arXiv:2603.05673v1 [cs.LG] 5 Mar 2026

用于潮流网络分析的强化学习

Alperen A. Erg¨ur[1], Julia Lindberg[2], 和 Vinny Miller[3]

[1]德克萨斯大学圣安东尼奥分校数学与计算机科学系[2]佐治亚理工学院数学系[3]德克萨斯大学圣安东尼奥分校计算机科学系

摘要

潮流方程是非线性多元方程，用于描述电力网络中功率注入与母线电压之间的关系。给定网络拓扑，我们希望找到具有多个平衡点的网络参数。这相当于寻找具有多个实数解的潮流方程实例。当前最先进的计算代数算法，尚无法解答涉及变量数量超过少数几个的网络中的这一问题。为解决这一问题，我们设计了一种概率奖励函数，可较好地近似根数，并构造了一个仿照潮流方程空间的状态空间。我们推导了高斯模型的平均根数，并将其用作强化学习智能体的基准。智能体发现的潮流方程实例，其解的数量远超平均基准。这表明，强化学习在潮流网络设计与分析方面具有潜力，也展现了强化学习为涉及复杂非线性代数或几何学的问题作出实质贡献的潜力。 [1]

1 引言

1.1 电力网络及其非线性方程

电力网络与潮流问题

现代工程最伟大的成就之一，是在全球各地实现了城市电气化。这是通过建设和维护电网实现的。潮流方程是一组描述电力在电网中传输时行为的非线性方程，在所有电力网络研究中都无处不在。潮流方程的实数解给出了电网的运行点。对于一个

[1]作者按字母顺序排列，所有作者贡献相同。

给定的电力网络，寻找潮流方程所有实数解的问题称为潮流问题。

在现实场景中，工程师通常只需找到潮流方程的一个实数解。然而，在某些类型的问题中，找到多个解更为理想。其中一个重要问题是稳定性分析，通常也称为动态安全评估。电力系统的动态安全评估（DSA）用于评估系统在特定条件下承受所有可信故障的能力。对于直接动态安全评估方法，潮流方程的多个实数解对应于底层动力系统的平衡点。这些平衡点分为稳定平衡点（SEP）和不稳定平衡点（UEP）。

稳定性通常通过小信号线性化模型进行评估，该模型仅适用于运行点附近的一个小区域。相比之下，基于能量函数的直接法将大信号稳定性分析扩展至稳定平衡点（SEP）周围更广的区域。这些方法用于确定动态轨迹是否始终处于 SEP 的吸引域内。值得注意的是，该区域的边界由附近不稳定平衡点（UEP）的稳定流形定义， [35, 45, 49], 而特定的 UEP 则用于计算稳定性所需的能量阈值。

潮流方程

一个具有 n 个节点的电力网络被建模为无向图 $G = (V, E)$，其中每个顶点 $v_k \in V$ 表示电力网络中的一个节点（母线）。若电力网络中对应的节点相连，则顶点 v_k 与 v_m 之间存在一条边 e_{km}。

每个顶点 $v_k \in V$ 都有一个相关联的复功率注入 $P_k + \mathbf{i}Q_k$，每条边$e_{km} \in E$ 都有一个已知的复导纳 $b_{km} + \mathbf{i}g_{km}$，其中 $\mathbf{i} = \sqrt{-1}$。若顶点 v_k 与 v_m 之间没有边，则将导纳取为零。

基于基尔霍夫定律和欧姆定律，潮流方程在电网的每个节点上平衡电力的供需。具体而言，在每个节点 $v_k \in V$ 处，有功（实）功率 P_k 与无功（虚）功率 Q_k 之间的关系由如下非线性方程刻画

$$P_k = \sum_{m=0}^{n-1} |V_k||V_m|(g_{km}\cos(\theta_k - \theta_m) + b_{km}\sin(\theta_k - \theta_m)) \tag{1}$$

$$Q_k = \sum_{m=0}^{n-1} |V_k||V_m|(g_{km}\sin(\theta_k - \theta_m) - b_{km}\cos(\theta_k - \theta_m)) \tag{2}$$

其中， V_k 是复电压幅值， θ_k 表示节点 v_k处的复电压相角。

在传统的潮流研究中，假设已知四个参数中的两个（P_k， Q_k， V_k， θ_k），并求解其余参数。具体来说，有三种类型的节点决定了所考虑的方程：

1. **PV：** 假定已知有功功率注入 P_k 和电压幅值 V_k，而无功功率注入 Q_k 和电压相角 θ_k 未知。这些表示发电机母线。

2. **PQ：** 假定已知有功功率和无功功率注入 P_k，Q_k 而电压幅值 V_k 和电压相角 θ_k 未知。这些表示负荷母线。

3. **平衡节点：** 电压相角和幅值已知，而有功功率和无功功率注入未知。我们始终假设每个电网都有一个平衡母线，并且不失一般性地，令其为 v_0，并假设 $\theta_0 = 0$ 和 $|V_0| = 1$。

为简化起见，我们将研究范围限定为所有节点均为PV节点且有一个平衡母线的情况，不过本文提出的方法也适用于PQ节点。实际中 $g_k \ll b_k$，因此我们假设 $g_k = 0$。这称为无损电网。通过考虑变量替换 $x_k = |V_k|\cos(\theta_k)$和 $y_k = |V_k|\sin(\theta_k)$，我们将(1)-(2)转化为代数方程。

通过对变量 x_k，y_k 重新缩放，我们可以假设对于所有 $k = 0, \ldots\ldots, n-1$，$V_k = 1$。通常我们假设平衡母线 v_0 的$(x_0, y_0) =$ 为(1，0)。在我们的情形中，我们将稍微放宽这一假设，并假设在 v_0 处有如下方程:

$$1 = x_0^2 + y_0^2, \qquad 1 = x_0^2.$$

对于 $k = 1, \ldots, n-1$，电网 G 的潮流方程是由 $2n$个方程组成的方程组，包含 $2n$ 个未知数 $(x_0, \ldots, x_n, y_0, \ldots, y_n)$:

$$P_k = \sum_{m=0}^{n-1} b_{km}(x_m y_k - x_k y_m) \tag{3}$$

$$1 = x_k^2 + y_k^2 \tag{4}$$

$$1 = x_0^2. \tag{5}$$

$$1 = x_0^2 + y_0^2 \tag{6}$$

对于 $k = 1, \ldots, n$，方程 (3)-(6) 是潮流方程，而潮流方程的实数解是电力网络的运行点。

我们声称，(3)-(6) 可以表示为 $\mathbb{R}^{2n}$ 中 $2n$ 个椭球面的交集。考虑变量向量 $z = (x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1})$。那么，(3)-(6) 可以重写为 $z^T A_k z = 1$，对所有 $1 \le k \le 2n$ 均成立。关于这一构造的详细参考资料见 [23]，下面我们简要概述。

