利用对抗式图神经网络学习弹性选举

Hao Xiang Li[1*] **Yash Shah**[1*] **Lorenzo Giusti**[2]

[1]剑桥大学计算机科学系，剑桥，英国[2]欧洲核子研究组织，日内瓦，瑞士

hxl23@cst.cam.ac.uk ys562@cam.ac.uk lorenzo.giusti@cern.ch

摘要

面对不利动机，实现共识至关重要。自17世纪以来，选举一直是现代民主运作的经典方式。如今，选举规范着市场，为现代推荐系统或点对点网络提供驱动力，并且仍是体现民主的主要途径。然而，设计一种能满足所有假设情形的理想普适投票规则仍然颇具挑战，而此类系统的设计也处于机制设计研究的前沿。自动化机制设计是一种颇具前景的方法，近期研究表明，集合不变架构尤其适合对选举制度建模。然而，诸多顾虑阻碍了其在现实场景中的直接应用，例如对策略性投票的鲁棒性。在本文中，**我们拓展了学习所得投票规则的表达能力，并将神经网络架构的改进与对抗训练相结合，在最大化社会福利的同时提升投票规则的韧性**。我们在合成数据集和真实世界数据集上评估了这些方法的有效性。我们的方法通过使用二部图表示选举，并利用图神经网络学习相应的投票规则，解决了以往投票规则学习研究中的关键局限。我们相信，这为机器学习应用于现实世界的选举开辟了新的前沿。

1 引言

尽管选举最常用于解决政治分歧，但也可以将其抽象为一种结构化工具，用于解决争端并在日常情境中做出集体决策 [1]。选举让群体能够**汇总偏好**，使每位利益相关者都有发声的机会，并**确保公平性**，让每位参与者享有平等的发言权。选举还有助于通过和平方式解决问题来**避免冲突**。最后，选举能够**使结果具有正当性**，因为当每个人都有机会参与时，参与者更愿意接受结果 [2]。

当一群具有各自偏好的选民需要从一组可能的结果中选出一位候选人时，选举便成为汇总这些偏好、形成反映群体意愿的集体决策的一种机制；其应用不止于政治体系，还包括多智能体机器人学、去中心化自主智能体和推荐系统 [3]。在选举中，投票机制是指一种算法过程，用于 eliciting 个体偏好，并依据特定标准选出结果 [4]。理想的机制应具备社会期望特性：在两两比较中，被选出的候选人应优于其他候选人 [5]；理性的投票者会如实表达投票意向，或者至少，诚实表达意见不应主动损害其偏好候选人当选的机会 [6]；这一过程必须公平，且每位投票者和每位候选人都应获得平等的代表权。

[*]平等贡献。

预印本。

遗憾的是，要在一组看似合理的标准下，选出一条始终能产生社会最优结果的确定性规则，往往仍是一个悬而未决的问题；更糟的是，有时可以证明这样的规则根本不可能存在 [7], [8]。研究者已投入大量精力寻找更完善的投票机制，由此形成了一个名为投票理论的完整研究领域，其中的研究成果探讨了公平性标准 [8], 、策略性操纵 [9], 以及计算复杂性 [7]。

在这项工作中，我们研究了学习尚未被发现的投票规则的方法。我们的主要贡献是一种福利最大化学习方法，该方法满足选民匿名性、候选人中立性以及其他可选的各种标准，能够推广到任意数量的候选人，并且对策略性投票具有鲁棒性。为此，我们提出：（a）由以下部分组成的置换等变神经网络组合：一个**图投票网络**（GEVN）和一个**图策略网络**（GESN）；（b）社会福利损失和单调性损失的算法设计；以及（c）对抗性评估，以应对策略性投票。

2 背景与相关工作

图神经网络： 设 $G = (\mathcal{V}, \mathcal{E})$ 为一个图，其中 $\mathcal{N}_v = \{u \in \mathcal{V} : (u, v) \in \mathcal{E}\}$ 是节点 v的一跳邻域，具有邻域特征 $\boldsymbol{X}_{\mathcal{N}_v} = \{\{\boldsymbol{x}_u : u \in \mathcal{N}_v\}\}$，其中 $\{\{\cdot\}\}$ 表示一个多重集。我们将消息传递函数定义为关于 $\boldsymbol{X}_{\mathcal{N}_v}$的置换不变函数，其形式如下：

$$f\left(\boldsymbol{x}_v, \boldsymbol{X}_{\mathcal{N}_v}\right) = \phi\left(\boldsymbol{x}_v, \bigoplus_{u \in \mathcal{N}_v} \psi(\boldsymbol{x}_v, \boldsymbol{x}_u)\right) \tag{1}$$

其中，ψ 和ϕ 分别是可学习的消息函数和更新函数，而$\oplus$ 是一个置换不变的聚合函数（例如求和、求均值、求最大值）。消息传递神经网络（MPNN）由依次应用的消息传递层组成，其形式如式（1）所示。

投票理论： 设有一组 n 选民 $V = \{v_1, v_2, ..., v_n\}$ 和 m 候选人 $\{c_1, c_2, ..., c_m\}$ $n \times m$ $\boldsymbol{U} \in \mathbb{R}^{n \times m}$ $C =$ 。效用概况矩阵用于指定选民偏好的真实情况，其中效用u_{ij} 表示选民v_i 对候选人c_j的偏好程度。社会福利函数$\mathrm{sw}_{\boldsymbol{U}}(c \in C) : C \to \mathbb{R}$ 表示候选人c 在效用概况$\boldsymbol{U}$下的可取性。例如，功利主义福利 [10] 对应于求和：$\mathrm{sw}_{\boldsymbol{U}}^{\mathrm{util}}(c_j) = \sum_{i \le n} \boldsymbol{U}_{ij}$. 纳什福利[11] 对应于求积：$\mathrm{sw}_{\boldsymbol{U}}^{\mathrm{nash}}(c_j) = \prod_{i \le n} \boldsymbol{U}_{ij}$，而罗尔斯式福利[12] 对应于取最小值：$\mathrm{sw}_{\boldsymbol{U}}^{\mathrm{rawl}}(c_j) = \min_{i \le n} \boldsymbol{U}_{ij}$.

机制设计： 投票机制是一个函数，将所有选民给出的偏好映射为一名候选人。概率社会选择函数（PSCF）将投票机制的定义扩展为输出候选人的概率分布，而非单一候选人 [13]。社会福利函数代表最大化目标，由系统的领域专家选定。一般来说，投票规则可根据所收集信息的类型进行分类。一方面，对于排序 $R = \{1, ..., m\}$，排序投票规则要求每位投票者提供候选人到排序的双射：$\sigma_i^{\mathrm{rank}} : C \to R$ 表示偏好的全序。另一方面，基数投票规则接收选民对每位候选人给出的等级分数：$\sigma_i^{\mathrm{cardinal}} : C \to \mathbb{R}$ [4], [14], [15]，从而允许选民表达其支持力度。

我们的工作主要关注基数投票规则的学习，且所有描述的方法都可通过将归一化博尔达计数 作为投票者得分，改编为排序投票（C节）算法。对于排序位置为 r 的候选人，其博尔达计分为 $1 - \frac{r}{m}$。我们还关注 PSCF：它是通过神经网络进行学习的自然选择，因为它能够产生可微损失，并通过避免平局简化设计 [13]。此外，对候选人概率取 argmax，即可将 PSCF 转换为确定性投票机制。

相关工作：除最大化选定的社会福利函数外，投票理论文献还提供了一组直观的准则，用于评估投票机制 [17]。例如，孔多塞准则总是选出两两胜利次数最多的候选人；单调性准则则体现了这样一种理念：增加对某位候选人的支持应当对该候选人有利。众所周知，单记可转移投票机制不满足这一要求 [7]。

匿名性准则指出，投票机制不会歧视参与者（即匿名投票机制是集合输入函数）。Anil 和 Bao [2] 证明，置换不变网络（PIN）是集合输入函数的通用逼近器。因此，DeepSets [18] 和 Set Transformers [19] 适合用于学习具有匿名性保证的投票机制，并且取得了出色的实证结果。然而，PIN 无法泛化到数量未见过的候选人。一个类似但不同的准则是中立性准则：投票机制不应偏袒某位特定候选人。中立的投票机制应关于候选人具有置换等变性；PIN 并不能保证这一性质。

遗憾的是，吉巴德定理 [8] 和希兰德定理 [20] 指出，所有满足理想标准的合理投票机制都可能被投票者操纵。通俗地说，对于任何投票机制，以下情况至少有一种成立：(a) 它是独裁规则，由单个投票者决定结果；(b) 只有两名候选人；(c) 该机制不是策略无关的。值得注意的是，对于任何投票者而言，都不存在占优策略。因此，投票者的最优选票取决于系统中的其他投票者。

聚焦于第三种情况，作为投票机制设计者，我们需要研究策略性投票；不能指望投票者直接报告其效用概况，而必须预期他们会提交“经过操纵的”选票，这些选票会以社会成本为代价带来个人收益。例如，在一个针对罗尔斯式福利进行优化的基数制中，投票者可能会为自己最喜欢的候选人报告最大效用，而为所有其他候选人报告 0，从而确保只有其最喜欢的候选人当选。请注意，投票机制的可操纵性取决于投票者拥有的信息量 [21]。

自动化机制设计是学习现有及新型投票机制的一个有前景的研究方向。Procaccia 等人 [22] 证明，传统投票规则能够通过神经网络实现可能近似正确可学习（PAC），为机器学习投票机制奠定了可靠基础。Firebanks [23] 尝试通过软约束学习改良后的投票机制。Matone 等人 [13] 使用偏好剖面的置换不变嵌入和多层感知机架构，学习满足参与准则 [6]的投票规则。Holliday 等人 [21] 评估了传统投票规则对操纵的鲁棒性。近期，Anil 和 Bao [2] 评估了置换不变架构，将其作为较简单的多层感知机（MLP）[24] 模型的替代方案，从而确保选民匿名性。他们的方法能够有效推广到不同的投票者人数和效用分布，并可通过行为克隆直接学习福利最大化投票机制。他们表明，即使假设信息有限，神经网络也能有效操纵现有投票规则。然而，在现实场景中，候选人数可能会动态变化，而且部分投票者可能表现出对抗行为。现有方法无法适应这些情况。我们的贡献是通过引入更高效的架构和福利最大化损失函数，弥合机器学习与现实应用之间的差距。

3 通过图神经网络学习有韧性的投票规则

我们关注如何学习能够优化社会福利的有韧性机制。具体而言，该机制应当（1）预期选民群体会进行策略性投票；（2）受理想设计标准的约束。我们考虑匿名性和中立性（公平性），并提出如何施加孔多塞准则和单调性准则的约束。此外，与以往研究相比，我们提出的方法能够推广到数量不受限的候选人，这是一个重大进展。图 1（左）展示了系统概览。

选举二部图（EBG）：鉴于 $n = |V|, m = |C|$，需要考虑 $n \times m$ 种选民—候选人映射。如果我们直接将展平后的向量输入多层感知机，网络就会受限于固定数量的投票者和候选人，也无法保证匿名性或中立性。为部分解决这一问题（即匿名性以及对投票者数量的泛化），以往研究采用了选民置换不变嵌入 [23]。Anil 和 Bao 可以视为使用置换不变网络架构学习嵌入 [2]。Matone 和 Firebanks [13] 则使用基于社会选择理论文献手动构造的嵌入，再输入多层感知机。然而，这些方法无法兼顾中立性以及对候选人数量的泛化。

