionally, by focusing learning on settings where many candidates receive similar levels of support the networks are actually learning a *harder* task than if they were to simply learn on a highly structured distribution such as Mallows [36].

[3]Note that these results are from an experiment using profiles generated by sampling from each preference distribution normally and excluding a profile if it results in tied winners from any voting rule.

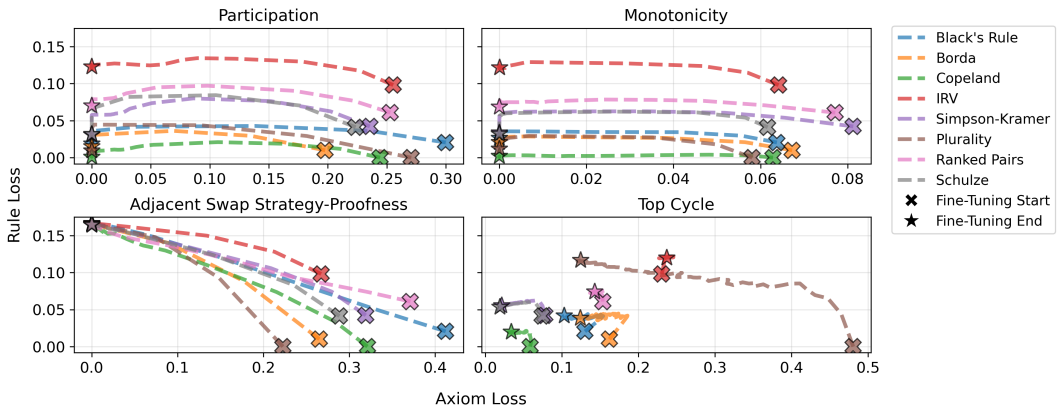

Figure 3: Axiom validation loss and rule validation loss during finetuning of models trained on each learned rule using T_{CO} features.

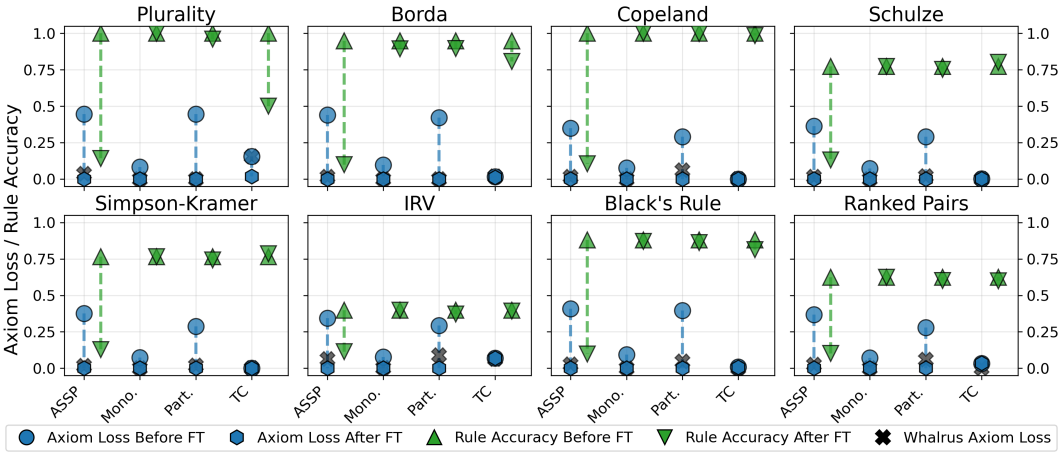

Figure 4: Axiom loss and rule accuracy for each rule after training on rule loss (base model) and after finetuning.

5 Fine-Tuning for Axiom Satisfaction

We now turn to our primary contribution: using transfer learning to *adapt the existing base models* trained in Section 4 for improved axiomatic performance by applying a second stage of learning using the four axiomatic loss (combined with rule loss) functions we defined in Section 3.3.

Brandt et al. [13] show that Condorcet Consistency (which Top Cycle implies) and strong SD-strategyproofness are incompatible for PSCFs. Likewise, Brandt et al. [12] show that Top Cycle and strong SD-participation are also impossible. Gibbard [26] and Sato [52] show that adjacent strong SD-strategyproofness is equivalent to strong SD-strategyproofness, hence we need only test ASSP to ensure strategyproofness. In fact, Top Cycle mixed with either ASSP or Participation is a *known impossibility* for PSCFs. Hence, in this section we set *impossible voting rules* as our learning target.

Experimental Setup Fine-tuning uses the same architectures, and profile size as our initial training. We use a batch size of 20 for 100 epochs, training on an NVIDIA A100 GPU for $\approx$ 18 hours per network. Models are trained to minimize the sum of *axiom loss* and *rule loss*, allowing simultaneous propagation of both losses through the network. Our inter-profile axioms require exhaustive comparison against n alternative profiles for Participation (each single abstention) and $n(m-1)$ for Monotonicity and Strategyproofness ($m-1$ adjacent swaps for each voter), which becomes computationally prohibitive for even low values of n. To mitigate this, we pre-compute a set of *profitable* alternative profiles for each original profile in our new set. A profitable swap or abstention is one

which results in a profile that stochastically dominates the original profile. We generate train and test sets of 10,000 profiles each by sampling profiles from the impartial culture distribution and keeping profiles which have a non-zero loss on at least one PSCF and axiom (from ASSP, Participation, or Monotonicity). We restrict our search of adjacent swaps to $k = 5$, for computational tractability and as previous studies have found that human voters nearly always strategize near the top of their preference profile [40, 53]. This pre-computation greatly reduces the runtime of our loss functions.

Performance of Fine-Tuned PSCFs Figure 3 shows how axiom and rule losses change over fine-tuning. Figure 4 gives final axiom loss and rule accuracy on a test set of 10,000 profiles.

Three patterns emerge: **(1)** Participation and Monotonicity are learned well by all rules. Axiom loss approaches zero with little-to-no increase in rule loss. For rules satisfying Top Cycle, such as Copeland, which is incompatible with Participation, **these results show a near-impossible outcome**: a version of Copeland which (nearly) satisfies Participation. We find this learned "Copeland + Participation" rule primarily differs from Copeland when a voter has a profitable abstention. In such cases, the rule tends to place most probability mass on the alternative that would have been elected in the absence of the potential abstainer. On the other hand, **(2)** ASSP is learned effectively by each rule but at the expense of rule loss: The axiom teaches rules to select a random winner, trivially satisfying strategyproofness. **(3)** Top Cycle shows additional complexity: the Plurality model illustrates the tradeoff between mimicking Plurality and satisfying Top Cycle. Appendix G provides extensive results comparing model performance on each axiom and PSCF before and after fine-tuning.