由于 (3)-(6) 是齐次二次型，(3) 和 (5) 可以写成 $z^T Q_k z = 1$，其中 Q_k 是对称矩阵；(4) 和 (6) 可以写成 $z^T S_k z = 1$，其中 $0 \le k \le n-1$。

请注意，在这个公式中，S_k 是一个除（$k+1$，$k+1$）和（$k+n$，$k+n$）位置上的元素为 1 外，其余元素均为 0 的矩阵。此外，$S_0 + \ldots + S_{n-1} = I_{2n}$。因此，我们不再使用 $z^T Q_k z = 1$ 和 $z^T S_k z = 1$，而是定义

$$A_k = \alpha_{0,k} S_0 + \ldots + \alpha_{n-1,k} S_{n-1} \qquad \text{if} \qquad 0 \le k \le n-1$$

$$B_k = \beta_{0,k} S_0 + \ldots + \beta_{n-1,k} S_{n-1} + \gamma_{0,k} Q_0 + \ldots + \gamma_{n-1,k} Q_{n-1} \qquad \text{if} \qquad n \le k \le 2n-1,$$

其中，α_{ij}，$\gamma_{ij} \in \mathbb{R}_{>0}$ 和 $\beta_{ij} \in \mathbb{R}$ 的选取应满足每个 A_k，$B_k \succ 0$，且它们定义的线性变换是满秩的。

在这种设置下，矩阵 A_k，B_k 具有相当多的结构。具体来说，每个 A_k 都是对角矩阵，其（i，i）和（$n+i-1$，$n+i-1$）位置上的元素值相同。类似地，每个 B_k在（i，i）和（$n+i-1$，$n+i-1$）位置上的元素值也相同。此外，当且仅当存在一条边 $e_{i+1,\ j+1} \in E$时，每个 B_k 在（i，j）和（j，i）位置上才有元素。

例1.1。 举例来说，我们考虑一个极度简化的电网：一棵包含三个节点 v_0、 v_1、 v_2 的树，其中边 (v_0, v_1) 的权重为 b_{01}，边 (v_1, v_2) 的权重为 b_{12}。

这里的潮流方程为

$$\begin{aligned} P_1 &= b_{01}(x_0y_1 - x_1y_0) + b_{12}(x_2y_1 - x_1y_2) \\ P_2 &= b_{12}(x_1y_2 - x_2y_1) \\ 1 &= x_1^2 + y_1^2 \\ 1 &= x_2^2 + y_2^2 \\ 1 &= x_0^2 \\ 1 &= x_0^2 + y_0^2 \end{aligned}$$

这些方程的矩阵形式见 *C* 节。这些矩阵的任意线性组合都会得到一个具有以下稀疏模式的矩阵：

$$\begin{bmatrix} * & 0 & 0 & 0 & * & 0 \\ 0 & * & 0 & * & 0 & * \\ 0 & 0 & * & 0 & * & 0 \\ 0 & * & 0 & * & 0 & 0 \\ * & 0 & * & 0 & * & 0 \\ 0 & * & 0 & 0 & 0 & * \end{bmatrix}$$

1.2 问题陈述与相关挑战

现在，我们可以简洁地陈述目标问题：给定一个电网 $G = (V, E)$，求一组 n 矩阵（$A_1, A_2, \cdots, A_n$），使方程组 (7) 的实数解数量最多。

$$\|A_1\|_2^2 = \|A_2\|_2^2 = \cdots = \|A_n\|_2^2 = 1 \tag{7}$$

解决这一问题主要面临两大挑战：

1. 最先进的计算代数软件在 n 方面的扩展性很差，这使得实验能力仅限于规模较小的实例。也就是说，对于给定的方程实例，只有当 n 较小时，我们才能求解。

2. 即使对于较小的 n，计算代数领域最先进的技术在探索潮流方程空间时表现也不佳。因此，似乎无法用这些技术来解决上述问题。

本文通过设计一种概率奖励函数来解决第一个技术障碍。该函数以数学上严谨的方式近似实数解的数量，且比计算代数算法更具可扩展性。本文的核心主张是，强化学习为应对第二个挑战提供了一个强有力的框架。

1.3 我们的贡献

- 我们开发了首个基于机器学习的潮流方程建模方法。具体而言，我们将寻找具有多个解的潮流方程表述为一个强化学习问题。

- 我们分析了这些方程的平均情形行为，作为基线。此前，这一平均值尚不为人所知。

- 我们设计了一种新颖的概率奖励函数，能够严格近似潮流方程的根的数量。目前最先进的计算代数方法只能处理非常小规模的实例，而我们的方法可并行化且具有良好的可扩展性。

- 实验结果表明，强化学习智能体能够发现解的数量远超通常预期的方程组。除了在潮流网络分析中的直接应用外，这一成果还展示了强化学习解决非线性代数和几何学复杂问题的潜力。具体而言，实代数几何领域中有许多尚未解决的猜想，如今正适合通过强化学习进行检验

1.4 相关工作

据我们所知，文献中尚无基于机器学习、针对我们所研究问题的方法。因此，我们回顾机器学习在潮流系统设计与建模中更广泛任务上的应用。随后，我们回顾计算数学领域开发的方法。

1.4.1 电力系统中的机器学习

机器学习方法，包括监督学习、无监督学习和深度学习，已用于负荷与可再生能源发电预测 [18, 39], 状态估计 [11], 故障检测 [49], 稳定性评估 [32], 以及最优潮流近似 [46]。更全面的综述请参见 [19]。