用于学习 PSCF 的神经网络应将选民偏好档案映射为候选人上的概率分布。因此，选择一种适合输入神经网络的选民偏好档案表示方式至关重要。

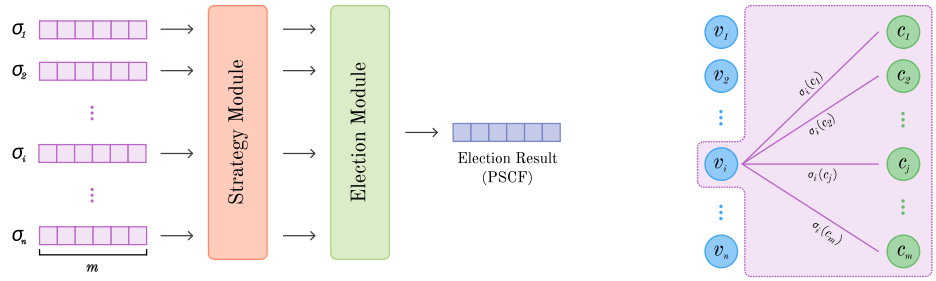


图 1：左侧：系统概览。模型的输入是每位投票者的偏好剖面。策略模块随后将其转换为策略性偏好剖面。最后，选举模块对其进行处理，得出结果（PSCF）。箭头表示数据流。右侧：将 $n \times m$ 偏好剖面表示为二部图的示意图，其中投票者和候选人是节点，偏好分数是边。请注意，图中仅显示了投票者 i 的边。

受 Pointer-Networks [25], 的启发，我们将选举表示为带属性的无向图，并将先前的研究推广到任意数量的候选人。因此，我们提出将投票者的偏好转换为选举二部图（EBG），即由彼此不相交的投票者节点 V 和候选人节点 C组成的集合。每个节点都采用独热编码，以区分投票者和候选人。每条边都双向连接某个投票者 v_i 与某个候选人 c_j，表示偏好 $\sigma_i(c_j)$——边 (v_i, c_j) 的一维特征为 $x_{v_i c_j} = \sigma_i(c_j)$。由于偏好剖面完整定义了所有选民-候选人对上的偏好，EBG 是一个完全二部图（如图 1（右）所示）。

选举模块：为了保证匿名性和中立性，只需学习以 EBG 为输入的置换等变函数。评估获胜者概率可转化为针对候选节点的节点级分类任务，同时对投票者节点进行掩码操作。

命题 3.1。对 EBG 上的置换等变函数进行投票者节点掩码处理后，即可转化为满足匿名性和中立性的投票机制。

我们在 A.1 节中给出了上述命题的证明。

图神经网络（GNN）[26] 是学习图域上置换等变函数的自然选择。我们使用一种定制的图网络（GN）[27], ，它是消息传递神经网络（MPNN）的扩展，能够同时考虑边特征和节点特征，并将我们的模型称为图选举投票网络（GEVN）。

考虑一个图 $G = (\mathcal{V}, \mathcal{E})$，其节点为 $\mathcal{V}$，边为 $\mathcal{E}$。定义节点特征 x_i 和边特征 x_{ij}，以及潜在特征 h_i^l、h_{ij}^l，其中 l 表示层，初始层特征为 $h_i^0 = x_i$、$h_{ij}^0 = x_{ij}$。网络的每一层都按如下方式更新潜在特征：

$$
\begin{aligned}
h_{ij}^{l+1} &= \phi_e\left(h_i^l, h_j^l, h_{ij}^l\right) \\
h_i^{l+1} &= \phi_v\left(h_i^l, \bigoplus_{j \in \mathcal{N}_i} \psi\left(h_i^l, h_j^l, h_{i,j}\right)\right)
\end{aligned}
\tag{2}
$$

ϕ_e、ϕ_v、ψ 分别表示边操作、节点操作和消息传递操作。我们使用多层感知机作为构建模块，定义 GEVN 层中的参数化操作。经过若干这样的层后，我们对节点应用候选人掩码，并使用 Softmax 函数将标量值转换为概率分布；因此，GEVN 可以解释为一种 PSCF。

对于任何提出的架构而言，具备足够强的能力来学习任意投票机制都是理想的。基于 Xu 等人 [28] 和 Chen 等人 [29], 关于图同构测试与通用函数逼近之间联系的研究，我们在 A.2 节中证明了 GEVN 具有足够的表达能力。

定理 3.2。 当应用于 EBG 时，如果图神经网络将 Weisfeiler-Lehman 测试判定为不同构的任意两个图映射为不同的嵌入，则该图神经网络将成为通用逼近器。

以往关于训练[2], [13], [23] 的工作主要通过行为克隆学习现有投票规则；PSCF 通过最小化负对数似然（NLL）损失来训练，以对正确的候选人进行分类。由于我们关注的是学习高效的投票机制，因此可以直接针对社会福利函数进行优化。PSCF p 在候选人上的期望社会福利为：

$$\mathbb{E}[\mathrm{sw}_{\boldsymbol{U}}(p)] = \sum_j p(c_j)\mathrm{sw}_{\boldsymbol{U}}(c_j) \tag{3}$$

我们将福利损失定义为

$$L_{\mathrm{sw}_{\boldsymbol{U}}} = -\mathbb{E}[\mathrm{sw}_{\boldsymbol{U}}(p)] \tag{4}$$

最大化期望社会福利等价于最小化福利损失。

满足投票准则： 可根据需要对 GEVN 模块施加约束，使其满足各种投票准则。例如，将 GEVN 输出的支持集截断为史密斯集合 [30]，即可保证满足孔多塞准则。此外，还可以通过训练损失在实际中体现其他准则。例如，单调性准则可以表示为一阶约束 $\forall v_i, c_j.\ \frac{\partial p(c_j)}{\partial(\sigma_i(c_j))} \geq 0$，该约束规定，候选人的分配~~概率随~~对该候选人表达的偏好增强而非递减。设单调性损失为：

$$L_{\mathrm{mono}} = -\sum_{i,j} \mathbb{1}_{\{\partial p(c_j)/\partial(\sigma_i(c_j))<0\}} \partial p(c_j)/\partial(\sigma_i(c_j)) \tag{5}$$

通过在计算图上进行两次反向传播即可实现这一点。

策略模块： 我们将策略性投票定义为这样一种情形：投票者的偏好档案并未真实反映其内在效用。在现实世界中，这种行为十分常见：投票者会进行策略性投票，选择自己并非最偏好的候选人（即不能使该投票者效用最大化的候选人），而是选择在效用权衡下有合理胜选概率的候选人 [31]。理想的投票机制应当即使在存在策略性投票的情况下，也能选出符合社会利益的候选人。

为了学习鲁棒投票机制，我们提出使用图选举策略网络（GESN）对策略性投票者的行为建模，并以对抗方式联合训练 GEVN 和 GESN（见图 2）。

GESN 将效用概况 $\boldsymbol{U}$ 映射为每位投票者的一组偏好剖面。网络架构的选择可用于约束策略性投票者可获取的信息，从掌握所有真实效用的全局知识，到仅掌握个人效用，构成一个连续范围。我们方法的核心是，GESN 通过最小化每位投票者的理性损失进行训练。理性投票者 v_i 旨在通过控制其提交的偏好档案这一行动来优化自身的个人效用：

$$\max_{\sigma_i} \sum_{j=1\ldots m} p(j|\sigma_i)\boldsymbol{U}_{ij} \tag{6}$$

$$L^i_{\mathrm{rational}} = -\sum_{j=1\ldots m} p(j|\sigma_i)\boldsymbol{U}_{ij} \tag{7}$$

为了对行动空间施加必要限制，在对 GESN 应用梯度下降时，会冻结 GEVN 的权重。此外，在对 L^i_{rational}进行反向传播时，我们会截断与 v_i无关的行动所对应的梯度。这种理性损失使 GESN 能够学会以最大化每位投票者个人效用的方式修改输入偏好档案。GESN 的输出经过归一化，将偏好档案空间限制在一个紧空间内，并体现一人一票意义上的公平性。例如，当 $a,\ b \in \mathbb{R}$ 时，我们将选票总数限制为 a （投票预算），或将选票限制在 $[a, b]$ 的范围内。

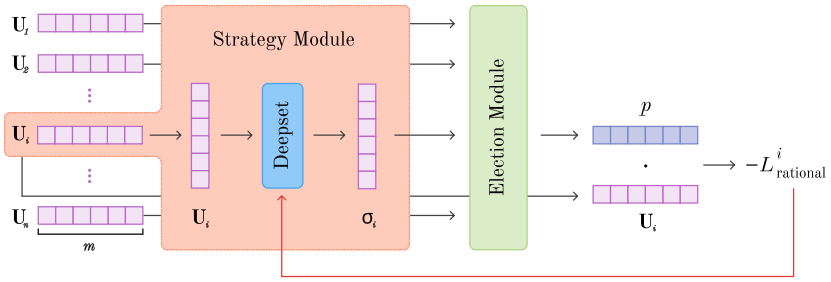


图 2：该图展示了如何使用理性损失训练策略模块。在此示例中，我们选择投票者 v_i 作为目标投票者，并针对其优化 GESN，同时阻止梯度向其他投票者传播。箭头表示数据流。该特定架构假设私人效用信息以及投票者之间不存在通信链路，但我们的通用框架并不受这些假设的限制。

GEVN 经过训练后会优化 L_{sw_U} ，并可选择优化 L_{mono} ；策略模块则优化 $\sum_i L^i_{\text{rational}}$。由于 GEVN 与 GESN 同时以端到端方式进行训练，因此它能够学会适应所实施的投票操纵。

4 实验

本节将通过实证验证我们模型的潜在优势。具体而言，我们关注以下问题。(1) GEVN 是否足够强大，能够学习投票机制，并能否推广到投票者／候选人数量可变的情况？(2) GEVN 能否成功学习福利最大化投票机制？(3) 在我们的对抗训练过程中学习到的投票机制能否抵御策略性投票者？

数据集：我们使用 狄利克雷分布（$\alpha = 1$ ）和空间模型（投票者和候选人均匀分布在单位立方体中，效用为 $\boldsymbol{U}_{ij} = 1 - \|v_i - c_j\|_2$）采样无偏合成偏好档案 [32] 。我们还使用真实世界数据集检验我们的方法：*MovieLens*数据集 [33] ，以及 2017 年法国选举期间在*Grenoble*市开展的一项实验 [34]。根据需要，我们从选举中均匀抽样，选取所需数量的候选人和投票者。

训练设置：我们使用 Adam 优化器 [35] 和带热重启的余弦退火学习率调度 [36] ，通过梯度下降训练各模块。完整的架构和实验细节将在 E 节中说明。

学习经典投票规则：为了验证我们的选举模块的表达能力，我们手动计算常见的投票规则（相对多数制、博尔达计数法、科普兰计数法、最大最小值法和单记可让渡投票制（STV）。详见 B 节。），然后使用负对数似然损失训练我们的模型，使其模拟这些经典方法。我们生成训练数据时，将选民数量设定在 3 至 50 的范围内，将候选人数量设定在 2 至 10 的范围内。此外，我们使用包含 75 名选民和 15 名候选人的验证数据集，在一次训练过程中选择最佳模型，然后在包含 100 名选民和 20 名候选人的测试集上进行评估。

作为比较，我们还实现了一个 DeepSets 选举模型，其输入采用博尔达分数或独热编码，如 [2] 中所述，以反映候选人排序。在先前的工作中，[2], DeepSets 能够达到最先进水平（SOTA）的性能。由于 DeepSets 无法外推到更多候选人的情况，我们将验证集中的候选人数量调整为 10，并省略测试集。

如表1所示，在模仿经典投票规则的任务中，我们提出的 GEVN 达到了最先进性能。具体而言，GEVN 对基于评分的经典规则建模效果尤为出色，在测试集上对相对多数制的准确率高达 0.92，对博尔达计数法则达到完美准确率。尽管面对的是具有分布外候选人数的难度严格更高任务，GEVN 在验证集上的泛化准确率仍远超 DeepSet。这种泛化准确率也体现在难度高得多的测试集以及真实世界数据集上（详见 D.1节）。我们推测，性能提升和泛化能力得益于中立性

表1：利用机器学习学习经典投票规则。展示了预期准确率及其在 10 次重复实验中的 95% 置信区间。小型模型的参数预算定义为 100, 000，否则参数预算为 1, 000, 000。GEVN 的表现显著优于 DeepSet模型，达到了最先进性能。省略的置信区间是指其95% 区间低于0.01。