6 Learning Axioms Directly

Table 5: Validation loss for each axiom loss function evaluated on models trained purely on the sum of axiom losses.

Axioms	Part.	Mono.	ASSP	Top Cycle
ASSP + TC	0	0	0	0.224
Part. + TC	0	0	0.448	0.12
All Axioms	0	0	0	0.224

In addition to fine-tuning networks we can also directly train models from scratch for axiom loss. We formed three sets of our axioms, each representing a theoretically impossible combination of axioms. Models are trained for 100 epochs using a loss function equal to the sum of each individual axiom loss function. Table 5 shows the validation loss after training models on each axiom set. Results are consistent with fine-tuning: Participation and Monotonicity are easy to learn while Strategyproofness and Top Cycle exhibit more complexity. Appendix F shows the training curve of each model and provides additional discussion. We include these results as a proof-of-concept, rather than to exemplify the best or most interesting axiom combinations to learn.

7 Conclusions and Future Work

This paper shows that learned voting rules can be imbued with theoretically (almost) impossible axiomatic properties. We first show standard embeddings allow learning existing voting rules with (1) similar or higher accuracy, and (2) a simpler architecture and a fraction of the parameters required by most previous approaches [3, 30, 34]. Subsequently, we show that we can fine-tune a trained model to maintain excellence at approximating a learned voting rule while also significantly reducing axiom violations. Different axioms present distinct challenges and come in three varieties. Axioms may be (1) learned well without reducing rule performance (e.g., Participation, Monotonicity), (2) learned at the cost of rule performance (e.g., ASSP), or (3) difficult to learn in general (e.g., Top Cycle).

These findings suggest that learning social choice axioms directly, in contrast with previous work which has avoided this challenge at the expense of generality [15], will pose novel theoretical challenges to the field. Our results demonstrate that the landscape of axiom loss functions varies dramatically between axioms; popular axioms such as Top Cycle may present a surprising challenge as a learning target. This work can be directly applied to tasks in AI alignment and evaluation which call for novel social choice functions with diverse axiomatic priorities [32, 51].

Acknowledgements

Matone, Armstrong, and Mattei were supported in part by NSF Awards IIS-RI-2007955, IIS-III-2107505, IIS-RI-2134857, IIS-RI-2339880 and CNS-SCC-2427237 as well as the Harold L. and Heather E. Jurist Center of Excellence for Artificial Intelligence at Tulane University and the Tulane University Center of Excellence for Community Engaged Artificial Intelligence (CEAI). Ben Abramowitz was supported by the NSF under Grant #2127309 to the Computing Research Association for the CIFellows Project. Portions of this research were conducted with high performance computational resources provided by the Louisiana Optical Network Infrastructure (LONI) (http://www.loni.org).

References

[1] B. Abramowitz and N. Mattei. Axiomatic Choice and the Decision-Evaluation Paradox. *arXiv preprint arXiv:2509.21836*, 2025.

[2] A. Aird, P. Farastu, J. Sun, E. Stefancová, C. All, A. Voida, N. Mattei, and R. Burke. Dynamic Fairness-Aware Recommendation Through Multi-Agent Social Choice. *ACM Transactions on Recommender Systems*, 3(2):1–35, 2024.

[3] C. Anil and X. Bao. Learning to Elect. *Advances in Neural Information Processing Systems*, 34:8006–8017, 2021.

[4] B. Armstrong. The Power of Experimental Approaches to Social Choice. *PhD Thesis*, 2025.

[5] B. Armstrong and K. Larson. Machine Learning to Strengthen Democracy. In *NeurIPS Joint Workshop on AI for Social Good*, 2019.

[6] D. Black. *The Theory of Committees and Elections*. Springer, 1958.

[7] N. Boehmer, R. Bredereck, P. Faliszewski, R. Niedermeier, and S. Szufa. Putting a compass on the map of elections. *arXiv preprint arXiv:2105.07815*, 2021.

[8] N. Boehmer, R. Bredereck, E. Elkind, P. Faliszewski, and S. Szufa. Expected Frequency Matrices of Elections: Computation, Geometry, and Preference Learning. *Advances in Neural Information Processing Systems*, 35:31347–31359, 2022.

[9] N. Boehmer, P. Faliszewski, Ł. Janeczko, A. Kaczmarczyk, G. Lisowski, G. Pierczyński, S. Rey, D. Stolicki, S. Szufa, and T. Was. Guide to Numerical Experiments on Elections in Computational Social Choice. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 7962–7970, 2024.

[10] C. Both, B. Hoover, H. Strobelt, D. Krotov, D. K. I. Weidele, M. Martino, and N. Dehmamy. Small Models, Smarter Learning: The Power of Joint Task Training. *arXiv preprint arXiv:2505.18369*, 2025.

[11] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[12] F. Brandt, C. Geist, and D. Peters. Optimal Bounds for the No-Show Paradox via SAT Solving. *Mathematical Social Sciences*, 90:18–27, 2017.

[13] F. Brandt, P. Lederer, and R. Romen. Relaxed Notions of Condorcet-Consistency and Efficiency for Strategyproof Social Decision Schemes. *Social Choice and Welfare*, 63(1):19–55, 2024.

[14] D. Burka, C. Puppe, L. Szepesváry, and A. Tasnádi. Voting: A Machine Learning Approach. *European Journal of Operational Research*, 299(3):1003–1017, 2022.

[15] J. Caiata, B. Armstrong, and K. Larson. What Voting Rules Actually Do: A Data-Driven Analysis of Multi-Winner Voting. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence*, 1 2026.

[16] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to Rank: From Pairwise Approach to Listwise Approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

[17] M. J. Curry, U. Lyi, T. Goldstein, and J. P. Dickerson. Learning Revenue-Maximizing Auctions with Differentiable Matching. In *International Conference on Artificial Intelligence and Statistics*, pages 6062–6073. PMLR, 2022.

[18] J. Dai and E. Fleisig. Mapping Social Choice Theory to RLHF. *arXiv preprint arXiv:2404.13038*, 2024.