强化学习（RL）尤其已成为电力系统序贯决策问题的一种颇具前景的框架 [16]。通过将电网运行建模为马尔可夫决策过程，强化学习能够为电压调节 [51], 频率控制 [1], 需求响应 [47] 和储能管理 [7]等任务提供数据驱动控制策略。随着可再生能源并网和分布式能源资源的发展，不确定性日益增加；在这种情况下，对所有系统动态进行显式建模可能难以实现，因此强化学习尤其具有吸引力。然而，由于安全约束、样本效率低，以及需要在罕见但关键的故障情况下保持稳健性能，实际部署仍面临挑战。因此，近期研究主要聚焦于安全强化学习、约束感知形式，以及将强化学习与基于优化的控制器和领域知识相结合，以确保现实电网运行中的稳定性、可靠性和可解释性。有关电力系统安全强化学习的综述，请参见 [41] 。

1.4.2 平均情形下实根的计数

对多项式系统实根期望数量的研究始于 20 世纪 90 年代。虽然贝祖定理等经典结果告诉我们，一个由 n 个次数为 d_1、.....、d_n 的多项式方程组成的系统，最多可以有 $d_1 \cdots d_n$ 个孤立复解，但

实数解的数量通常远低于这一上限。在 [40] 中，作者证明，当多项式方程的系数是相互独立、均值为零的高斯随机变量时，实数解的期望数量为 $\sqrt{d_1 \cdots d_n}$。此后，这项分析被拓展到稀疏多重齐次多项式系统的情形 [37, 29]。

由于潮流方程具有一般多项式系统所不具备的额外结构（例如，共享系数），上述结果并不适用于我们的问题。已有一些实证研究探讨潮流方程实数解的期望数量 [50, 26], 以及可产生固定数量实数解的网络参数空间 [25, 22]。

1.4.3 通过求根分析稳定性

识别吸引域边界上的关键不稳定平衡点（UEP）一直是一项挑战。为解决这一问题，研究人员提出了多种技术来求取潮流方程的多个解。一种广泛使用的方法是牛顿-拉夫逊算法，通过采用不同的起始点运行该算法来寻找各种平衡状态。然而，这些迭代方法的收敛性并不总是可靠，尤其是考虑到收敛区域可能具有分形特性 [43]。尽管寻找某些 UEP 的方法已有所改进 [44, 10, 9, 27, 21, 20],，但这些方法并不全面，也无法严格地 找出所有 UEP。

确定所有所需 UEP 的唯一可证明方法，是计算所有平衡点，即求出潮流方程的所有实数解。Homotopy 和追踪算法已被应用 [33, 42, 28, 31, 23, 38, 48, 30, 26], ，但一般而言，这些方法的扩展性较差[2]。最近，单值化算法已针对潮流方程进行了调整 [24]，并计算出一个含有 20 个节点的电网的所有解。这是目前已能以可证明方式找到所有解的最大规模现实场景。对于一般电网，寻找单值性方法的停止准则依赖于计算混合体积 [8], ，而这属于 # P 难问题 [13]。

同伦方法效率低下的一个根本原因是，这些算法必须先计算出所有复根，再从中筛选出实根。除了实根的期望数量远小于复根数量之外，人们还经常观察到，在整个参数空间中，拥有远超预期数量实根的参数实例所占比例非常小。因此，寻找具有大量实根的参数实例具有普遍意义。爬山法是解决这一问题的一种方法。爬山法是一种局部搜索技术，它在参数空间中迭代移动到实数解更多的参数值。 [12] 这种贪心方法最初是为运动学问题开发的，但在计数代数几何问题中也取得了成功 [6, 5]。爬山算法属于局部搜索启发式方法，因此对初始化高度敏感，并且很容易陷入对应于实数解较少配置的次优局部最大值。

[2]作者在 [28] 提出了一种算法，其运行规模随实数解的数量线性增长。然而，后来研究表明，该算法可能无法找到所有实数解 [33]。

2 平均情形分析

本节将给出平均情形行为的数学推导，并为读者理解下一节的概率奖励函数设计做好准备。本节采用的随机性模型如下： A_1， A_2， $\cdots$， A_n 是随机高斯矩阵，其独立元素的均值为零，方差为 σ^2。以下方程组的解的期望数量是多少?

$$\|A_1 x\|^2 = \|A_2 x\|_2^2 = \cdots = \|A_n x\|_2^2 = 1$$

这一估计将作为我们寻找具有许多实根的配置这一工作的基线。令人惊讶的是，尽管关于潮流方程已有大量研究，这一结果此前并不为人所知。 [2,定理 6.7]。

1. 我们有随机场 $Z(t) \in \mathbb{R}^d$，它构造为 $Z(t) = H[Y(t)]$，其中

$$\{Y(t) : t \in W\}$$

是一个以集合 $W \in \mathbb{R}^d$ 为指标、取值为 $Y(t) \in \mathbb{R}^n$ （对某个$n \geq d$）的高斯场，并且 $H: \mathbb{R}^n \to \mathbb{R}^d$ 是连续可微函数。

2. $Z(t)$ 具有密度 $p_{Z(t)}(x)$，该密度是数对 $(t, x) \in W \times \mathbb{R}^d$ 的连续函数

3. $\mathbb{P}\{\exists t \in W: \ Z(t) = u, \ \det(Z'(t)) = 0\} = 0$

我们的随机场 Z 定义如下：首先，我们定义 $Y(x)$ 如下：

$$Y : \mathbb{R}^n \to \mathbb{R}^{n^2}, \ Y(x) = (A_1 x, A_2 x, \ldots, A_n x)$$

然后，定义映射 H 如下：

$$H : \mathbb{R}^{n^2} \to \mathbb{R}^n, \ H(y_1, y_2, \ldots, y_n) = (\|y_1\|^2, \ldots, \|y_n\|^2)$$

其中 $\|.\|$ 表示标准欧几里得范数。于是，随机场定义为$Z(x) := H \circ Y(x)$。解的期望数量由以下公式给出：

$$\mathbb{E}\left(N(Z, \mathbb{R}^n)\right) = \int_{\mathbb{R}^n} \mathbb{E}\left(|det(Z'(x))| \mid Z(x) = (1, 1, \cdots, 1)\right) \ p_{Z(x)}(1, 1, \cdots, 1) \ dx \qquad (8)$$