		GEVN	GEVN（小）	DeepS.	DeepS.（小）	DeepS. 独热编码	DeepS. 独热编码（小）
相对多数制	验证	0.99 ± 0.01	**1.00**	0.48 ± 0.05	0.25 ± 0.04	0.83	0.80 ± 0.01
	Test	**0.92** ± 0.04	0.91 ± 0.03	[-]	[-]	[-]	[-]
博尔达计数法	验证	**1.00**	**1.00**	0.41 ± 0.05	0.38 ± 0.05	0.46 ± 0.01	0.44 ± 0.01
	Test	**1.00**	**1.00**	[-]	[-]	[-]	[-]
科普兰计数法	验证	**0.76**	**0.76**	0.37 ± 0.04	0.41 ± 0.05	0.45 ± 0.01	0.43 ± 0.02
	Test	**0.78**	**0.78**	[-]	[-]	[-]	[-]
极小极大	验证	**0.73**	**0.73**	0.40 ± 0.04	0.45 ± 0.04	0.43 ± 0.01	0.40 ± 0.01
	Test	**0.70**	**0.70**	[-]	[-]	[-]	[-]
STV	验证	**0.53**	**0.53**	0.34 ± 0.04	0.35 ± 0.04	0.42 ± 0.01	0.40 ± 0.01
	Test	**0.49**	**0.49**	[-]	[-]	[-]	[-]

由置换等变图神经网络强制保证：所有纳入考察的经典规则都具有中立性，因此无需消耗模型容量来近似中立性。

此外，GEVN 架构在规模扩展方面表现良好，所需参数数量固定，不受投票者或候选人数量影响；相比之下，随着候选人数量增加，DeepSets 模型需要增加输入层和输出层的参数数量。值得注意的是，参数预算仅为 10 万的 GEVN，其表现与拥有 100 万参数的模型相当，这进一步彰显了 GEVN 架构的高效性。

学习最大化福利：先前研究 [2], [13], [23] 使用负对数似然损失，我们将其称为规则损失；该损失通过行为克隆促进投票规则的学习。我们认为，既然投票制度的首要目标是在选民群体中最大化某种福利函数，那么我们提出的直接优化福利的福利损失 $L_{\mathrm{sw}} = -\mathbb{E}[\mathrm{sw}_{\boldsymbol{U}}(p)]$ 便是实现这一目标的更好方法。

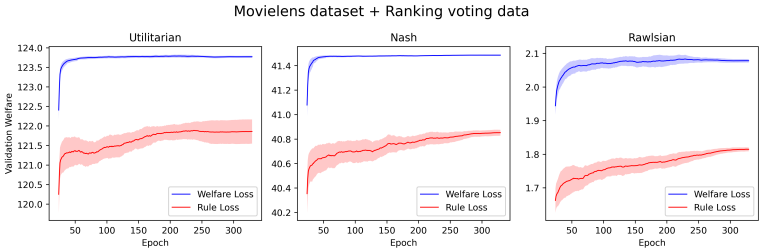
当目标是最大化福利，且无法获取真实的选民效用时，我们提出的福利损失优于规则损失。从图 3（左）（以及 D.2 节中的补充数据）可以看出，在测试的三种福利函数（功利主义、纳什和罗尔斯主义）下，福利损失始终能取得更高的福利。我们认为，这种提升源于福利损失的软惩罚结构。与对所有错误预测一视同仁地施加惩罚的规则损失不同，福利损失会区分接近最优的选择与较差的选择。

在 D.3 节中，我们看到，在验证集上，规则损失在找出获胜者方面的准确率往往高于福利损失，在测试集上偶尔也是如此（图 3（右））。然而，这并不重要，因为我们的目标并非优化模型识别获胜者（或学习投票规则／福利函数）的能力，而是试图最大化社会福利；福利损失成功实现了这一目标。

当真实的选民效用数据可用时，我们发现福利损失的表现几乎与规则损失相同。这是意料之中的，因为已知真实的基数效用时，最大化福利轻而易举。支持这一结论的数据见 D.4 节。

我们通过单调性损失学习不同福利函数的实验表明，GEVN 能够成功地强制满足单调性性质。简要分析见 D.5 节。

针对策略性投票的对抗训练：我们还考察了所学习的投票机制能否适应策略性投票行为。策略性投票者分为三种



Acc.	L_{sw}	L_{rule}
利用率。	**0.681** ± 0.003	0.674 ± 0.009
Nash	0.649 ± 0.007	**0.650** ± 0.011
Rawl。	0.393 ± 0.005	**0.397** ± 0.010

图 3：左：图中展示了在以选民的候选人排序为输入的情况下，采用我们提出的福利损失（蓝线）和规则损失（红线）训练 GEVN 架构时，各福利函数下的验证集福利。较深的线表示均值，阴影区域表示五次运行中均值上下一个标准差的范围。右：表格展示了相同五次运行中测试集上的准确率均值和标准差，最佳结果以**粗体**显示。L_{sw} 指（社会）福利损失，L_{rule} 指规则损失。

信息性逐渐增强的先验。在私人设定中，每位策略性投票者都能观察到自己的效用概况。在公开设定中，策略性投票者可以观察到整个选民群体的效用矩阵。在结果设定中，策略性投票者可以获知所有选民如实报告时会产生的选举结果。在所有情况下，投票者均匀抽样，因此预期有20% 的选民群体使用 GESN 生成的策略性选票，其余选民则如实报告效用。

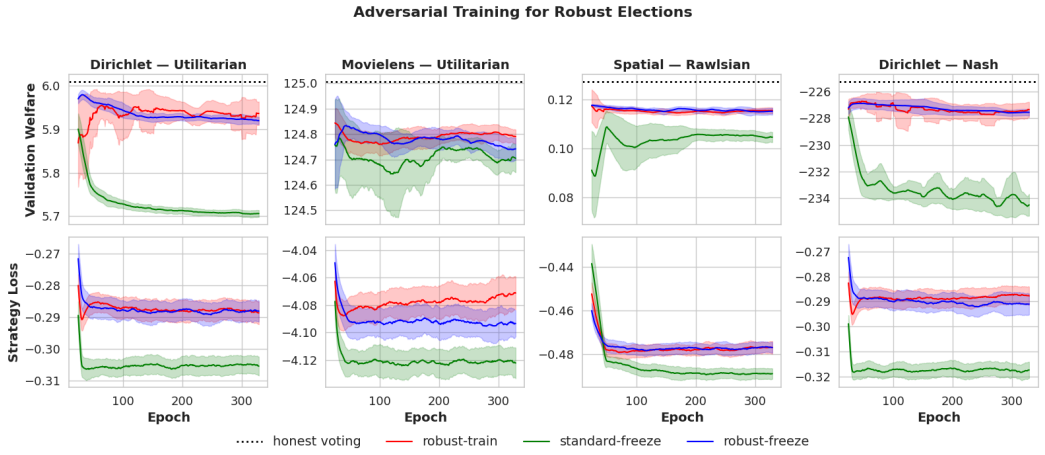


图 4：联合训练 GEVN 和 GESN 模块（私有信息）。上：验证数据集上的社会福利 下：训练过程中的理性损失。**standard-freeze**：冻结在如实投票上预训练的 GEVN 的权重，并训练 GESN。**robust-train**：联合训练 GESN，并对预训练的诚实 GEVN 进行微调。**robust-freeze**：将 robust-train 的输出作为预训练模型，并训练 GESN。图中使用 25 个训练轮次上的移动平均进行平滑处理，并绘制均值的 95% 置信区间。

私有信息设置下的结果如图 4 所示，公开结果信息设置下的结果见 D.6 节。对于所有目标，策略性投票都会导致福利降低。同时，理性损失有所下降，这表明即使信息有限，策略性投票者也能通过不诚实投票获得可衡量的优势 [21]。

在私有信息和公开信息设置下，与两种鲁棒情景相比，标准冻结情形的社会福利显著较低。这表明，所提出的对抗训练过程提高了系统应对一定比例策略性投票的韧性。特别值得一提的是鲁棒冻结情景：该情景仅针对固定的预训练鲁棒 GEVN 模块训练 GESN。在许多情景中，包括所有私有信息设置，新的策略性投票者都无法取得优于联合训练中所用 GESN 的结果。

相比之下，在投票者能够获知真实选举结果的情况下，鲁棒冻结情景并不总能取得成功。在这种设置下，训练后的 GEVN 仍可被部分利用（图 8）。这反映出，要针对一群信息充分且动机强烈的参与者学习鲁棒的选举规则，本身就存在困难；而吉巴德定理所蕴含的理论限制也进一步加剧了这一难题。

此外，据我们所知，我们的工作首次系统地探索了普选中的对抗训练图神经网络；在其余鲁棒训练设置下表现出的韧性，也展现了在现实假设下学习韧性的未来潜力。

5 局限性、讨论与结论

在本研究中，我们提出了一种利用图神经网络学习鲁棒投票机制的方法：通过使用二部图表示选举，确保选民匿名性、候选人中立性，并实现对任意数量选民和候选人的泛化。

Our 代码已开源，可访问 https://github.com/MarkHaoxiang/geometric-governance.

局限性与未来工作：我们的策略模块实例化考虑的信息先验集合并不完整，可能无法反映现实世界中的行为。通过引入其他类别的跨选民信号来扩展该框架，是一条明确的后续发展路径。此外，由于我们仅分析了单次选举，因此无法捕捉序贯选举中逐渐显现的更丰富动态；在序贯选举中，重复互动可以促成合作均衡（例如经典的囚徒困境解），从而更好地反映现实世界中的行为。受计算资源限制，我们无法充分探索不同策略性代理比例下的对抗训练——待获得更多预算后，我们计划对这一实证空间进行系统研究。最后，未来工作的一个方向是探索其他约束条件，例如一致性：合并两个具有相同获胜者的选民群体后，应保持该获胜者不变。进一步研究如何满足各种约束，尤其是通过几何或结构化方法，有助于为所学习的投票规则建立理论最坏情况保证，并增强用户信任。

结论：我们提出了一种利用图神经网络学习鲁棒投票机制的实用方法。首先，我们使用 EBG 表示一次选举，并利用 GEVN 训练投票机制，从设计上确保选民匿名性、候选人中立性，以及在选民数量和候选人数量方面的泛化能力。它还是投票机制设计空间上的通用逼近器。随后，我们提出福利损失，以改进社会福利最大化机制的无监督学习；并提出单调性损失，以遵循单调性约束。最后，我们使用 GESN 对策略性投票者的投票模式进行建模，并与 GEVN 进行对抗训练，以提升韧性。

更广泛的影响： 我们的方法在学习现有投票规则和新投票规则方面均取得了实证上令人瞩目的性能，同时能够抵御理性投票模式。这些贡献为解决阻碍先前研究应用于现实世界的关键问题提供了方案。我们相信，该方法将为机器学习技术应用于现实世界的选举环境开辟新的方向。