[19] P. Dütting, Z. Feng, N. Golowich, H. Narasimhan, D. C. Parkes, and S. S. Ravindranath. Machine Learning for Optimal Economic Design. In *The Future of Economic Design*, pages 495–515. Springer, 2019.

[20] F. Eggenberger and G. Pólya. Über die Statistik Verketteter Vorgänge. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 3 (4):279–289, 1923.

[21] U. Endriss. *Trends in Computational Social Choice*. AI Access, 2017.

[22] D. Firebanks-Quevedo. Machine Learning? In MY Election? It's More Likely Than You Think: Voting Rules via Neural Networks. *Oberlin College Honors Thesis 688*, 2020.

[23] P. C. Fishburn. Condorcet Social Choice Functions. *SIAM Journal on applied Mathematics*, 33 (3):469–489, 1977.

[24] P. C. Fishburn. *The Theory of Social Choice*. Princeton University Press, 2015.

[25] W. V. Gehrlein and P. C. Fishburn. Condorcet's Paradox and Anonymous Preference Profiles. *Public Choice*, pages 1–18, 1976.

[26] A. Gibbard. Manipulation of Schemes that Mix Voting with Chance. *Econometrica: Journal of the Econometric Society*, pages 665–681, 1977.

[27] G. T. Guilbaud. Les Théories de l'Intérêt Général et le Problème Logique de l'Agrégation. *Economie appliquée*, 5(4):501–584, 1952.

[28] D. Halpern, S. Hossain, and J. Tucker-Foltz. Computing Voting Rules with Elicited Incomplete Votes. In *Proceedings of the 25th ACM Conference on Economics and Computation*, pages 941–963, 2024.

[29] B. Heymann. Adaptive Preference Aggregation. *arXiv preprint arXiv:2503.10215*, 2025.

[30] L. Hornischer and Z. Terzopoulou. Learning How to Vote with Principles: Axiomatic Insights into the Collective Decisions of Neural Networks. *Journal of Artificial Intelligence Research*, 83, 2025.

[31] H. Kujawska, M. Slavkovik, and J.-J. Rückmann. Predicting the Winners of Borda, Kemeny and Dodgson Elections with Supervised Machine Learning. In *Multi-Agent Systems and Agreement Technologies: 17th European Conference, EUMAS 2020, and 7th International Conference, AT 2020, Thessaloniki, Greece, September 14-15, 2020, Revised Selected Papers 17*, pages 440–458. Springer, 2020.

[32] M. Lanctot, K. Larson, Y. Bachrach, L. Marris, Z. Li, A. Bhoopchand, T. Anthony, B. Tanner, and A. Koop. Evaluating Agents Using Social Choice Theory. *arXiv preprint arXiv:2312.03121*, 2023.

[33] M. Lanctot, K. Larson, I. Gemp, and M. Kaisers. Active Evaluation of General Agents: Problem Definition and Comparison of Baseline Algorithms. *arXiv preprint arXiv:2601.07651*, 2026.

[34] H. X. Li, Y. Shah, and L. Giusti. Learning Resilient Elections with Adversarial GNNs. *arXiv preprint arXiv:2601.01653*, 2026.

[35] A. Malakhov and R. V. Vohra. Optimal Auctions for Asymmetrically Budget Constrained Bidders. *Review of Economic Design*, 12(4):245–257, 2008.

[36] C. L. Mallows. Non-Null Ranking Models. I. *Biometrika*, 44(1/2):114–130, 1957.

[37] N. Mattei and T. Walsh. Preflib: A library for preferences http://www. preflib. org. In *International conference on algorithmic decision theory*, pages 259–270. Springer, 2013.

[38] N. Mattei and T. Walsh. A PrefLib.org Retrospective: Lessons Learned and New Directions. *Trends in Computational Social Choice*, pages 289–309, 2017.

[39] R.-R. Maura-Rivero, M. Lanctot, F. Visin, and K. Larson. Jackpot! Alignment as a Maximal Lottery. *arXiv preprint arXiv:2501.19266*, 2025.

[40] T. Mennle, M. Weiss, B. Philipp, and S. Seuken. The Power of Local Manipulation Strategies in Assignment Mechanisms. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[41] A. Mishra. AI Alignment and Social Choice: Fundamental Limitations and Policy Implications. *arXiv preprint arXiv:2310.16048*, 2023.

[42] F. Mohsin, A. Liu, P.-Y. Chen, F. Rossi, and L. Xia. Learning to Design Fair and Private Voting Rules. *Journal of Artificial Intelligence Research*, 75:1139–1176, 2022.

[43] H. Moulin. On Strategy-proofness and Single Peakedness. *Public Choice*, 35(4):437–455, 1980.

[44] H. Moulin. Condorcet's Principle Implies the No Show Paradox. *Journal of Economic Theory*, 45(1):53–64, 1988.

[45] G. K. Patro, A. Biswas, N. Ganguly, K. P. Gummadi, and A. Chakraborty. Fairrec: Two-sided Fairness for Personalized Recommendations in Two-sided Platforms. In *Proceedings of the Web Conference 2020*, pages 1194–1204, 2020.

[46] G. Pavlov. Optimal Mechanism for Selling Two Goods. *The BE Journal of Theoretical Economics*, 11(1), 2011.

[47] N. Peri, M. Curry, S. Dooley, and J. Dickerson. PreferenceNet: Encoding Human Preferences in Auction Design with Deep Learning. *Advances in Neural Information Processing Systems*, 34: 17532–17542, 2021.

[48] A. D. Procaccia, A. Zohar, Y. Peleg, and J. S. Rosenschein. The Learnability of Voting Rules. *Artificial Intelligence*, 173(12-13):1133–1149, 2009.

[49] A. D. Procaccia, B. Schiffer, and S. Zhang. Clone-robust AI Alignment. *arXiv preprint arXiv:2501.09254*, 2025.

[50] S. S. Ravindranath, Z. Feng, S. Li, J. Ma, S. D. Kominers, and D. C. Parkes. Deep Learning for Two-sided Matching. *arXiv preprint arXiv:2107.03427*, 2021.

[51] M. Rofin, V. Mikhailov, M. Florinsky, A. Kravchenko, T. Shavrina, E. Tutubalina, D. Karabekyan, and E. Artemova. Vote'n'Rank: Revision of Benchmarking with Social Choice Theory. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 670–686, 2023.

[52] S. Sato. A Sufficient Condition for the Equivalence of Strategy-proofness and Nonmanipulability by Preferences Adjacent to the Sincere One. *Journal of Economic Theory*, 148(1):259–278, 2013.