现在我们需要展开密度$p_{Z(x)}(u)$：

$$p_{Z(x)}(1, 1, \ldots, 1) = p_{H(Y(x))}(1, 1, \ldots, 1) = P_{H(Y(\|x\|e_1))}(1, 1, \ldots, 1)$$

其中第二个等式是由高斯场 Y 的旋转不变性得出的。现在注意到 A_1， A_2，.....， A_n 相互独立，因此我们有

$$P_{H(Y(\|x\|e_1))}(1, 1, \ldots, 1) = P_{H(Y\|x\|e_1)}(1, \ldots, 1) = \left(P_{\|x\|^2 \|A_1 e_1\|^2}(1)\right)^n$$

随机变量 $\|x\|^2\|A_1e_1\|^2 = \|x\|^2(\psi_1^2 + \ldots + \psi_n^2)$，其中 $\psi_i \sim \mathcal{N}(0,\ \sigma^2)$，且 $\frac{\|x\|^2\|A_1e_1\|^2}{\|x\|^2\sigma^2}$ 服从 χ_n^2 分布。因此我们有

$$P_{\|x\|^2\|A_1e_1\|^2}(1) = P_{\chi_n^2}(\frac{1}{\sigma^2\|x\|^2})\frac{1}{\|x\|^2\sigma^2}$$

and

$$p_{Z(x)}(1,1,\ldots,1) = \left(\frac{1}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})}(\frac{1}{\sigma^2\|x\|^2})^{\frac{n}{2}-1}e^{-\frac{1}{2\sigma^2\|x\|^2}}\frac{1}{\|x\|^2\sigma^2}\right)^n. \tag{9}$$

为了估计该公式，我们需要计算 $Z'(x)$。为方便起见，我们定义如下映射，将 $x \in \mathbb{R}^n$ 映射为一个 $n \times n$ 矩阵 $A(x)$:

$$A : \mathbb{R}^n \to \mathbb{R}^{n\times n}\ ,\ A(x)^T = [A_1^TA_1x, A_2^TA_2x, \ldots, A_n^TA_nx]$$

于是，我们有

$$Z'(x) = 2A(x)\ ,\ Z(x) = A(x)x$$

现在，对于正交矩阵 Q，令 $y = Qx$，定义 $B_i = A_iQ$，并将此定义扩展为映射 $B(x)$ 的定义，如下:

$$B(x)^T = [B_1^TB_1x, \ldots, B_n^TB_nx].$$

于是，我们有 $A(y)y = B(x)x$，但 $A(y) \neq B(x)$。不过，我们注意到，$B(x)$ 的每一行都是通过正交矩阵 Q 作用于 $A(y)$ 的各行得到的。因此，自然有 $\det(B(x)) = \det(A(y))$。现在注意， A_i 的行和列都是中心化高斯随机向量，因此，由于高斯分布具有旋转不变性， A_i 的分布在 Q 的作用下不会改变。这意味着:

$$\mathbb{E}\left(|det(Z'(x))| \mid Z(x) = (1,1,\ldots,1)\right) = \mathbb{E}\left(|det(Z'(y))| \mid Z(y) = (1,1,\ldots,1)\right)$$

还要注意，密度公式在 Q 的作用下也保持不变。这意味着，采用任意选择 $x = e_0$，我们可以将其简化如下:

$$|S^{n-1}|\int_0^\infty \mathbb{E}\left(|det(Z'(te_1))| \mid Z(te_1) = (1,\ldots,1)\right)\ p_{Z(te_1)}(1,\ldots,1)t^{n-1}\,dt \tag{10}$$

其中 $|S^{n-1}|$ 表示球面的体积 S^{n-1}。注意

$$Z(te_1) = (1,\ldots,1) \Leftrightarrow Z(e_1) = (\frac{1}{t^2},\ldots,\frac{1}{t^2}) = \left((A_iA_i^T)_{11}\right)_i, \text{ and } Z'(te_1) = tZ'(e_1).$$

因此,

$$\mathbb{E}\left(|det(Z'(te_1))| \mid Z(te_1) = (1,\ldots,1)\right) = t^n\,\mathbb{E}\left(|det(Z'(e_1))| \mid Z(e_1) = (\frac{1}{t^2},\ldots,\frac{1}{t^2})\right)$$

设矩阵 A_i 的列为 a_{ij}，其中 $j = 1,\ 2,\ \ldots,\ n$。假设 $Z(e_1) = (\frac{1}{t},\ \ldots,\ \frac{1}{t^2})$ 等价于对所有 i 都有 $\|a_{i1}\| = \frac{1}{t}$。此外，我们有

$$Z'(e_1) = 2\begin{bmatrix} \langle a_{11}, a_{11}\rangle & \langle a_{11}, a_{12}\rangle & \ldots & \langle a_{11}, a_{1n}\rangle \\ \vdots & & & \vdots \\ \langle a_{n1}, a_{n1}\rangle & \langle a_{n1}, a_{n2}\rangle & \ldots & \langle a_{n1}, a_{nn}\rangle \end{bmatrix}$$

也就是说，假设 $Z(e_1) = (\frac{1}{t}, \ldots, \frac{1}{t^2})$ 相对于 $Z(e_1) = (1, \ldots, 1)$，会按 $\frac{1}{t^2} \cdot \frac{1}{t^{n-1}} = \frac{1}{t^{n+1}}$ 缩放行列式。因此，

$$\mathbb{E}\left(|det(Z'(te_1))| \mid Z(te_1) = (1, \ldots, 1)\right) = t^{-1}\mathbb{E}\left(|det(Z'(e_1))| \mid Z(e_1) = (1, \ldots, 1)\right).$$

因此，我们可以进一步简化为

$$|S^{n-1}|\ \mathbb{E}\left(|det(Z'(e_1))| \mid Z(e_1) = (1, \ldots, 1)\right) \int_0^\infty t^{n-2} p_{Z(te_1)}(1, \ldots, 1)\, dt \tag{11}$$

我们注意到，用同一个常数缩放所有方程不会改变根数，因此从这里开始，不失一般性地假设 $\sigma = 1$。

命题 2.1。

$$|S^{n-1}|\ \mathbb{E}\left(|det(Z'(e_1))| \mid Z(e_1) = (1, \ldots, 1)\right) \sim \pi^{\frac{n-2}{2}} 2^{\frac{3n+1}{2}}$$