参考文献

[1] B. Manin, 《代议制政府的原则》。剑桥大学出版社，1997年。[2] C. Anil 和 X. Bao, “学习进行选举”，神经信息处理系统进展，第34卷，第8006–8017页，2021年。[3] Y. Shoham 和 K. Leyton-Brown, 《多智能体系统：算法、博弈论与逻辑基础》。剑桥大学出版社，2008年。[4] F. Brandt、V. Conitzer、U. Endriss、J. Lang 和 A. D. Procaccia, 《计算社会选择手册》。剑桥大学出版社，2016年。[5] H. P. Young, “孔多塞投票理论”，《美国政治学评论》，第82卷，第4期，第1231–1244页，1988年。[6] G. Doron 和 R. Kronick, “单记 transferable 投票：一个反常社会选择函数的例子”，《美国政治学杂志》，第303–311页，1977年。[7] K. J. Arrow, 《社会选择与个人价值》，第12卷。耶鲁大学出版社，2012年。[8] A. Gibbard, “投票方案的操纵：一般性结果”，《计量经济学：计量经济学会杂志》，第587–601页，1973年。[9] V. Conitzer 和 T. Sandholm, “候选人数量较少时操纵选举的复杂性”，载于 AAAI人工智能大会，2002年，第314–319页。[10] C. Boutilier、I. Caragiannis、S. Haber、T. Lu、A. D. Procaccia 和 O. Sheffet, “最优社会选择函数：功利主义视角”，载于 ACM电子商务会议，2012年，第197–214页。[11] I. Caragiannis、D. Kurokawa、H. Moulin、A. D. Procaccia、N. Shah 和 J. Wang, “最大纳什福利不合理的公平性”，《ACM经济学与计算汇刊》（TEAC），第7卷，第3期，第1–32页，2019年。[12] J. Rawls, 《正义论》，《应用伦理学》。Routledge，第21–29页，2017年。[13] L. Matone、B. Abramowitz、N. Mattei 和 A. Balakrishnan, “DeepVoting：使用定制化嵌入学习投票规则”，arXiv预印本 arXiv:2408.13630，2024年。[14] P. Faliszewski 和 A. D. Procaccia, “人工智能对操纵的战争：我们赢了吗？”，《人工智能杂志》，第31卷，第4期，第53–64页，2010年。[15] H. Nurmi, 《比较投票制度》，第3卷。施普林格科学与商业媒体，2012年。[16] P. Emerson, “原始博尔达计数法与部分投票”，《社会选择与福利》，第40卷，第2期，第353–358页，2013年。[17] D. R. Woodall, “单席位优先选举规则的单调性”，《离散应用数学》，第77卷，第1期，第81–98页，1997年。[18] M. Zaheer、S. Kottur、S. Ravanbakhsh、B. Poczos、R. R. Salakhutdinov 和 A. J. Smola, “深度集合”，神经信息处理系统进展，第30卷，2017年。[19] J. Lee、Y. Lee、J. Kim、A. Kosiorek、S. Choi 和 Y. W. Teh, “集合变换器：一种用于置换不变神经网络的注意力框架”，载于国际机器学习会议，2019年，第3744–3753页。[20] A. Hylland, “以彩票为结果且策略集合无限的投票程序的策略无关性”，技术报告，1980年。[21] W. H. Holliday、A. Kristoffersen 和 E. Pacuit, “在有限信息下学习操纵”，载于 AAAI人工智能大会，2025年。[22] A. D. Procaccia、A. Zohar、Y. Peleg 和 J. S. Rosenschein, “投票规则的可学习性”，《人工智能》，第173卷，第12–13期，第1133–1149页，2009年。[23] D. Firebanks-Quevedo, “机器学习？用于我的选举？这比你想象的更有可能：通过神经网络实现投票规则”，2020年。[24] F. Rosenblatt, “感知机：一种用于大脑中信息存储与组织的概率模型”，《心理学评论》，第65卷，第6期，第386页，1958年。

[25] O. Vinyals、M. Fortunato 和 N. Jaitly，《指针网络》，《神经信息处理系统进展》，第 28 卷，2015年。[26] Z. Wu、S. Pan、F. Chen、G. Long、C. Zhang 和 P. S. Yu，《图神经网络综合综述》，《IEEE神经网络与学习系统汇刊》，第 32 卷，第 1 期，第 4–24 页，2020年。[27] P. W. Battaglia 等，《关系归纳偏置、深度学习与图网络》，arXiv 预印本 arXiv:1806.01261，2018年。[28] K. Xu、W. Hu、J. Leskovec 和 S. Jegelka，《图神经网络的能力究竟有多强？》，发表于《国际学习表征会议》，2019年。[29] Z. Chen、S. Villar、L. Chen 和 J. Bruna，《图同构测试与基于图神经网络的函数逼近之间的等价性》，《神经信息处理系统进展》，第 32 卷，2019年。[30] J. H. Smith，《可变选民群体下的偏好聚合》，《计量经济学：计量经济学会杂志》，第 1027–1041 页，1973年。[31] S. D. Fisher，《策略性投票的定义与测量：理性选择的作用》，《英国政治学杂志》，第 34 卷，第 1 期，第 152–166 页，2004年。[32] A. Rolland、J.-B. Aubin、I. Gannaz 和 S. Leoni，《关于评价式投票数据模拟的说明》，arXiv 预印本 arXiv:2104.07666，2021年。[33] F. M. Harper 和 J. A. Konstan，《MovieLens数据集：历史与背景》，《ACM交互式智能系统汇刊》，第 5 卷，第 4 期，第 1–19 页，2015年。[34] S. Bouveret 等，《法国总统选举的 Voter Autrement 2017》，2019年。[35] D. P. Kingma 和 J. Ba，《Adam：一种随机优化方法》，发表于《国际学习表征会议》，2015年。[36] I. Loshchilov 和 F. Hutter，《SGDR：带热重启的随机梯度下降》，发表于《国际学习表征会议》，2017年。[在线]。网址：https://openreview.net/forum?id=Skq89Scxx[37] L. Hornischer 和 Z. Terzopoulou，《学习如何遵循原则进行投票：神经网络集体决策的公理化见解》，arXiv 预印本 arXiv:2410.16170，2024年。[38]S.-M. Hsieh、C.-C. Hsu 和 L.-F. Hsu，《一种高效的标记图同构测试方法》，发表于《计算科学及其应用》，2006年，第 422–431 页。[39] A. Leman 和 B. Weisfeiler，《将图约化为规范形及约化过程中产生的代数》，《科学技术信息》，第 2 卷，第 9 期，第 12–16 页，1968年。[40] M. Blondel、O. Teboul、Q. Berthet 和 J. Djolonga，《快速可微排序与排名》，发表于《国际机器学习会议》，2020年，第 950–959 页。[41] A. Paszke 等，《PyTorch：一种命令式风格的高性能深度学习库》，第 32 卷，2019年。[42]M. Fey 和 J. E. Lenssen，《使用 PyTorch Geometric 进行快速图表征学习》，发表于《ICLR图与流形表征学习研讨会》，2019年。[43] J. L. Ba、J. R. Kiros 和 G. E. Hinton，《层归一化》，arXiv 预印本 arXiv:1607.06450，2016年。[44] A. F. Agarap，《使用修正线性单元（ReLU）进行深度学习》，CoRR，2018, [在线]。网址：http://arxiv.org/abs/1803.08375[45] S. Ioffe 和 C. Szegedy，《批归一化：通过减少内部协变量偏移加速深度网络训练》，发表于《国际机器学习会议》，2015年，第 448–456 页。[46]J. Xu、Z. Li、B. Du、M. Zhang 和 J. Liu，《让 Reluplex 更实用：带泄漏的ReLU》，发表于《IEEE计算机与通信研讨会》，2020年，第 1–7 页。

A 证明

A.1 匿名性与中立性的证明

设 $G = (V \cup C, E)$ 为一个选举二部图（EBG），其中 $V = (v_1, ..., v_n)$ 是选民集合，$C = (c_1, ..., c_m)$ 是候选人集合。G 的拓扑结构由邻接矩阵 $\boldsymbol{A} \in \{0,1\}^{(n+m)\times(n+m)}$完全编码。此外，设 $\boldsymbol{X} \in \mathbb{R}^{(n+m)\times d}$ 为节点特征矩阵，其中包含用于区分选民和候选人的独热编码向量：前 n 行对应选民，后 m 行对应候选人。

如果函数 $f : (\boldsymbol{X}, \boldsymbol{A}) \to \mathbb{R}^{n+m}$ 满足以下条件，则称其相对于对称群 S_{n+m} 置换不变：$\forall \pi \in S_{n+m}$

$$f(\boldsymbol{P}_\pi \boldsymbol{X}, \boldsymbol{P}_\pi \boldsymbol{A} \boldsymbol{P}_\pi^T) = f(\boldsymbol{X}, \boldsymbol{A}), \tag{8}$$

其中，$\boldsymbol{P}_{\{\pi\}}$ 是一个置换矩阵，它根据 π 对 $\boldsymbol{X}$ 和 $\boldsymbol{A}$ 的行和列进行置换。不失一般性，如果函数 $f : (\boldsymbol{X}, \boldsymbol{A}) \to \mathbb{R}^{n+m}$ 满足以下条件，则称其相对于 S_{n+m} 置换等变：$\forall \pi \in S_{n+m}$

$$f(\boldsymbol{P}_\pi \boldsymbol{X}, \boldsymbol{P}_\pi \boldsymbol{A} \boldsymbol{P}_\pi^T) = \boldsymbol{P}_\pi f(\boldsymbol{X}, \boldsymbol{A}). \tag{9}$$

投票机制 $\nu : (\boldsymbol{X}, \boldsymbol{A}) \to \mathbb{R}^m$ 通过使用矩阵 $\boldsymbol{M} = [\boldsymbol{0}_{m\times n} \mid \boldsymbol{I}_m] \in \{0,1\}^{m\times(n+m)}$ 对 f 的结果进行掩码操作，从而将 m 候选人隔离开来。形式化地说：

$$\nu(\boldsymbol{X}, \boldsymbol{A}) = \boldsymbol{M} f(\boldsymbol{X}, \boldsymbol{A}). \tag{10}$$

因此，当 ν 在选民的任意置换下保持 **匿名性** 不变时，ν 满足 **匿名性**；当 ν 在候选人 [37] 的任意置换下保持 等变 时，ν 满足 **中立性**。

证明：设 f $\mathbb{R}^{n+m}$ $\boldsymbol{z}_V \in \mathbb{R}^n$ V $\boldsymbol{z}_C \in \mathbb{R}^m$ $\boldsymbol{z} = [\boldsymbol{z}_V^T, \boldsymbol{z}_C^T]^T \in$ 为一个图神经网络 C（GNN），其输出为 ，其中 是选民的得分， 是候选人的得分。

匿名性证明：

证明。设 $\sigma \in S_V \cong S_n$ 为作用于投票者节点 V上的一个置换，其表示为如下分块对角矩阵 $\boldsymbol{P}_\sigma \in \{0,1\}^{(n+m)\times(n+m)}$ ：

$$\boldsymbol{P}_\sigma = \begin{pmatrix} \boldsymbol{P}_\sigma^{(V)} & \boldsymbol{0}_{n\times m} \\ \boldsymbol{0}_{m\times n} & \boldsymbol{I}_m \end{pmatrix}, \tag{11}$$

whe其中 $\boldsymbol{P}_\sigma^{(V)}$ 是作用于选民节点的 σ 的$n \times n$ 置换矩阵。根据置换 n equi$f$的不变性，可得：

$$f(\boldsymbol{P}_\sigma \boldsymbol{X}, \boldsymbol{P}_\sigma \boldsymbol{A} \boldsymbol{P}_\sigma^T) = \boldsymbol{P}_\sigma f(\boldsymbol{X}, \boldsymbol{A}). \tag{12}$$

现在，考虑一个投票机制 ν，我们有：

$$\nu(\boldsymbol{P}_\sigma \boldsymbol{X}, \boldsymbol{P}_\sigma \boldsymbol{A} \boldsymbol{P}_\sigma^T) = \boldsymbol{M} \boldsymbol{P}_\sigma f(\boldsymbol{X}, \boldsymbol{A}). \tag{13}$$

该乘积 $\boldsymbol{M}\boldsymbol{P}_\sigma$ 可分解为：

$$\begin{aligned} \boldsymbol{M}\boldsymbol{P}_\sigma &= [\boldsymbol{0}_{m\times n} \mid \boldsymbol{I}_m] \begin{pmatrix} \boldsymbol{P}_\sigma^{(V)} & \boldsymbol{0}_{n\times m} \\ \boldsymbol{0}_{m\times n} & \boldsymbol{I}_m \end{pmatrix} \\ &= \left[\boldsymbol{0}_{m\times n} \cdot \boldsymbol{P}_\sigma^{(V)} + \boldsymbol{I}_m \cdot \boldsymbol{0}_{m\times n} \mid \boldsymbol{0}_{m\times n} \cdot \boldsymbol{0}_{n\times m} + \boldsymbol{I}_m \cdot \boldsymbol{I}_m\right] \\ &= [\boldsymbol{0}_{m\times n} \mid \boldsymbol{I}_m] \\ &= \boldsymbol{M}. \end{aligned} \tag{14}$$