[53] J. Scheuerman, J. Harman, N. Mattei, and K. B. Venable. Modeling Voters in Multi-winner Approval Voting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5709–5716, 2021.

[54] M. C. Schmidtlein. Voting by Axioms. *Master's thesis. ILLC, University of Amsterdam*, 2022.

[55] A. Siththaranjan, C. Laidlaw, and D. Hadfield-Menell. Distributional Preference Learning: Understanding and Accounting for Hidden Context in RLHF. *arXiv preprint arXiv:2312.08358*, 2023.

[56] S. Szufa, N. Boehmer, R. Bredereck, P. Faliszewski, R. Niedermeier, P. Skowron, A. Slinko, and N. Talmon. Drawing a Map of Elections. *Artificial Intelligence*, 343, 2025.

[57] T. Walsh. Generating Single Peaked Votes. *arXiv preprint arXiv:1503.02766*, 2015.

[58] N. Wilson. Generating Voting Rules from Random Relations. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2267–2269, 2019.

[59] L. Xia. Designing Social Choice Mechanisms Using Machine Learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 471–474, 2013.

Appendix

All code used in this paper is openly available on GitHub: `https://github.com/V993/deep-vote`. Authors are happy to discuss additional experiments, future work, or to explain any aspect of the paper.

A Voting Rule Definitions

In this section we give full definitions of each voting rule under study and describe how they are represented as probabilistic social choice functions. We have chosen a wide range of rules to represent those used commonly in practice (e.g., Plurality, IRV) as well as those often studied in social choice settings. Most of the rules function quite differently from one another and have a wide range of informational requirements, providing a strong base from which to explore best practices for learning voting rules. See the Whalrus package documentation for implementation details.

Definition 11 (Plurality) *The Plurality score of candidate $c \in C$ from profile X is $L(c) = |\{i \in V : x_i(c) = 1\}|$. Let $W(X) = \arg\max_{c \in C} L(c)$ be the subset of candidates with maximum Plurality score. The probabilistic Plurality rule returns the lottery $U(W(X))$.*

Definition 12 (Borda) *The Borda score of candidate $c \in C$ from profile X is $B(c) = \sum_{i \in V} (m - x_i(c))$. Let $W(X) = \arg\max_{c \in C} B(c)$ be the subset of candidates with maximum Borda score. The probabilistic Borda rule returns the lottery $U(W(X))$ (Referred to as Borda Max by Endriss [21]).*

Definition 13 (Copeland) *The Copeland score of candidate $c \in C$ from profile X is the number of other candidates it beats in pairwise competition plus $\frac{1}{2}$ times the number of other candidates it ties with in direct competition (if n is even). Let $W(X)$ be the subset of candidates with maximum Copeland score on profile X. The probabilistic Copeland rule returns the lottery $U(W(X))$.*

The Simpson-Kramer (Maximin), Schulze, and Ranked Pairs rules are each computed from T_{WT}. We will call these weighted-tournament rules. Let $G_X(C, E)$ be the directed tournament graph with edges corresponding to all positive values of the tournament matrix induced by X. Let each directed edge $(a, b) \in E$ have weight $d(a, b) = |i \in V : a \succ_i b|$.

Definition 14 (Schulze) *For each path from a to b in G_X, we let the strength of the path be the minimum weight edge in that path. For each pair of candidates $a, b \in C$ with a path from a to b, we let $p(a, b)$ be the maximum strength of any path from a to b, and let $p(a, b) = 0$ otherwise. Finally, let $W(X) = \{a \in C : p(a, b) \geq p(b, a)$ for all $b \in C\}$. The probabilistic Schulze rule returns the lottery $U(W(X))$.*

Definition 15 (Simpson-Kramer) *Let W be the subset of candidates whose maximum weight incoming edge is minimal in G_X. The probabilistic Simpson-Kramer rule returns the lottery $U(W(X))$.*

Definition 16 (Instant Runoff Voting (IRV)) *IRV is a deterministic, iterative voting rule that, in each of $m - 1$ rounds, eliminates the candidate with the lowest plurality score and removes them from the preference orders of all voters before the next round. When candidates are tied for lowest plurality score we break ties lexicographically. The rule returns the lottery that assigns all probability to the single candidate that was never eliminated; $U(W(X))$ where $|W(X)| = 1$.*

Definition 17 (Black's Rule) *If the profile X admits a Condorcet winner c, then let $W(X) = \{c\}$. Otherwise, if there is no Condorcet winner, let $W(X)$ be the subset of candidates with maximum Borda score $B(c)$. The probabilistic Black's rule returns the lottery $U(W(X))$.*

Definition 18 (Ranked Pairs) *Given profile X, sort the edges of G_X in descending order by weight, breaking ties lexicographically. Construct an initially empty set $S = \{\}$, and in order by weight, add each edge of G_X to S as long as it would not cause the edges of S to contain a cycle. In the directed acyclic graph defined by S, there is necessarily a single undominated candidate c with only edges to other candidates that it beats in pairwise competition. The rule returns a lottery that assigns all probability mass to c.*

B Preservation Under Embedding

An embedding may not preserve all of the information needed to compute the output of a function.

Definition 19 (Function Preservation) *A function $f : \mathcal{X} \to \mathcal{Y}$ is preserved by embedding $T : \mathcal{X} \to \mathcal{X}'$ if $\exists f' : \mathcal{X}' \to \mathcal{Y}$ such that $f'(T(X)) = f(X)$ for all profiles $X \in \mathcal{X}$.*

Proposition 1 says that for an embedding T to preserve a function f, there cannot be two profiles with the same embedding under T for which the function returns different outcomes.

Proposition 1 *Embedding T preserves function f if and only if $T(X) = T(\hat{X}) \Rightarrow f(X) = f(\hat{X})$ for all $X, \hat{X} \in \mathcal{X}$.*

Some embeddings preserve strictly more information than others. For instance, T_{WT} preserves all information necessary to compute T_T from a profile. This implies that if T_T preserves a function f, then T_{WT} must preserve f as well.