证明。以 $Z(e_1) = (1, 1, \ldots, 1)$ 为条件，等价于假设

$$\|A_1 e_1\| = \|A_2 e_1\| = \cdots = \|A_n e_1\| = 1$$

注意，$\|A_i e_1\|^2 = e_1^T A_i^T A_i e_1$，这就是 A_i 第一列的范数。因此，我们所取的条件是所有矩阵的第一列范数均为 1，这等价于假设各第一列是单位球面上的独立均匀随机向量。现在注意到

$$Z'(e_1) = 2[A_1^T A_1 e_1 A_2^T A_2 e_1 \cdots A_n^T A_n e_1]$$

注意到，向量 $A_i^T A_i e_1$ 是在其第一坐标为 1 的条件下得到的，其余坐标是由均匀分布在球面上的向量与高斯向量相乘得到的。也就是说，该向量的第一坐标为 1，其余坐标是均值为零、方差为一的独立高斯变量。相应地，矩阵

$$[A_1^T A_1 e_1 \ A_2^T A_2 e_1 \ \cdots \ A_n^T A_n e_1]$$

的第一列固定为 $(1, 1, \cdots, 1)$，其余各列为独立高斯向量。注意，由于海达玛行列式公式

$$|\det\left([A_1^T A_1 e_1 A_2^T A_2 e_1 \cdots A_n^T A_n e_1]\right)| = \sqrt{n}|\det(PX)|$$

其中，P 是向量 $(1, 1, \cdots, 1)$ 的正交补上的投影，而 X 是 $n \times (n-1)$ 高斯矩阵，其元素相互独立且方差为 1。$\mathbb{E}|\det(PX)|$ 的期望是已知的（引用），因此我们有

$$\mathbb{E}|\det(PX)| \sim \frac{2^{\frac{n-2}{2}}\Gamma(\frac{n-1}{2})}{\sqrt{\pi}}$$

请注意

$$|S^{n-1}| = \frac{2\pi^{\frac{n-1}{2}}}{\Gamma(\frac{n}{2})}$$

因此，我们有

$$|S^{n-1}|\mathbb{E}\left(|det(Z'(e_1))| \mid Z(e_1)=(1,\dots,1)\right)=2^n\sqrt{n}\pi^{\frac{n-2}{2}}2^{\frac{n}{2}}\frac{\Gamma(\frac{n-1}{2})}{\Gamma(\frac{n}{2})}$$

$$\frac{\Gamma(\frac{n-1}{2})}{\Gamma(\frac{n}{2})}\sim\frac{1}{\sqrt{\frac{n-1}{2}}}$$

□

利用这一论断，我们可以证明以下定理。

定理 2.2. 设 $A_1, A_2, \dots, A_n$ 为具有独立元素且标准差为 $\sigma>0$ 的随机高斯矩阵。那么，以下方程组的解的期望数量

$$\|A_1x\|^2=\|A_2x\|^2=\ldots=\|A_nx\|^2=1$$

is

$$c_1n^{-\frac{1}{2}}2^{\frac{n}{2}}.$$

定理 2.2 中的估计将作为我们探索智能体的基准。利用命题 2.1，定理 2.2 的证明如下完成:

$$p_{Z(te_0)}(1,\dots,1)=\left(2^{\frac{n}{2}}\Gamma(\frac{n}{2})\right)^{-n}t^{-n^2}e^{-\frac{n}{2t^2}}$$

$$\int_0^\infty t^{n-2}p_{Z(te_0)}(1,\dots,1)\,dt=\left(2^{\frac{n}{2}}\Gamma(\frac{n}{2})\right)^{-n}\int_0^\infty t^{-2}t^{-n^2+n}e^{-\frac{n}{2t^2}}\,dt$$

对于 $u=\frac{1}{t}$，$du=-\frac{1}{t^2}\,dt$，我们有

$$\int_0^\infty t^{-2}t^{-n^2+n}e^{-\frac{n}{2t^2}}\,dt=\int_0^\infty u^{n^2-n}e^{-\frac{nu^2}{2}}\,du$$

$$\mathbb{E}\left(N_u(Z,\mathbb{R}^n)\right)\sim\pi^{\frac{n-2}{2}}2^{\frac{3n+1}{2}}\left(2^{\frac{n}{2}}\Gamma(\frac{n}{2})\right)^{-n}\int_0^\infty u^{n^2-n}e^{-\frac{nu^2}{2}}\,du$$

现在注意，右侧的积分表达式几乎等价于 $z\sim\mathcal{N}(0,\frac{1}{n})$ 的 n^2-n 阶矩。利用已知的高斯矩公式，可得

$$\int_0^\infty u^{n^2-n}e^{-\frac{nu^2}{2}}\,du=2^{\frac{n^2-n-1}{2}}n^{-\frac{n^2-n+1}{2}}\Gamma(\frac{n^2-n+1}{2})$$

因此，我们有

$$\mathbb{E}\left(N_u(Z,\mathbb{R}^n)\right)\sim\pi^{\frac{n-2}{2}}2^n\Gamma(\frac{n}{2})^{-n}n^{-\frac{n^2-n+1}{2}}\Gamma(\frac{n^2-n+1}{2})$$

回顾一下（斯特林）：

$$\ln\left(\Gamma(z)\right)\sim(z-\frac{1}{2})\ln(z)-z+\frac{1}{2}\ln(2\pi)+\frac{1}{12z}$$

我们用它来估计

$$n\ln(2)+\frac{n-2}{2}\ln(\pi)-n\ln(\Gamma(\frac{n}{2}))+\ln(\Gamma(\frac{n^2-n+1}{2}))-\frac{n^2-n+1}{2}\ln(n)$$

于是我们得到

$$\frac{n+1}{2}\ln(2)-\frac{1}{2}\ln(n)-\frac{1}{2}\ln(\pi)+\frac{1}{12}$$

这意味着,

$$\mathbb{E}\left(N_u(Z,\mathbb{R}^n)\right)\sim n^{-\frac{1}{2}}2^{\frac{n+1}{2}}\frac{e^{\frac{1}{2}}}{\sqrt{\pi}} \tag{12}$$