因此，

$$\nu(\boldsymbol{P}_\sigma \boldsymbol{X}, \boldsymbol{P}_\sigma \boldsymbol{A} \boldsymbol{P}_\sigma^T) = \boldsymbol{M} f(\boldsymbol{X}, \boldsymbol{A}) = \nu(\boldsymbol{X}, \boldsymbol{A}). \tag{15}$$

Thu因此，投票机制 ν 对投票者的任意排列均保持不变，满足匿名性。□

中立性证明：

*Pro*的。 令 $\tau \in S_C \cong S_m$ 为作用于候选节点 C上的一个置换，以块的形式表示 k-diag矩阵 $\boldsymbol{P}_\tau \in \{0,1\}^{(n+m)\times(n+m)}$ 如下：

$$\boldsymbol{P}_\tau = \begin{pmatrix} \boldsymbol{I}_n & \mathbf{0}_{n\times m} \\ \mathbf{0}_{m\times n} & \boldsymbol{P}_\tau^{(C)} \end{pmatrix}, \tag{16}$$

其中，$\boldsymbol{P}_\tau^{(C)}$ 是作用于候选人节点的$m \times m$ 置换矩阵。

根据 f 的置换等变性，我们有：

$$f(\boldsymbol{P}_\tau \boldsymbol{X}, \boldsymbol{P}_\tau \boldsymbol{A} \boldsymbol{P}_\tau^T) = \boldsymbol{P}_\tau f(\boldsymbol{X}, \boldsymbol{A}). \tag{17}$$

应用投票机制 ν:

$$\nu(\boldsymbol{P}_\tau \boldsymbol{X}, \boldsymbol{P}_\tau \boldsymbol{A} \boldsymbol{P}_\tau^T) = \boldsymbol{M}\boldsymbol{P}_\tau f(\boldsymbol{X}, \boldsymbol{A}). \tag{18}$$

乘积 $\boldsymbol{M}\boldsymbol{P}_\tau$:

$$\begin{aligned} \boldsymbol{M}\boldsymbol{P}_\tau &= [\mathbf{0}_{m\times n} \mid \boldsymbol{I}_m] \begin{pmatrix} \boldsymbol{I}_n & \mathbf{0}_{n\times m} \\ \mathbf{0}_{m\times n} & \boldsymbol{P}_\tau^{(C)} \end{pmatrix}, \\ &= \left[\mathbf{0}_{m\times n} \cdot \boldsymbol{I}_n + \boldsymbol{I}_m \cdot \mathbf{0}_{m\times n} \mid \mathbf{0}_{m\times n} \cdot \mathbf{0}_{n\times m} + \boldsymbol{I}_m \cdot \boldsymbol{P}_\tau^{(C)}\right] \\ &= \left[\mathbf{0}_{m\times n} \mid \boldsymbol{P}_\tau^{(C)}\right] \\ &= \left[\boldsymbol{P}_\tau^{(C)} \cdot \mathbf{0}_{m\times n} \mid \boldsymbol{P}_\tau^{(C)} \cdot \boldsymbol{I}_m\right] \\ &= \boldsymbol{P}_\tau^{(C)} [\mathbf{0}_{m\times n} \mid \boldsymbol{I}_m] \\ &= \boldsymbol{P}_\tau^{(C)} \boldsymbol{M}. \end{aligned} \tag{19}$$

因此，$\boldsymbol{M}\boldsymbol{P}_\tau = \boldsymbol{P}_\tau^{(C)}\boldsymbol{M}$。代回方程（18）：

$$\begin{aligned} \nu(\boldsymbol{P}_\tau \boldsymbol{X}, \boldsymbol{P}_\tau \boldsymbol{A} \boldsymbol{P}_\tau^T) &= \boldsymbol{P}_\tau^{(C)} \boldsymbol{M} f(\boldsymbol{X}, \boldsymbol{A}) \\ &= \boldsymbol{P}_\tau^{(C)} \nu(\boldsymbol{X}, \boldsymbol{A}). \end{aligned} \tag{20}$$

因此，候选人 C 的得分 $\nu(\boldsymbol{X}, \boldsymbol{A})$ 会根据 $\boldsymbol{P}_\tau^{(C)}$ 进行置换，也就是说，按照对候选人自身施加的置换进行置换，从而实现中立性。□

由于 ν 在所有投票者置换下保持不变（匿名性公理），并且在所有候选人置换下保持等变（中立性公理），因此该命题成立。■

A.2 GEVN 普适性的证明

我们重述带标签图同构的定义 [38]。设 l、l' 表示顶点上的标记函数。

定义 11 $G' = (\mathcal{V}', \mathcal{E}', l')$ 与 $G = (\mathcal{V}, \mathcal{E}, l)$ 同构于，当且仅当存在一个双射函数 $f : \mathcal{V} \to \mathcal{V}'$，使得（标记图的同构）

$$\forall u \in \mathcal{V}. l(u) = l'(f(u)), \text{and} \tag{21}$$

$$\forall u, v \in \mathcal{V}. (u, v) \in \mathcal{E} \leftrightarrow (f(u), f(v)) \in \mathcal{E}' \tag{22}$$

我们证明，1-WL [39] 测试在投票图上是完备的。由于 1-WL 测试无法处理边特征，我们将边信息编码为附加节点上的标签。

定义 1.2。（三部 EBG）三部 EBG（T-EBG）$(V, \mathcal{E}, l)$ 是一个具有 $\mathcal{V} = V \cup C \cup U$ 的 EBG $(V \cup C, \mathcal{E}', l')$ 的表示。U 是提取出的偏好剖面集合，V 是选民集合，C 是候选人集合：$\{u_{vc} \mid \forall (v, c) \in \mathcal{E}'\}$。接下来，对于每个 u_{vc}，在 $\mathcal{E}$ 中构造 (v, u) 和 (u, c) 作为两条边。与 EBG 一样，T-EBG 中的每个节点都根据其用途进行标记：

$$l(u) = \begin{cases} 0' & \text{if } u \in V \\ l'(v, c) & \text{if } u_{vc} \in U \\ 1' & \text{if } u \in C \end{cases} \tag{23}$$

注意，T-EBG 构成一个三部图，所有边均连接 V、U 或 U、C。

回顾一下，1-WL 算法为每个顶点分配一个编码 l 的初始颜色，然后使用 $\text{HASH}(\text{col}(v), \{\text{col}(u) : u \in \mathcal{N}_v\})$ 迭代细化颜色。随后，只有在达到稳定着色后，顶点颜色多重集相匹配时，它才无法区分两个图 G 和 G'。

引理 1.3。设 G 和 G' 是由两个 EBG 按照 定义 1.2生成的 T-EBG。如果 1-WL 无法区分 G 和 G'，那么存在一个保标签同构 $f : G \leftrightarrow G'$。

证明 引理 1.3 $\{\{\varphi(x) : x \in G\}\} = \{\{\varphi'(x') : x' \in G'\}\}$。假设稳定的 1-WL 颜色的多重集相同，。我们通过一个 **颜色保持遍历**构造一个双射 $f : V_G \leftrightarrow V_{G'}$，该遍历恰好访问 G 的每条边一次。

1) 初始匹配。选取一个尚未匹配的投票者 $v \in V_G$，并任选一个满足 $\varphi(v) = \varphi'(v')$ 的 $v' \in V_{G'}$；令 $f(v) := v'$。如果此时所有顶点都已匹配，则完成。

2) 前向步骤 $V \leftrightarrow U \leftrightarrow C$。选取一条尚未遍历的边 $(v, u) \in \mathcal{E}_G$，使其满足 v 已匹配且 $u \in U_G$。令 $(u, c) \in \mathcal{E}_G$ 为满足 $c \in C_G$ 的边的延续。

a) 如果 c 已匹配。路径 $(f(v) \leftrightarrow u' \leftrightarrow f(c))$ 存在于 G' 中，因为 G' 是完全二部图。将 u 与这个唯一的 u' 匹配；1-WL 得出 $\varphi(u) = \varphi'(u')$。

b) 如果 c 尚未匹配。任取一个与满足 $\varphi(u) = \varphi'(u')$ 的 $f(v)$ 相邻的 $u' \in G'_{(u',c')} \in E_{G'}$ 并令 $f(u) := u'$ $f(c) := c'$。设定 和 。从 c 继续遍历；如果没有尚未遍历的出边，就跳转到任意一个仍有未遍历出边的已匹配顶点。这样的顶点一定存在，因为 T-EBG 是欧拉的。

3) 后向步骤 $C \leftrightarrow U \leftrightarrow V$。与前向步骤对称，从一个已匹配的候选人出发，先遍历 (c, u)，再遍历 (u, v)。

4) 重新开始。如果仍有未匹配的顶点，就从一名未匹配的投票者开始，返回前向步骤。由于 G 有限，最终必定会让每个顶点都匹配，并遍历每条边，从而终止。

由此得到的映射 f 按构造是双射，满足 $\varphi(x) = \varphi'(f(x))$（颜色相等意味着标签相等，从而证明了（21），并且保持邻接关系，因为每条像边都是在 G'（证明了（22）中选取的）。因此，f 就是所需的保标签同构。

□

B 经典投票规则的更多细节

相对多数投票

In t在这种投票方案中，每位投票者都投票给自己最喜欢的候选人，而选举的获胜者 s the 获得选票最多的候选人。

如果选举由一个大小为 $n \times m$ （选民数 × 候选人数）的投票矩阵 $\boldsymbol{V}$ 表示，其中 $\boldsymbol{V}_{i,\ j}$ 表示候选人 j 在投票者 i的偏好档案中的排名位置，那么每位候选人的得分为

$$\text{plurality}(\boldsymbol{V}, j) = \sum_{i=1}^{n} \mathbf{1}_{\boldsymbol{V}_{i,j}=1} \tag{24}$$

波达计数法

根据波达计数为每位候选人赋予相应的分数

$$\text{borda}(\boldsymbol{V}, j) = \sum_{i=1}^{n} \left(1 - \frac{\boldsymbol{V}_{i,j}}{m}\right) \tag{25}$$

科普兰计数法

科普兰计数法对单次选举中的 $\frac{m(m-1)}{2}$ 个两两子选举进行建模。候选人在两两选举中获胜得 1 分，平局得 $\frac{1}{2}$ 分，否则得 0 分。得分最高的候选人即为获胜者。

最大最小值法

与科普兰法类似，最大最小值法也对两两选举进行建模。候选人的得分取决于其在最不利对决中获得的选票数。

单记可转移投票（STV）

在单记可让渡投票中，每位选民最初按照相对多数投票的方式投票。计票后，如果没有候选人获得绝对多数选票，就淘汰表现最差的候选人。此前投票支持该候选人的选民，可以改为投票支持剩余候选人。这个淘汰过程会不断重复，直到只剩下获胜者。

C 序数投票系统的应用

虽然我们的主要关注点是基数投票规则，但该框架也很容易扩展到序数（排序）投票规则。如第2节所述，可以将已知排序的信号转换为归一化博尔达分数（B节），作为 EBG 中的边特征，同时保留匿名性和中立性保证。我们在学习经典投票规则的实验中应用了这一转换。

将策略模块扩展到序数系统需要端到端可微。关于可微排序算子的研究 [40] 使基数得分能够平滑地转换为序数排名，同时保留反向传播能力。正文主要关注基数制，因为它天然适用于自主智能体的应用，也能避免可微排序带来的计算开销（尽管理论上效率很高，但撰写本文时现成的实现尚不支持 GPU训练）。