Proposition 2 *Suppose that for $T : \mathcal{X} \to \mathcal{X}'$ there exist $T_1 : \mathcal{X} \to \hat{\mathcal{X}}$ and $T_2 : \hat{\mathcal{X}} \to \mathcal{X}'$ such that $T(X) = T_2(T_1(X))$ for all $X \in \mathcal{X}$. Then for all $f \in \mathcal{F}$, T preserves f only if T_1 preserves f.*

Since we can compute T_T from T_{WT}, T_T can only preserve a function if T_{WT} does as well. The reverse does not hold, since T_{WT} may preserve functions that are not preserved by T_T. If an embedding preserves a function, then the function can be learned to arbitrary accuracy for a given number of candidates using that embedding.

The functions we are concerned with are PSCFs. Table 7 (green highlights) shows which PSCFs are preserved by these embeddings. See Appendix C for proofs of the negative results where PSCFs are not preserved. Each of these proofs consists of a counterexample with two profiles whose outcomes differ under the PSCF yet have the same embedding. Finally, many of our rules fall into Fishburn's categorization: the winner of C1 rules (Copeland) can be computed using the information encoded in T_T, while the winner of C2 rules (Borda, Schulze, Simpson-Kramer, Black's) can be computed from T_{WT}. A separate categorization of voting rules, positional scoring rules (Plurality, Borda), can be computed using only T_{RF} [11]. IRV has recently been shown *not* to belong to either C1 or C2, and is not a positional scoring rule [28].

C Rule Preservation Failure

Below we provide proofs for each case where we have noted that a PSCF cannot be computed using information found in a specific embedding. Each proof is described by two profiles, X_1 and X_2. For each embedding-PSCF pair (T, f) below, the profiles X_1 and X_2 have the same representation in the embedding but the PSCF produces a different lottery; $T(X_1) = T(X_2)$ but $f(X_1) \neq f(X_2)$. This proves that f is not preserved by T. For all proofs where the profiles contain 3 voters and 4 candidates, we have confirmed by brute force search that no smaller example exists with 3 voters and 3 candidates.

Table 6: PSCF preservation under embedding.

	T_{RF}	T_{WT}	T_T
Plurality	✓	×	×
Borda	✓	✓	×
Copeland	×	✓	✓
Schulze	×	✓	×
Simpson-Kramer	×	✓	×
IRV	×	×	×
Black's Rule	×	✓	×
Ranked Pairs	×	✓	×

C.1 Plurality

Plurality is a scoring rule, and therefore necessarily computable from a rank frequency embedding. It requires only one column of information from the rank frequency matrix, representing how often each

candidate is ranked first by a voter. By contrast, Plurality is not preserved by the weighted tournament embedding, and therefore not by the tournament embedding either.

Theorem 1 *The weighted tournament embedding does not preserve Plurality.*

Proof 1 *Let* $X_1 = (a \succ b \succ c), (b \succ a \succ c), (c \succ a \succ b)$ *and let* $X_2 = (a \succ b \succ c), (a \succ b \succ c), (c \succ b \succ a)$. *Observe that* $f^{Plur}(X_1) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ *because its plurality winner set is* $W(X_1) = \{a, b, c\}$, *while* $f^{Plur}(X_2) = (1, 0, 0)$ *because* $W(X_2) = \{a\}$ *has a unique plurality winner. But in both profiles, there are two voters who prefer* $a \succ b$, *two voters who prefer* $b \succ c$, *and two voters who prefer* $a \succ c$, *so the weighted tournament embeddings are the same* $T_{WT}(X_1) = T_{WT}(X_2)$.

Corollary 1 *The tournament embedding does not preserve Plurality.*

C.2 Borda

Theorem 2 *The tournament embedding does not preserve Borda.*

Proof 2 $X_1 = (a \succ b \succ c), (a \succ b \succ c), (b \succ c \succ a)$, $X_2 = (a \succ b \succ c), (a \succ b \succ c), (a \succ b \succ c)$.

C.3 Copeland

Copeland is the only probabilistic social choice function we consider that is preserved by the tournament embedding, and hence by the weighted tournament as well. However, Copeland is not preserved by the rank frequency embedding.

Theorem 3 *The rank frequency embedding does not preserve Copeland.*

Proof 3 $X_1 = (a \succ b \succ c \succ d), (b \succ c \succ d \succ a), (d \succ a \succ b \succ c)$, $X_2 = (a \succ b \succ c \succ d), (b \succ a \succ d \succ c), (d \succ c \succ b \succ a)$.

Schulze and Simpson-Kramer are weighted-tournament rules that are not preserved by the tournament or rank frequency embedding.

C.4 Schulze

Theorem 4 *The rank frequency embedding does not preserve Schulze.*

Proof 4 $X_1 = (a \succ b \succ c \succ d), (b \succ c \succ d \succ a), (d \succ a \succ b \succ c)$, $X_2 = (a \succ b \succ c \succ d), (b \succ a \succ d \succ c), (d \succ c \succ b \succ a)$.

Theorem 5 *The tournament embedding does not preserve Schulze.*

Proof 5 $X_1 = (a \succ b \succ c \succ d), (b \succ c \succ d \succ a), (d \succ a \succ c \succ b)$, $X_2 = (a \succ b \succ c \succ d), (b \succ c \succ d \succ a), (d \succ a \succ b \succ c)$.

C.5 Simpson-Kramer (Maximin)

Theorem 6 *The rank frequency embedding does not preserve Simpson-Kramer.*

Proof 6 $X_1 = (a \succ b \succ c \succ d), (b \succ c \succ d \succ a), (d \succ a \succ b \succ c)$, $X_2 = (a \succ b \succ c \succ d), (b \succ a \succ d \succ c), (d \succ c \succ b \succ a)$.

Theorem 7 *The tournament embedding does not preserve Simpson-Kramer.*

Proof 7 $X_1 = (a \succ b \succ c \succ d), (b \succ c \succ a \succ d), (d \succ c \succ a \succ b)$, $X_2 = (a \succ b \succ c \succ d), (b \succ c \succ a \succ d), (c \succ a \succ b \succ d)$.

C.6 IRV

Theorem 8 *The rank frequency embedding does not preserve IRV.*

Proof 8 $X_1 = (a \succ b \succ c \succ d), (b \succ c \succ d \succ a), (d \succ a \succ b \succ c)$, $X_2 = (a \succ b \succ c \succ d), (b \succ a \succ d \succ c), (d \succ c \succ b \succ a)$.

Theorem 9 *The tournament embedding does not preserve IRV.*

Proof 9 $X_1 = (a \succ b \succ c \succ d), (b \succ c \succ d \succ a), (d \succ a \succ c \succ b)$, $X_2 = (a \succ b \succ c \succ d), (b \succ c \succ d \succ a), (d \succ a \succ b \succ c)$.