3 奖励函数设计

对于大量变量的二次方程组，计算其实根数量超出了当前最先进的计算代数算法的能力范围；这才是智能体真正要获得的奖励。利用上一节中的工具，我们可以设计出如下严格的近似奖励：假设我们要计算根的数量的方程组为 $(A_1,\ A_2,\ \cdots,\ A_n)$，~我们定义一组随机矩阵 $A_i=A_i+\delta X_i$，其中 $0<\delta<1$ 是一个插值参数，而 X_i 是元素均值为零、方差为一的随机高斯矩阵。如果我们能够估计解的期望数量

$$\|\tilde{A_1}x\|_2^2=\|\tilde{A_2}x\|_2^2=\cdots=\|\tilde{A_n}x\|_2^2=1$$

那么这就可以作为一种有原则的代理。这正是我们所做到的，但这需要若干数学洞见和实践技巧。本节将更多地讨论数学，而第4.2节将更多地讨论算法。我们先陈述一个来自 [3, 定理 4.5.9] 的工具：

引理3.1。 设 $M_1,M_2,\ldots,M_n$ 为正定矩阵，则存在一个可逆矩阵 Z 和正标量 $\alpha_1,\alpha_2,\ldots,\alpha_n$，使得 $T_i=\alpha_i Z^T M_i Z$ 满足

$$\mathrm{Trace}(T_i)=1\ for\ each\ i=1,2,\ldots,n,\ and\ T_1+T_2+\ldots+T_n=\mathbb{I}$$

我们关注以下方程组的实数解的数量：

$$\|A_1x\|^2=\|A_2x\|^2=\ldots=\|A_nx\|^2=1\text{ for }i=1,2,\ldots,n. \tag{13}$$

显然，方程（13）的解的数量等于方程（14）的解的数量

$$\|A_1Zx\|^2=\|A_2Zx\|^2=\ldots=\|A_nZx\|^2=1\text{ for }i=1,2,\ldots,n. \tag{14}$$

对于任意可逆矩阵 Z。还要注意，同时对所有方程进行缩放不会改变解的数量。利用引理3.1，并注意到迹$(Z^TA_i^TA_iZ)=\|A_iZ\|_F^2$，我们可以找到一个 Z 使得

$$\frac{1}{\|A_1Z\|_F^2}Z^TA_1^TA_1Z+\frac{1}{\|A_2Z\|_F^2}Z^TA_2^TA_2Z+\ldots+\frac{1}{\|A_nZ\|_F^2}Z^TA_n^TA_nZ=\mathbb{I}$$

注意，对于任意矩阵 A ，求解

$$\|Ax\|^2 = 1 \text{ is equivalent to solving } \|\frac{1}{\|A\|}Ax\|^2 = \frac{1}{\|A\|^2}$$

因此，我们定义 $C_i = \frac{1}{\|A_iZ\|}A_iZ$ 方程（13）。

$$\|C_i\| = 1 \text{ for each } i, \text{ and } C_1^TC_1 + C_2^TC_2 + \ldots + C_n^TC_n = \mathbb{I}$$
$$c_i = \frac{1}{\|A_iZ\|^2} \text{ for } i = 1, 2, \ldots, n$$
$$\text{Count } x \in \mathbb{R}^n \text{ such that } \|C_ix\|^2 = c_i$$

现在，为了简化计算，我们进行如下操作:

1. 计算 $g := \left(\frac{1}{\|A_1Z\|^2} + \frac{1}{\|A_2Z\|^2} + \cdots + \frac{1}{\|A_nZ\|^2}\right)^{\frac{1}{2}}$。

2. 通过设定 $x = \frac{\sqrt{n}}{g}x$对变量和方程进行变量缩放。之后，形如 $\|Ax\|^2 = \beta$ 的方程变为 $\|Ax\|^2 = \frac{n\beta}{g^2}$。

总之，我们定义了 $C_i = \frac{1}{\|A_iZ\|}A_iZ$，$g = \left(\sum_{i=1}^n \frac{1}{\|A_iZ\|^2}\right)^{\frac{1}{2}}$。现在，方程组变为:

$$\|C_i\| = 1 \text{ for each } i, \text{ and } C_1^TC_1 + C_2^TC_2 + \ldots + C_n^TC_n = \mathbb{I}$$
$$(c_1, c_2, \ldots, c_n) \in (\mathbb{R}_+)^n \ , \ c_i = \frac{n}{g^2\|A_iZ\|^2} \ . \ c_1 + c_2 + \cdots + c_n = n \tag{15}$$
$$\text{Count } x \in \mathbb{R}^n \text{ such that } \|C_ix\|^2 = c_i.$$

我们的随机性模型设置如下：我们使用 $c = (c_1, c_2, \ldots\ldots, c_n)$表示我们的$n$元组，其中 $c_i > 0$ 和 $c_1 + c_2 + \ldots + c_n = n$，并使用 $\mathcal{C} = (C_1, C_2, \ldots\ldots, C_n)$，使得 $C_1^TC_1 + C_2^TC_2 + \ldots + C_n^TC_n = \mathbb{I}$。于是，我们得到定义如下的随机高斯矩阵 A_i

$$A_i = C_i + \delta X_i \text{ where } X_i \ , \ \|A_1x\|^2 = c_1 \ , \ \|A_2x\|^2 = c_2 \ , \cdots \ , \ \|A_nx\|^2 = c_n \tag{16}$$

其中 X_i具有相互独立的 $\mathcal{N}(0, 1)$ 条目。我们将使用方程 (16) 的解的期望数量，作为精确计算方程 (15) 的实数解数量的代理。现在假设我们有方程 (16) 的解 $x \in \mathbb{R}^n$ ，那么

$$n = \sum_{i=1}^n x^TA_i^TA_ix = \|x\|^2 + x^T\left(\delta\sum_{i=1}^n(X_i^TC_i + C_i^TX_i) + \delta^2\sum_{i=1}^n X_i^TX_i\right)x \tag{17}$$