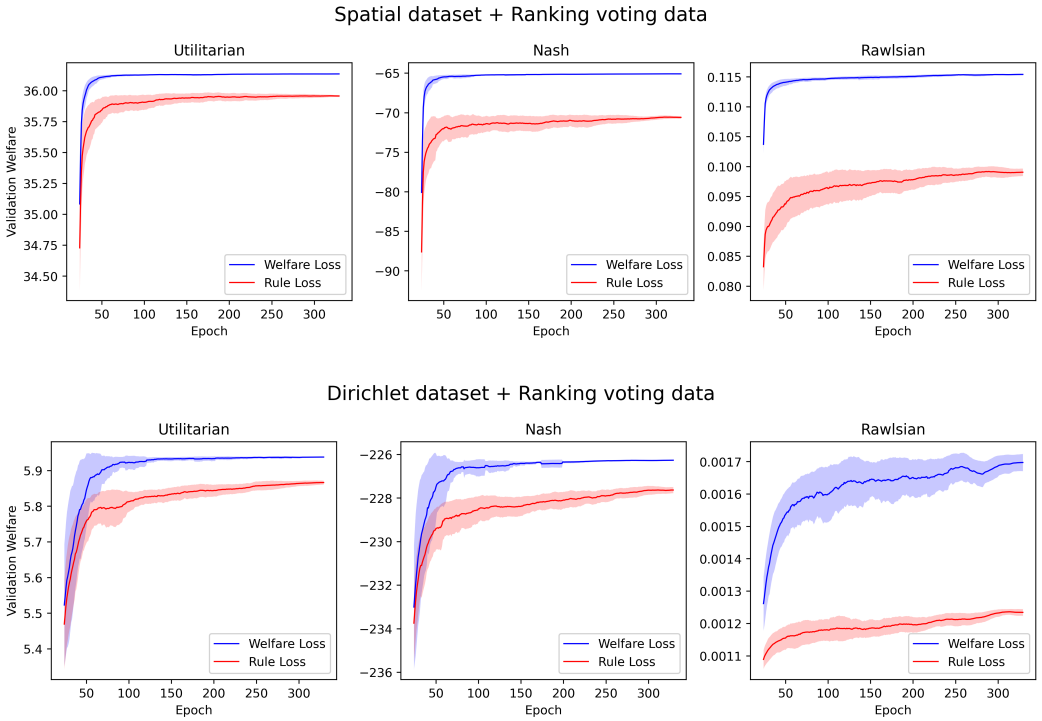
D 其他实验结果

D.1 学习投票规则：泛化到真实世界数据集

我们在真实世界数据集上测试了训练好的 GEVN 模型。结果见表 2。我们验证了 GEVN 在学习现有投票机制时能够有效地生成真实世界数据集。与 [2], 类似，我们发现，在真实世界数据上的评估准确率高于在合成数据上的评估准确率。

D.2 使用排名数据时，福利损失优于规则损失

下方的图展示了有关利用排名数据实现福利最大化的更多数据。我们看到，在其余两个数据集中，福利损失的表现优于规则损失。深色线条表示均值，阴影区域表示每张图中五次运行的结果高于和低于均值一个标准差的范围。



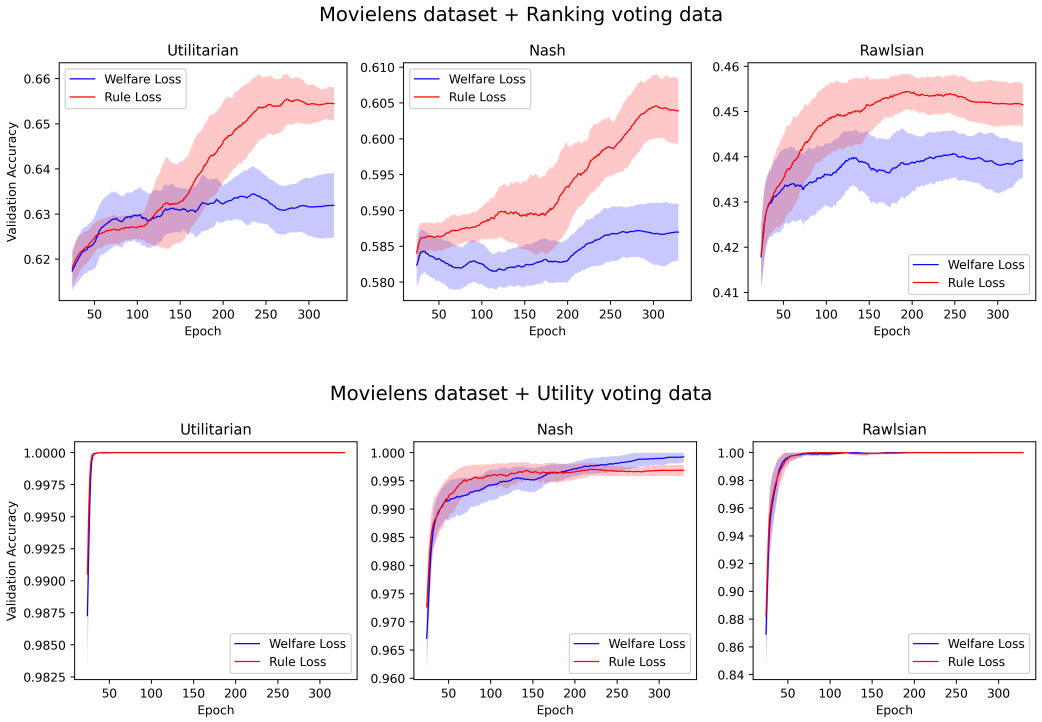
D.3 规则损失在准确率方面优于福利损失

表 2：在真实世界数据集上测试模仿经典投票规则的效果。对于 *Grenoble* 数据集，抽样选取 50 名选民和 5 名候选人。对于 *MovieLens* 数据集，抽样选取 100 名选民和 10 名候选人。省略的置信区间是指 95% 区间低于 0.01。

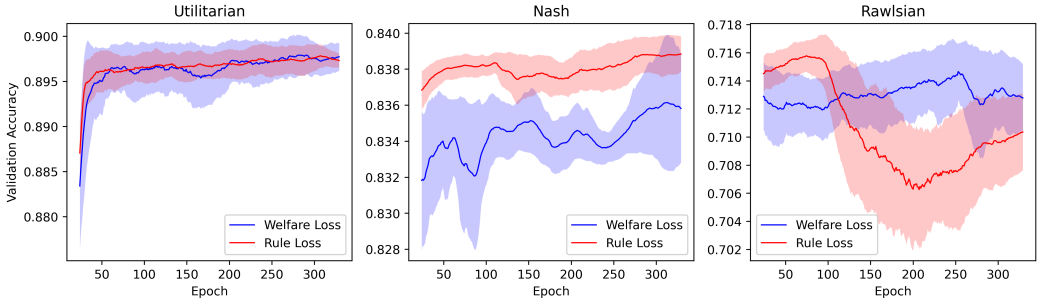
		小型 GEVN	中型 GEVN
相对多数制	Grenoble	1.00	1.00
	MovieLens	1.00	1.00
博尔达计数法	Grenoble	1.00	1.00
	MovieLens	1.00	1.00
科普兰计数法	Grenoble	0.94 ± 0.01	0.92 ± 0.01
	MovieLens	0.89 ± 0.01	0.89
最大最小值法	Grenoble	0.92 ± 0.01	0.92 ± 0.01
	MovieLens	0.90 ± 0.01	0.90 ± 0.01
STV	Grenoble	0.94 ± 0.01	0.95 ± 0.01
	MovieLens	0.87	0.87

下方的图展示了在全部三种福利函数（功利主义、纳什和罗尔斯主义）下，以选民排名和效用为输入时，福利损失和规则损失的验证准确率。深色线条表示均值，阴影区域表示每张图中五次运行的结果高于和低于均值一个标准差的范围。

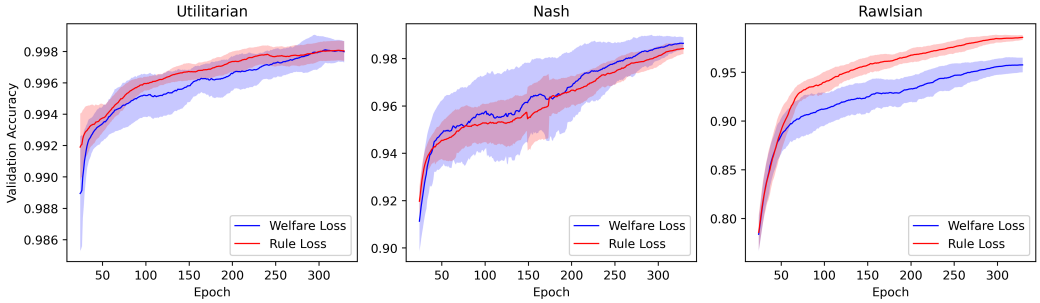
我们经常观察到，在排名数据上，规则损失的准确率高于福利损失，但这并不重要，因为我们的主要目标是最大化福利。当有真实的选民效用数据时，这两种损失函数的表现相似。请注意，在狄利克雷数据集的验证集上，模型尚未收敛；我们认为，足够大的训练集或许能够改善这一问题。