Theorem 10 *The weighted tournament embedding does not preserve IRV.*

This follows as a corollary to Halpern et al. [28]. Thanks to Daniel Halpern for pointing this out!

C.7 Black's Rule

Theorem 11 *The rank frequency embedding does not preserve Black's Rule.*

Proof 11 $X_1 = (a \succ b \succ c \succ d), (b \succ c \succ d \succ a), (d \succ a \succ b \succ c)$, $X_2 = (a \succ b \succ c \succ d), (b \succ a \succ d \succ c), (d \succ c \succ b \succ a)$.

Theorem 12 *The tournament embedding does not preserve Black's Rule.*

Proof 12 $X_1 = (a \succ b \succ c \succ d), (b \succ c \succ a \succ d), (d \succ c \succ a \succ b)$, $X_2 = (a \succ b \succ c \succ d), (b \succ c \succ a \succ d), (c \succ a \succ b \succ d)$.

The weighted tournament embedding does preserve Black's Rule because it preserves enough information to determine the Condorcet-winner when there is one (e.g., preserves the Copeland Rule), and it preserves the Borda Rule.

C.8 Ranked Pairs

Theorem 13 *The tournament embedding does not preserve the Ranked Pairs Rule.*

Proof 13 $X_1 = (a \succ c \succ d \succ b), (c \succ d \succ b \succ a), (d \succ b \succ a \succ c)$, $X_2 = (a \succ c \succ b \succ d), (c \succ d \succ b \succ a), (d \succ b \succ a \succ c)$.

Theorem 14 *The rank frequency embedding does not preserve the Ranked Pairs Rule.*

Proof 14 $X_1 = (a \succ c \succ b), (b \succ a \succ c), (c \succ b \succ a)$, $X_2 = (a \succ b \succ c), (b \succ c \succ a), (c \succ a \succ b)$.

D Generalization

Our embeddings give us the ability to work with profiles with different numbers of voters. Although we trained our models on profiles with 44 voters, we can test with larger and smaller numbers of voters to assess generalization. The following experiments are performed on preliminary models which are not identical to the main results in this paper. The performance of these models is close to those in our final report. We test first on profiles with 199 votes (Table 8), and then again with only 13 voters (Table 9).

We see that there is a very mild increase in the loss across most rules and embeddings, though losses remain very similar to their value on profiles with 44 voters (see Table 7). In some cases there is no increase in loss: With the concatenated embedding, T_{CO}, both Copeland (with 13 voters) and Black's rule (199 voters) have the same loss. Loss even decreases when evaluating Black's rule on profiles with 13 voters and T_T. As well, on all but 4 rule-embedding pairs, loss is lower using profiles with 13 voters than those with 199 voters. These results highlight the importance of choosing embeddings that fit the rule, corresponding to the learning objective. As a result of our embeddings our learned rules are able to generalize extremely effectively to profiles of varying sizes without the need for additional training.

Table 7: L1 loss on the validation split of the training data (Impartial Culture, $(m = 11, n = 44)$) at the epoch with the lowest validation loss, for models targeting each rule using each embedding. Cells show mean $\pm$ standard error of the mean over 5 independently trained models. Shaded cells indicate the embedding contains sufficient information to learn the rule perfectly.

	T_{RF}	T_T	T_{WT}	T_{CO}
Plurality	0.003 ± 0.003	0.127 ± 0.000	0.126 ± 0.000	0.000 ± 0.000
Borda	0.036 ± 0.005	0.047 ± 0.000	0.022 ± 0.003	0.009 ± 0.000
Copeland	0.072 ± 0.003	0.000 ± 0.000	0.055 ± 0.002	0.000 ± 0.000
Schulze	0.066 ± 0.001	0.040 ± 0.000	0.061 ± 0.000	0.040 ± 0.000
Simpson-Kramer	0.066 ± 0.001	0.044 ± 0.002	0.063 ± 0.001	0.041 ± 0.000
IRV	0.109 ± 0.002	0.101 ± 0.002	0.105 ± 0.002	0.096 ± 0.002
Black's Rule	0.043 ± 0.003	0.043 ± 0.000	0.020 ± 0.003	0.022 ± 0.001
Ranked Pairs	0.089 ± 0.003	0.047 ± 0.004	0.078 ± 0.002	0.052 ± 0.005

Table 8: L1 loss on 512 Impartial Culture profiles with 11 candidates and 199 voters, for networks trained on each embedding with 11 candidates and 44 voters. Cells show mean $\pm$ standard error of the mean over 5 independently trained models.

Target Rule	T_{RF}	T_T	T_{WT}	T_{CO}
Plurality	0.041 ± 0.005	0.134 ± 0.000	0.135 ± 0.001	0.043 ± 0.005
Borda	0.039 ± 0.004	0.060 ± 0.001	0.025 ± 0.003	0.025 ± 0.001
Copeland	0.084 ± 0.004	0.000 ± 0.000	0.072 ± 0.001	0.000 ± 0.000
Schulze	0.075 ± 0.003	0.056 ± 0.001	0.069 ± 0.002	0.055 ± 0.001
Simpson-Kramer	0.071 ± 0.002	0.058 ± 0.001	0.070 ± 0.001	0.056 ± 0.001
IRV	0.118 ± 0.003	0.104 ± 0.003	0.111 ± 0.002	0.102 ± 0.002
Black's Rule	0.049 ± 0.003	0.049 ± 0.001	0.028 ± 0.003	0.037 ± 0.001
Ranked Pairs	0.097 ± 0.003	0.055 ± 0.004	0.082 ± 0.002	0.063 ± 0.006

Table 9: L1 loss on 512 Impartial Culture profiles with 11 candidates and 13 voters, for networks trained on each embedding with 11 candidates and 44 voters. Cells show mean $\pm$ standard error of the mean over 5 independently trained models.