注意， $X := \delta\sum_{i=1}^n(X_i^TC_i + C_i^TX_i) + \delta^2\sum_{i=1}^n X_i^TX_i$ 是一个实对称矩阵，并且我们有 $n = \|x\|^2 + x^TXx$。此外，由于高斯分布具有非常强的集中性质， $X \sim \delta^2 n \ \mathbb{I}$ 以高概率成立。因此，我们以高概率有

$$n \sim (1 + \delta^2 n)\|x\|^2. \tag{18}$$

现在，我们可以概述一下实现此奖励函数的方案：引理3.1可以表述为一个凸优化程序，并使用拟牛顿法高效求解。完成缩放后，我们将对与方程（16）对应的 Kac-Rice公式进行蒙特卡洛近似，其中 x 将从满足 $n(1-\delta) \leq \|x\|^2 \leq n(1+\delta)$ 的区域中抽样。在给定样本 x 处计算 Kac-Rice公式中的条件期望较为繁琐，因此我们将结合重要性抽样，采用 [14] 中的一个技巧来简化计算。这些步骤将在第4.2节中说明。

4 强化学习设置

上一节已建立概率根计数代理的理论基础，接下来的核心挑战是如何有效地探索潮流方程的复杂参数空间。通过将系统矩阵映射到环境的状态空间，并将允许的扰动界定为动作，智能体逐步调整网络参数，以最大化实数解的期望数量。本节余下部分将详细介绍用于探索该参数空间的演员-评论家架构，以及奖励函数的具体计算实现。

4.1 问题表述与演员-评论家架构

我们将寻找具有大量实数解的潮流方程表述为一个强化学习问题，并采用双延迟演员-评论家架构 [15]。随后，我们通过使用 Julia Homotopy [4] 对小规模实例进行实验，验证所得结果。强化学习环境定义如下

- 状态空间：状态空间建模为所有 $n \times n$ 矩阵的集合，其元素取值范围为 $[-1, 1]$。这种表示是有效的，因为将潮流方程中所有矩阵的所有元素都乘以相同的标量，不会改变其解。

- 动作空间：虽然动作空间在理论上与状态空间相同，但为确保稳健更新，动作被限制为幅度有上限的扰动。具体而言，智能体在单步中对矩阵元素的修改幅度至多为$\hat{}a$ 。

- 奖励：在每一步，智能体输出一个矩阵系统 $\mathcal{A} = （A_1，...，A_n）$。随后通过在该系统上运行概率的蒙特卡洛近似来计算奖励，以估计实数解数量。

- 回合结构：由于奖励函数依赖于近似，很难定义严格的终止条件。因此，回合长度被视为一个超参数。

- 目标：智能体的目标是在一系列步骤中改进某个初始矩阵系统 $\mathcal{A}_s = (A_1^s, \ldots, A_n^s)$，最终得到一个矩阵系统 $\mathcal{A}_f = (A_1^f, \ldots, A_n^f)$，使其拥有的实数解远多于 $\|A_1^s x\|_2^2 = \cdots = \|A_n^s x\|_2^2 = 1$。

4.2 奖励函数实现

引理3.1可通过最小化函数 f 来证明：该函数定义如下，其中 $Q_1, \ldots, Q_n$ 是正定矩阵。

$$f(t_1, \ldots, t_n) = \ln \det(e^{t_1}Q_1 + \cdots + e^{t_n}Q_n) \text{ s.t. } \quad t_1 + \cdots + t_n = 0$$

通过此最小化，我们得到满足引理3.1的 α_i和 Z，使得 $\alpha_i = e^{t_i}$，且对于 $S = e^{t_1}Q_1 + \cdots + e^{t_n}Q_n$， $Z = S^{-1/2}$。根据引理4.5，函数 f关于 $[3,$是凸的。$7],$因此，通过设置 $t_n = -t_1 - \cdots - t_{n-1}$来消除约束，我们可以使用一阶方法。对于 $t_1, \ldots, t_{n-1}$，令 $E_n = e^{-t_1 - \cdots - t_{n-1}}Q_n$和 $S = e^{t_1}Q_1 + \cdots + e^{t_{n-1}}Q_{n-1} + E_n$。梯度为

$$\nabla f(t_1, \ldots, t_{n-1}) = \nabla \ln \det(S) = \begin{bmatrix} \text{Trace}(S^{-1}(e^{t_1}Q_1 - E_n)) \\ \text{Trace}(S^{-1}(e^{t_2}Q_2 - E_n)) \\ \vdots \\ \text{Trace}(S^{-1}(e^{t_{n-1}}Q_{n-1} - E_n)) \end{bmatrix}$$

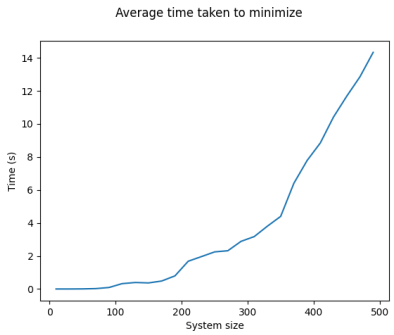
为了将其与我们的设定联系起来，令 $\mathcal{A} =$(A1, ..., An)为一个未归一化矩阵系统。我们将使用半正定矩阵 $A_1^TA_1, \ldots, A_n^TA_n$，而不是正定矩阵 $Q_1, \ldots, Q_n$。归一化过程将返回矩阵$T_i = \alpha_i Z^T A_i^T A_i Z$，其中$c_i = \frac{1}{\|A_i Z\|^2} = \frac{1}{\text{迹}(Z^T A_i^T A_i Z)} = \alpha_i$。尽管 Barvinok 的归一化过程假设矩阵为正定矩阵，而我们只能保证其为半正定矩阵，但我们的归一化计算从未遇到未定义的 f。Gurvits 和 Samorodnitsky $[17,$ 定理1.$10]$ 给出了矩阵系统必须满足的条件，以便归一化过程能够正常工作。