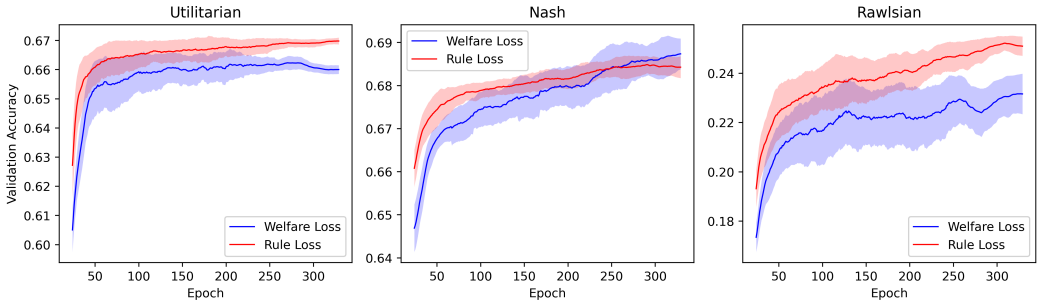
Spatial dataset + Ranking voting data



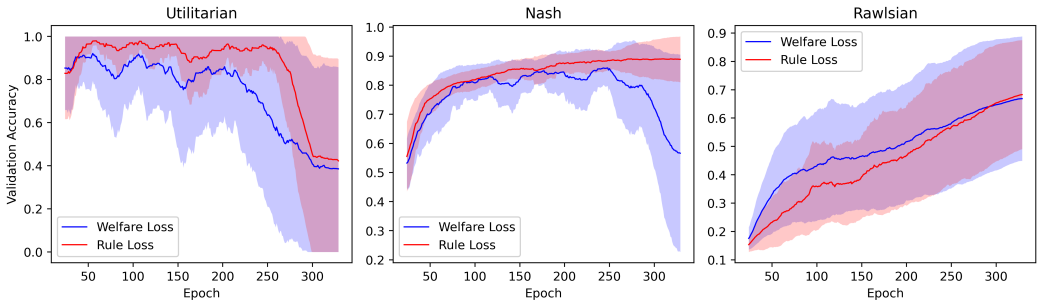
Spatial dataset + Utility voting data



Dirichlet dataset + Ranking voting data



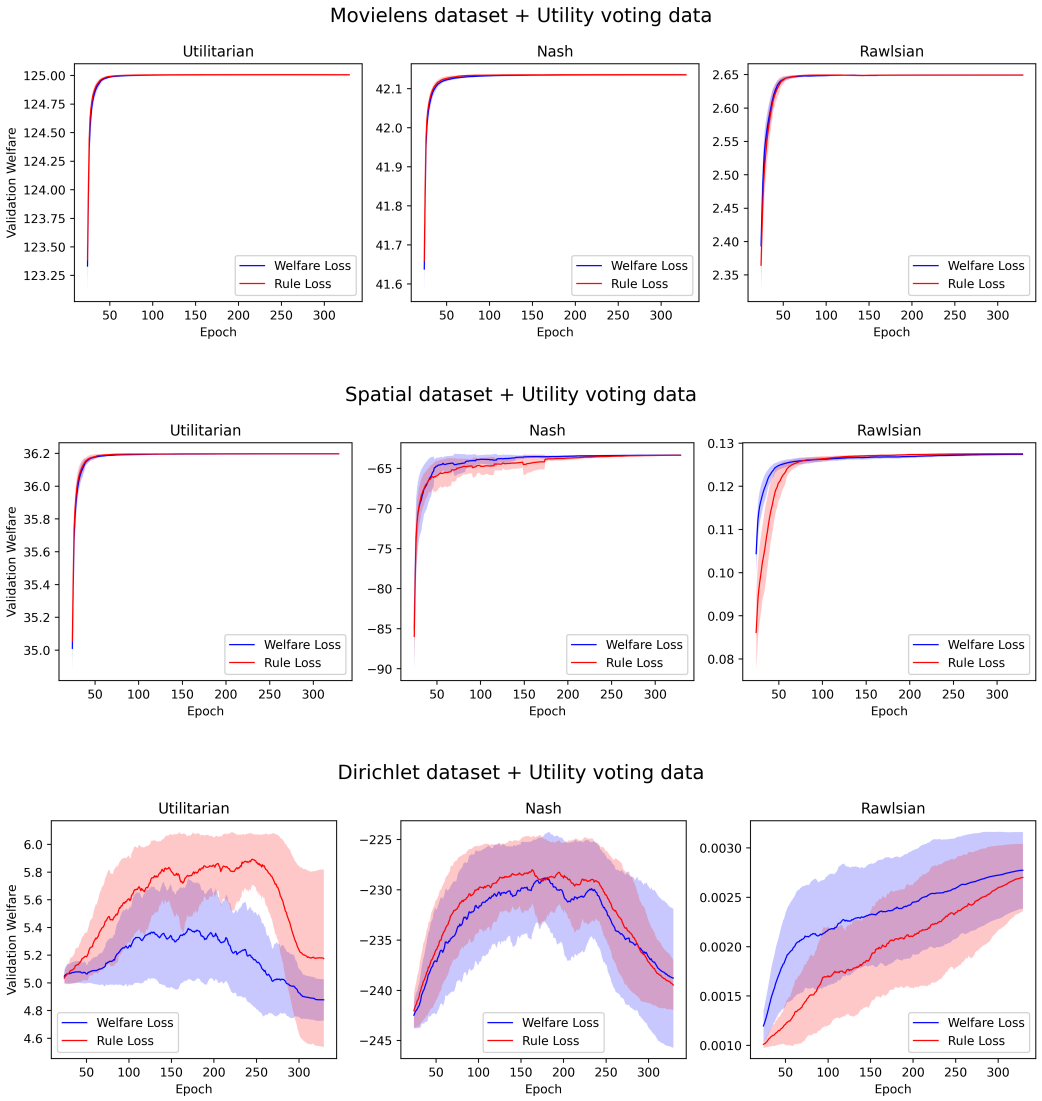
Dirichlet dataset + Utility voting data



D.4 已知选民效用时，福利损失与规则损失高度吻合

下图显示，在 Movieslens 和空间数据集的情况下，当真实选民效用数据可用时，福利损失和规则损失所达到的最大福利相同。深色线条表示均值，阴影区域则显示每幅图中五次运行所得均值上下各一个标准差的范围。

请注意，对于狄利克雷数据集，两种损失函数在验证集上均未实现收敛（尽管在训练期间实现了收敛）。通过使用更大的训练集进一步微调，很可能可以改善这一情况。



D.5 单调性损失对学习福利函数的影响

通过添加单调性损失来施加单调性性质时，GEVN 仍能成功学习功利主义、纳什和罗尔斯主义福利，且验证准确率或测试准确率均未退步（结果见表3）。这是预期之中的，因为这些福利函数本质上是单调的，因此成功学习福利函数自然会使新增的单调性损失最小化。我们的实证结果也为前述说法提供了证据：即使我们不主动施加单调性约束，单调性损失也会降至 0。

Ta表 3：显示测试集上平均测试准确率和标准差的表格
fiv次运行。

福利函数	功利主义 + 单调性损失	纳什 + 单调性损失	罗尔斯主义 + 单调性损失
测试准确率	0.999 ± 0.001	0.996 ± 0.001	0.989 ± 0.006

从下方的图中可以看到，在学习单记可让渡投票制规则时，我们能够成功降低单调性损失，同时不影响验证准确率。尽管从理论上讲，STV 并不具备单调性，我们仍能学到一个单调的局部极小值。

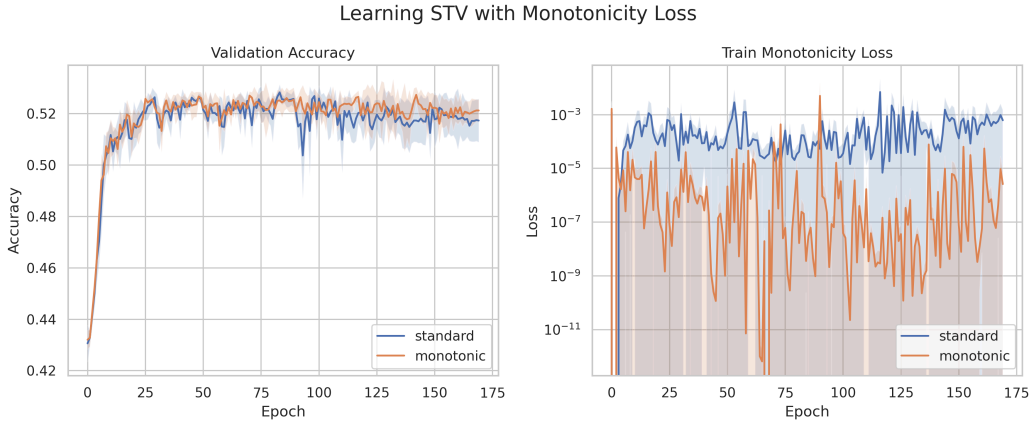


图5：深色线表示均值，阴影区域表示每个图中五次运行结果的均值上下一个标准差范围。

D.6 对抗训练的附加结果

下方的图展示了在使用私有、公开和结果信息设置时，对抗式策略性训练过程（第 4.5 节）的训练曲线（经 25 个数据点的窗口平滑处理）。

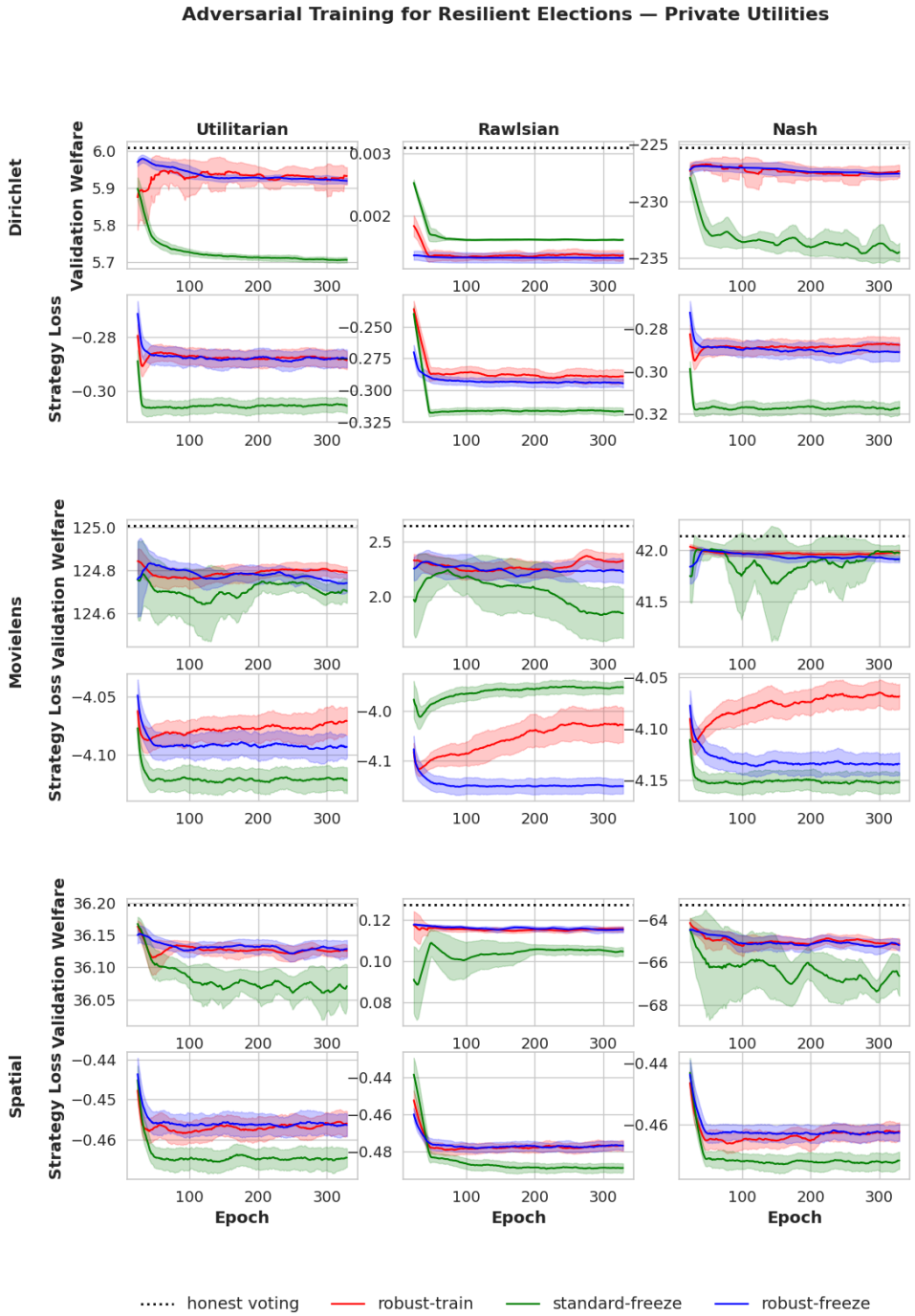


图 6：在选民能够获取自身效用的设置下进行策略性训练。这是一个有限信息设置；即使投票模块被冻结，采用对抗训练的投票模块在存在20% 策略性投票的情况下，也能持续带来更高的社会福利。