Target Rule	T_{RF}	T_T	T_{WT}	T_{CO}
Plurality	0.004 ± 0.003	0.119 ± 0.001	0.117 ± 0.000	0.005 ± 0.001
Borda	0.037 ± 0.006	0.053 ± 0.001	0.025 ± 0.003	0.023 ± 0.000
Copeland	0.076 ± 0.002	0.000 ± 0.000	0.060 ± 0.001	0.000 ± 0.000
Schulze	0.068 ± 0.001	0.048 ± 0.001	0.064 ± 0.001	0.048 ± 0.000
Simpson-Kramer	0.067 ± 0.001	0.051 ± 0.002	0.064 ± 0.001	0.048 ± 0.001
IRV	0.109 ± 0.002	0.103 ± 0.003	0.110 ± 0.002	0.094 ± 0.002
Black's Rule	0.049 ± 0.003	0.040 ± 0.001	0.031 ± 0.003	0.028 ± 0.001
Ranked Pairs	0.093 ± 0.005	0.046 ± 0.004	0.077 ± 0.002	0.051 ± 0.006

Table 10: Test loss on 10,000 profiles ($m = 11, n = 44$) from each preference distribution, for networks trained using Impartial Culture preferences and the T_{CO} embedding. Cells show mean $\pm$ standard error of the mean over 5 independently trained models.

Target Rule	IC	IAC	Urn	Mall.	SP	PrefLib
Plurality	.000 $\pm$ .000	.000 $\pm$ .000	.005 $\pm$ .000	.002 $\pm$ .000	.006 $\pm$ .001	.003 $\pm$ .000
Borda	.005 $\pm$ .000	.004 $\pm$ .000	.078 $\pm$ .001	.095 $\pm$ .001	.123 $\pm$ .025	.112 $\pm$ .001
Copeland	.000 $\pm$ .000	.000 $\pm$ .000	.000 $\pm$ .000	.000 $\pm$ .000	.000 $\pm$ .000	.000 $\pm$ .000
Schulze	.017 $\pm$ .000	.017 $\pm$ .000	.024 $\pm$ .001	.023 $\pm$ .001	.026 $\pm$ .011	.032 $\pm$ .001
SK	.017 $\pm$ .000	.016 $\pm$ .000	.024 $\pm$ .001	.022 $\pm$ .001	.040 $\pm$ .020	.032 $\pm$ .001
IRV	.088 $\pm$ .003	.088 $\pm$ .003	.073 $\pm$ .005	.063 $\pm$ .006	.088 $\pm$ .007	.065 $\pm$ .006
Black's	.014 $\pm$ .000	.015 $\pm$ .000	.079 $\pm$ .001	.093 $\pm$ .001	.080 $\pm$ .023	.110 $\pm$ .002
RP	.036 $\pm$ .008	.036 $\pm$ .008	.032 $\pm$ .008	.026 $\pm$ .009	.006 $\pm$ .001	.030 $\pm$ .008

D.1 Beyond Impartial Culture

Recent [9, 8, 56] and older work [37, 38] in the COMSOC community has illustrated the need to test on a wide variety of synthetic and real world preference distributions to ensure generalization. We test the empirical merit of this fact by evaluating our networks trained on IC preferences on test sets sampled from a wide range of distributions, including real-world preference data. In Section 4 we show how well rules trained on Impartial Culture preferences generalize to other preference distributions. The main body shows the accuracy of each rule and Table 10 shows the test loss for each model across distributions. Specifically, we consider the following distributions:

Impartial Culture (IC) Each unique preference order is equally likely, regardless of which orders any other voters have selected [27].

Impartial Anonymous Culture (IAC) Preferences profiles are generated collectively, rather than as individual preference orders. Each multi-set of preference orders (i.e. a preference profile) is equally likely to be generated, making voter identities irrelevant [25].

Mallows Preference orders are noisy estimates of some reference ranking r, with the amount of noise related to a parameter ϕ [36]. A value of $\phi = 0$ results in all voters have identical preferences while $\phi = 1$ results in the Impartial Culture distribution. For each profile we sample $\phi \in [0, 1]$ uniformly at random using the Mallow's distribution described by Boehmer et al. [7].

Urn All $m!$ preference orders exist in an "urn." Each voter decides their ranking by sampling a ranking from the urn. Once a ranking is selected, $\alpha m!$ copies of it are added to the urn [20]. For each profile we sample α from a Gamma distribution with shape parameter $k = 0.8$ and scale parameter $\theta = 1$ as described by Boehmer et al. [7].

Single-Peaked There is some global ordering of alternatives. Each voter has some favourite alternative and prefers all alternatives closer to their favourite over those further away. We sample single-peaked profiles from Walsh's distribution [57].

PrefLib An online repository containing profiles corresponding to real human preferences expressed across many domains [37]. We use all complete profiles with strict orders and $m \geq 11$ candidates. For profiles with greater than 11 candidates we form a profile by selecting a subset of candidates uniformly at random. This results in a test set of 6945 profiles.

E Other Tested Neural Network Setups

We note that in all cases, our results are very similar between runs using the same parameters. Across different parameter settings our results are broadly similar – rules that are learned well are learned well and rules learned less well have worse outcomes – in all settings.

Before we settled on our final experiment configuration, we conducted several tests to explore the efficiency of learning and target larger profile dimensionalities without making significant changes to our models or embeddings. Experimenting with our model's representational threshold, we found

that with higher dimensionalities from our initial baseline of 7 candidates and 29 voters we were able to achieve faster and more consistent convergence with L1Loss and TanH activation.

With 15 candidates and 44 voters, we observe training loss decrease much more consistently than our ReLU baseline with lower variance and better generalization. Similarly, we find that utilization of a distribution-based loss like KL-Divergence or Jensen-Shannon divergence is inefficient when targeting a sparse output vector, and that even with 7 candidates these approaches see a severe reduction in the generalizability of our models.

While we did want to focus on smaller model architectures to more directly rely on our embeddings, we did experiment with more complex models. We tested two alternate networks which we observed to have below-satisfactory performance when compared to our base feed-forward network. The first was a simple network with a depth of 6 and a constant layer width of $m^2 \times 3$, and the second had the same depth, but a funnel architecture which decomposed the width of each hidden layer from the input to the output layer to minimize information dropoff. Both networks underperformed when compared to our baseline.

F Learning Axioms Directly

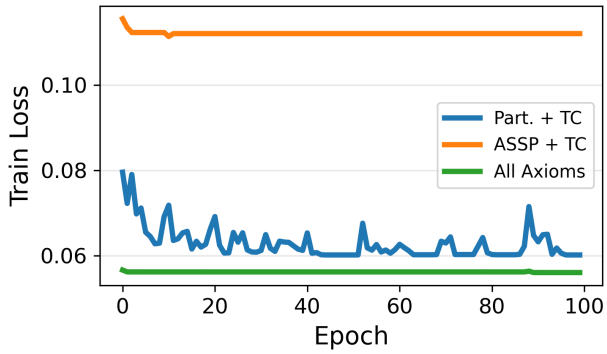

Figure 5: Training loss of each set of axioms on networks trained exclusively for axioms loss.