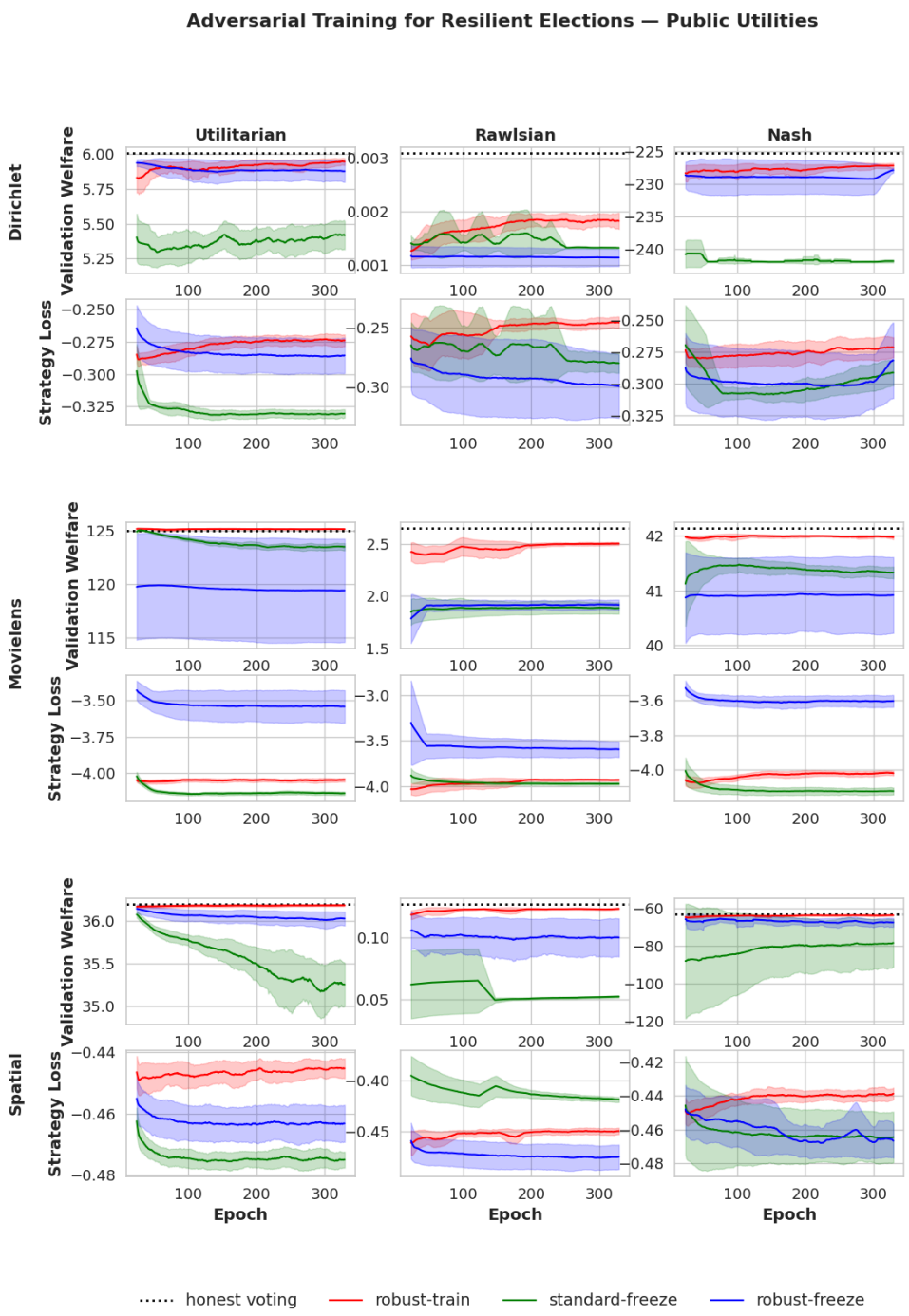


图7：在选民可以获取整个选民群体公共效用的情形下进行策略性训练。与冻结 GEVN 相比，同时训练 GEVN 和 GESN 总能带来较高的社会福利；但在某些数据集与社会福利函数的组合中，对抗训练后冻结的 GEVN 表现不如未训练的情况。尽管如此，值得注意的是，在大多数（6/9）实验设置中，我们的方法都能产生能够抵御策略性投票的投票方法。

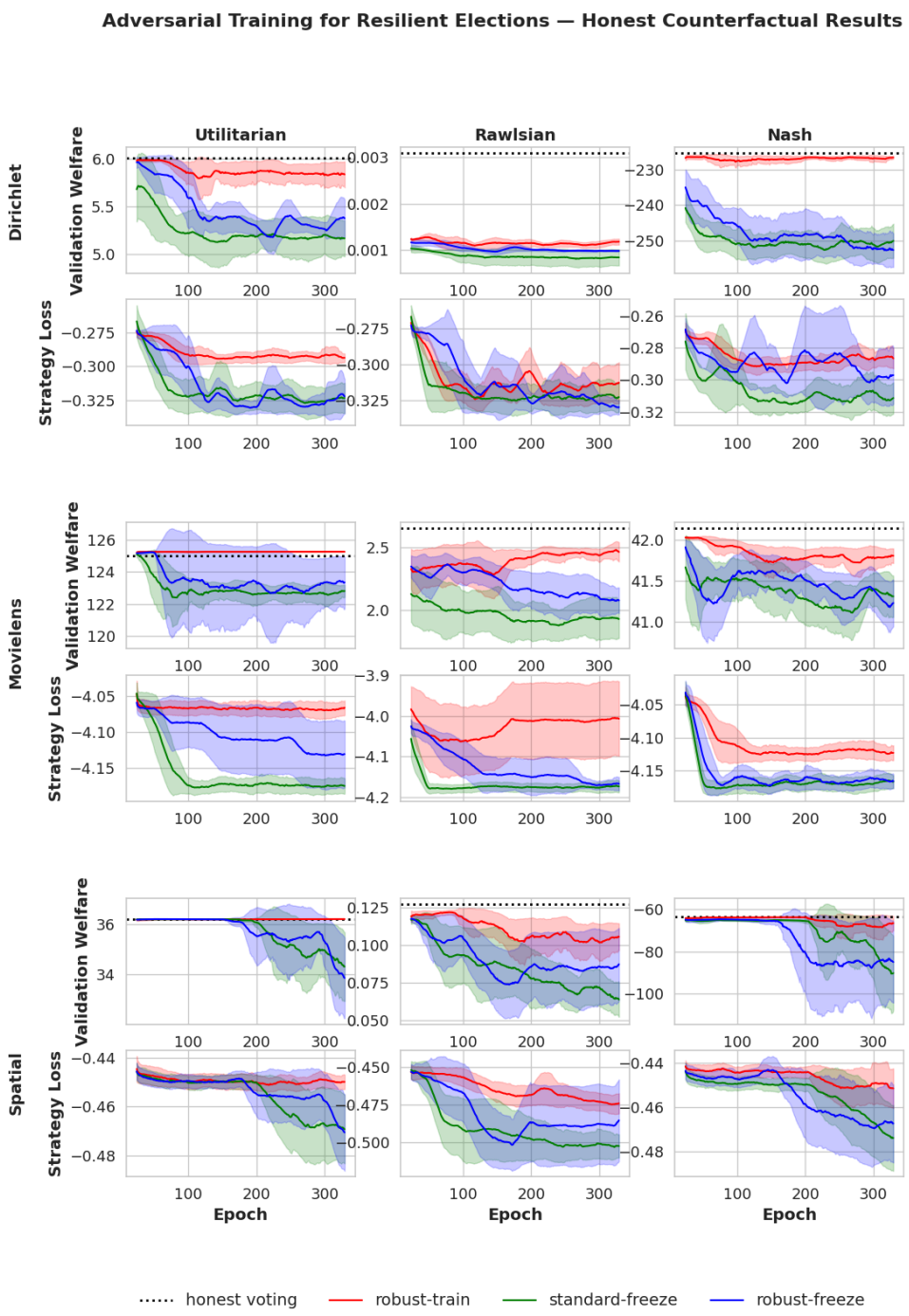


图 8：在选民可以获取假设情境下所有选民均诚实投票时的选举结果这一设定中进行策略性训练。与策略性投票者同时训练 GESN，实证表明能够维持稳定且较高的社会福利，也反映了现实世界中的情形：设计者可以在每次选举后对选举机制进行微调。冻结后的 GESN 不具备鲁棒性，这体现了针对信息充分且具有动机的行为者学习鲁棒投票规则所固有的困难。

E 实验详情与超参数

本节详细介绍复现实验所需的信息。我们的方法基于 PyTorch Geometric 库 [42] （MIT 许可证），使用 PyTorch 深度学习框架 [41] （BSD 许可证）实现，同时还使用了 NumPy（BSD 许可证）、Pandas（BSD 许可证）和 tqdm（MIT 许可证）。图表由 Matplotlib（PSF 许可证）和 Seaborn（BSD 许可证）生成。参数扫描使用 Hydra（MIT 许可证）和 Pydantic（MIT 许可证）进行管理。实验跟踪使用 Weights and Biases（MIT 许可证）。

E.1 数据集

除表 2 外，对于合成数据集，我们使用包含 100,000 个样本的训练数据集，其中有 $[3-50]$ 名选民和 $[2-10]$ 位候选人。验证数据集抽取 1,280 个样本，包含 75 名选民和 15 位候选人。测试数据集抽取 1,280 个样本，包含 100 名选民和 20 位候选人。由于数据集的限制，对于 MovieLens 数据集，我们抽取一个包含 5,000 个样本的训练数据集，其中有 $[10-25]$ 名选民和候选人；验证数据集 $[3-7]$ 的样本量、选民数量和候选人数量未注明；测试数据集包含 512 个样本，具有 45 名选民和 15 位候选人。

MovieLens：MovieLens 32M 数据集 [33] 是由 3200 万条 87,585 部电影的 200,948 名用户评分汇总而成的集合。每部电影的评分范围为 $[0.5, 5]$ ，增量为 5。我们筛选出评分数超过 20,000 的电影，并抽样 m 部电影作为候选集。随后，我们筛选出对候选集中的所有电影都表达偏好的用户，并抽样 n 名投票者。使用许可见： https://files.grouplens.org/datasets/movielens/ml-32m-README.html。

2017 年法国选举（Grenoble）[34]

该数据集来自一项在线及现场调查，调查于2017 年 4 月法国选举第一轮期间开展。我们使用了在 Grenoble 两个投票站收集的数据，投票者被要求用铅笔在一条连续线上做标记，以评估每位候选人。这产生了从 $[0-1]$ 编码的表达偏好。该数据集依据开放数据库许可证提供访问。

我们推测，该数据集中收集的偏好可能包含策略性行为，因为选民往往会优先标记极端立场的候选人。

E.2 网络架构

图选举投票网络（GEVN）

The GEVN 构建的组件多层感知机采用 LayerNorm 归一化[43] 和 ReLU [44] 激活函数 n.

- ψ 是一个 2 层多层感知机，对边嵌入与两个端点节点嵌入的拼接结果进行运算。
- ϕ_v 是一个 2 层多层感知机，对聚合消息与局部嵌入的拼接结果进行运算。
- ϕ_e 是一个 2 层多层感知机，对边嵌入与两个端点节点嵌入的拼接进行处理。

对于小型和标准尺寸的模型，我们分别使用节点嵌入维度 $(58, 185)$ 和边嵌入维度 $(19, 60)$，两者均采用 4 个消息传递层。输入和输出通过线性层转换为所需维度。

DeepSets选举网络

我们使用 PyTorch Geometric 的 DeepSetsAggregation 工具实现 DeepSet选举模型。编码器和解码器均为多层感知机；对于所考虑的小型和大型模型，它们分别包含 3 层和 5 层，嵌入维度分别为 155 和 352。多层感知机中使用 BatchNorm [45] 和 ReLU，聚合函数则采用求和聚合。

表 4：训练期间使用的超参数。

超参数	值
优化器	Adam
学习率调度器（预热）	线性预热（$0.1 \to 1$ 跨越 20 个训练轮次的因子）
学习率调度器（主）	带热重启的余弦退火（$T_0 = 20,\ T_{\text{mult}} = 2$）
学习率	0.0003
梯度范数裁剪	1.0
批量大小	128
单调性损失批量大小	32

图选举策略网络（GESN）

对于 私有信息的情况，我们使用自定义的 DeepSet模型来表示 GESN。v_i 的输入是一个 $\mathbb{R}^m$ 维向量，表示 $\boldsymbol{U}$ 的第 i^{th} 行——选民信息受到限制。在网络的前向传播中，首先通过线性层将 1 维选票转换为 32 维嵌入。随后，在每个 DeepSet层中，网络将嵌入输入一个带有 LeakyReLU [46] 激活函数的 2 层多层感知机，接着拼接对选票嵌入进行求和聚合后的结果，再输入另一个 2 层多层感知机。我们使用两个这样的 DeepSet层。最后，将输出嵌入与原始嵌入拼接形成残差连接，再输入一个 3 层多层感知机，以获得策略性投票。该架构总共有 13,729 个参数。

否则，我们使用与 GEVN 类似的消息传递神经网络。

E.3 训练超参数

在所有投票者—候选人对上计算单调性损失所需的二阶微分，计算成本很高。在训练过程中，我们在每个小批量中对候选人进行子采样，以计算单调性损失；通过实证发现，这能够提供足够稳定的训练信号。

表 4 展示了训练过程中使用的其他超参数值：

E.4 使用的计算机资源

所有实验均在英伟达图形处理器上运行。

福利最大化实验在 NVIDIA RTX 4080 图形处理器上进行，该图形处理器配备 16GB 显存和 10,240 个 CUDA 核心。设备使用 AMD RYZEN 9 7900X 中央处理器，配备 12 个核心和 32GB 内存。每次单独运行耗时 1–3 分钟，我们在参数扫描过程中并行执行各次运行。

关于学习现有投票规则和策略性投票的实验，均在一块配备 24GB 显存和 10,496 个 CUDA 核心的 NVIDIA RTX 3090 图形处理器上运行。该设备搭载 Intel i5-13600KF 中央处理器，拥有 14 个核心和 32 GB 内存。每次单独运行需要 $3-30$ 分钟，我们会在参数扫描中并行执行各次运行。