Here we provide results and additional discussion around our experiments training networks purely focused on learning axioms. We fix three subsets of axioms: (1) Strategyproofness and Top Cycle [13], (2) Participation and Top Cycle [44], and (3) all four axioms. Each of these axiom represents a known impossibility. Using the same architecture as our other experiments we train a network targeting a loss function equal to the unweighted sum of each axiom's individual loss function.

Figure 5 shows the training loss of each network over 100 epochs of training. Final axiom loss values for each axiom are shown in Table 5. We observe that loss only decreases very little and expect this is caused by a highly discrete loss landscape – a function must change significantly in order to reduce loss. Similar experiments (not shown) with a higher learning rate did not fundamentally change this result. While the axioms are, in each case, mutually incompatible and reaching a loss of zero is impossible we suspect different learning techniques may be more well suited for this task and provide superior results.

A particularly striking aspect of Table 5 is that Participation and Monotonicity always have approximately 0 loss. Participation's lack of loss can be explained by the fact that an untrained randomized network will never violate the Participation axiom; while we do not prove it we expect that learning to minimize ASSP loss also adds pressure to reduce Participation loss (otherwise we would expect ASSP + TC to increase Participation loss). Monotonicity appears a relatively "easy" axiom to avoid violating and is closely tied to ASSP. We also note that the high loss found for Top Cycle is conceptually similar to the result of Caiata et al. [15] (under a very different model) who also found learning rules which avoid violating the Condorcet property to be challenging.

G Complete Axiom Finetuning Results

Tables 11 to 22 report axiom loss, rule loss, and axiom violation rate for each rule and each type of rule (actual rule, base model trained for rule loss, finetuned model). All trained models used T_{CO} features. An axiom is said to have been violated on an example if the axiom loss on the example is greater than $1e^{-5}$. We report results for a single training run only. Our other results show that the variability between individual training runs is quite small.

Table 11: Axiom Loss of all trained models and actual rules on Monotonicity axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.083	0.096	0.076	0.072	0.073	0.077	0.094	0.071
Finetuned	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rule	0.000	0.000	0.000	0.000	0.000	0.003	0.000	0.001

Table 12: Axiom Violation Rate of all trained models and actual rules on Monotonicity axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.842	0.822	0.742	0.866	0.839	0.624	0.808	0.597
Finetuned	0.029	0.003	0.010	0.016	0.009	0.000	0.001	0.000
Rule	0.000	0.000	0.000	0.000	0.000	0.166	0.000	0.026

Table 13: Rule Loss of all trained models and actual rules on Monotonicity axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.000	0.013	0.000	0.054	0.056	0.119	0.026	0.069
Finetuned	0.004	0.027	0.004	0.057	0.059	0.121	0.031	0.072
Rule	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 14: Axiom Loss of all trained models and actual rules on Participation axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.446	0.422	0.291	0.291	0.288	0.293	0.396	0.280
Finetuned	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rule	0.000	0.000	0.060	0.019	0.019	0.090	0.047	0.059

Table 15: Axiom Violation Rate of all trained models and actual rules on Participation axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.944	0.890	0.783	0.911	0.879	0.656	0.880	0.641
Finetuned	0.052	0.001	0.003	0.006	0.014	0.000	0.000	0.000
Rule	0.000	0.000	0.405	0.257	0.271	0.330	0.100	0.215

Table 16: Rule Loss of all trained models and actual rules on Participation axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.000	0.013	0.000	0.054	0.056	0.119	0.026	0.069
Finetuned	0.017	0.023	0.003	0.058	0.059	0.123	0.029	0.073
Rule	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 17: Axiom Loss of all trained models and actual rules on Top Cycle axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.155	0.015	0.000	0.001	0.001	0.069	0.009	0.032
Finetuned	0.017	0.012	0.000	0.001	0.001	0.070	0.007	0.035
Rule	0.155	0.017	0.000	0.000	0.000	0.063	0.001	0.000

Table 18: Axiom Violation Rate of all trained models and actual rules on Top Cycle axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.210	0.017	0.014	0.009	0.006	0.069	0.011	0.032
Finetuned	0.060	0.013	0.050	0.004	0.012	0.070	0.008	0.035
Rule	0.192	0.018	0.000	0.000	0.000	0.063	0.001	0.000

Table 19: Rule Loss of all trained models and actual rules on Top Cycle axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.000	0.013	0.000	0.054	0.056	0.119	0.026	0.069
Finetuned	0.115	0.039	0.018	0.060	0.063	0.120	0.038	0.073
Rule	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 20: Axiom Loss of all trained models and actual rules on ASSP axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.446	0.440	0.348	0.363	0.377	0.343	0.409	0.369
Finetuned	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rule	0.036	0.015	0.016	0.015	0.016	0.064	0.027	0.025

Table 21: Axiom Violation Rate of all trained models and actual rules on ASSP axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.964	0.828	0.825	0.914	0.912	0.678	0.805	0.744
Finetuned	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Rule	0.678	0.131	0.404	0.512	0.516	0.565	0.169	0.329

Table 22: Rule Loss of all trained models and actual rules on ASSP axiom.

	Plurality	Borda	Copeland	Schulze	SK	IRV	Black's	RP
Base Model	0.000	0.013	0.000	0.054	0.056	0.119	0.026	0.069
Finetuned	0.165	0.165	0.166	0.165	0.165	0.165	0.165	0.164
Rule	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes. Results in Section 4 and 5 discuss these claims.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes. We discuss impossibility theorems and different possible outcomes of axiom fine-tuning in Section 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Proofs of theoretical claims are found in appendices.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Yes. We discuss parameter details in the text and provide all code to reproduce our data and results.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We provide our code and instructions on its use.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Yes. We discuss this in Sections 4 and 5. Our code includes additional details.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report five runs with standard error on most of our experiments. For axiom-related work, we report one run.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Sections 4 and 5.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the potential use of this method. Misuse appears unlikely.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: We do not anticipate a clear method of misusing this work.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Yes. We create our own code and use existing libraries as noted in the text and in our code.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: Our code is documented.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: We do not do human experiments.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: We do not do human experiments.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are used only for very minor portions of the codebase and not for any part of the writing.

Guidelines:

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