有了这些准备，我们通过 SciPy 使用 BFGS 方法 [34]进行无约束优化。对于每个系统规模 n，我们生成并归一化了 50 个系统，图中显示的是所有 50 个生成系统的平均结果。系统 $\mathcal{A} =$(A1、.....、 An) 的生成方式是创建 n 个高斯随机矩阵，其元素独立地取自 $\mathcal{N}$(0，1) 分布。图 1a 展示了归一化这些系统所需的时间，图 1b 和图 1c 展示了归一化后的系统与引理3.1条件的接近程度。图 1c 展示了系统求和矩阵（即 $A_1^TA_1 + \ldots + A_n^TA_n$）与 $\mathbb{I}$ 之间的欧几里得距离。图 1b 展示了由归一化矩阵的迹构成的向量（即 [Trace($A_1^TA_1$) 、.....、Trace($A_n^TA_n$)],）与全1向量之间的欧几里得距离。就准确性而言，需要特别留意迹距离，因为在图 1b 的后半部分，它呈跷跷板式增长，趋近于 1e-6 。

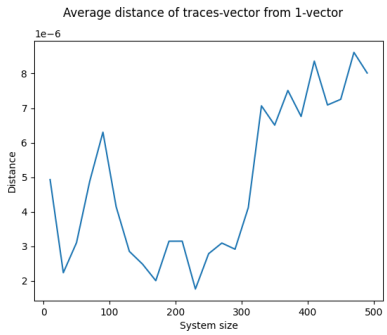
为了改善迹距离，我们在 BFGS 后使用牛顿共轭二阶优化方法 [34] 进行了五步优化，但如表1所示，计算时间的增加并不值得这点微小的精度提升。计算时间增加是因为 f 的海森矩阵计算开销很大

$$\frac{\partial^2 f}{\partial t_i \partial t_j}(x) = -\text{Trace}(S^{-1}(e^{x_i}Q_i - E_n)S^{-1}(e^{x_j}Q_j - E_n)) + \text{Trace}(S^{-1}E_n)$$

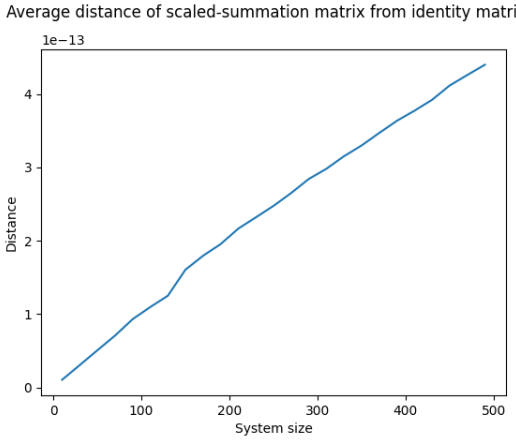
此外，正如表1最后一列所示，精度误差导致大多数系统无法得到改进。因此，我们没有采用牛顿修正。



(a) 对矩阵系统进行归一化所需的平均时间。



(b) 系统矩阵迹与全1向量之间的平均欧几里得距离。



(c) 系统矩阵求和与单位矩阵之间的平均欧几里得距离。

	耗时（秒）		迹距离		求和距离		
系统规模	**BFGS**	**Newton-CG算法**	**BFGS**	**Newton-CG算法**	**BFGS**	**Newton-CG算法**	**成功率**
50	0.0079	0.11	1.3e-5	8.9e-6	5.0e-14	5.0e-14	0.3
150	0.37	10	7.6e-7	5.8e-7	1.6e-13	1.6e-13	0.08
250	2.3	82	8.8e-6	7.5e-6	2.5e-13	2.5e-13	0.2

表1：使用牛顿-共轭校正器进行5步所实现的精度提升。此处使用了与第4.2节相同的系统。仅对成功校正的系统取平均值。

完成此归一化后，我们采用若干数学技巧来简化蒙特卡罗计算。我们在第4.3节中说明蒙特卡罗计算的细节。主要结论是，蒙特卡罗计算高度可并行，因而具有可扩展性，而计算机代数替代方案则扩展性不佳。

奖励函数本质上是一个以概率方式计算的配分函数（如统计物理学中的配分函数）。因此，我们不能指望它区分根数相近的系统。我们的目标是利用它区分根数较多的系统（目标状态）与根数较少的系统（初始状态）。为验证这一点，我们在小规模实例中使用计算代数软件提供的真实值进行了实验。在此过程中，我们还调整了蒙特卡洛实现中引入的超参数。实现细节和超参数配置将在第4.3节讨论。实验过程见附录A。

4.3 蒙特卡罗实现细节

我们使用 $\mathbb{E}N(\mathcal{B},\ \delta,\ c)$ 表示方程组的解的平均数量

$$\|A_1x\|^2 = c_1\ , \|A_2x\|^2 = c_2\ ,\ \cdots,\ \|A_nx\|^2 = c_n$$

根据第3节的讨论，该公式为

$$\int_{\mathbb{R}^n}\int_{K_c(x)} |\det Z^{'}(x)|\ d\mu(X_1, X_2, \ldots, X_n)\ dx.$$

现在回想一下，权重 $\mu(K_c(x))$ 非常小，除非 x 满足方程 (18)，也就是说，我们必须有

$$\|x\| \sim \frac{\sqrt{n}}{\sqrt{1+\delta^2 n}}.$$

我们还应注意，要使 $A_i = C_i + \delta X_i$ 成为 B_i 的良好代理，必须满足

$$\delta n = \mathbb{E}\|\delta X_i\|_F < \|C_i\| = 1 \Rightarrow \delta < \frac{1}{n}.$$

因此，我们将 $0 < \delta \leq \frac{1}{n}$ 限制在此参数范围内，并且必须满足 $\|x\| \sim \sqrt{n}$，这样 $\mu(K(x))$ 才不会可以忽略。我们将加入一个 $\varepsilon > \frac{4}{\sqrt{n}}$ 容差，并尝试从环域中均匀采样向量

$$(1-\varepsilon)\sqrt{n} < \|x\| < (1+\varepsilon)\sqrt{n}$$

这里回顾关于高斯测度的两个事实：(1) 高斯测度具有旋转不变性；(2) 对于标准高斯向量 $x \in \mathbb{R}^n$，有

$$\mathbb{P}\{(1-\varepsilon)\sqrt{n} \leq \|x\| \leq \frac{\sqrt{n}}{1-\varepsilon}\} \geq 1 - 2e^{-\frac{\varepsilon^2 n}{4}}$$

由于我们固定了 $\varepsilon > \frac{4}{\sqrt{n}}$，因此有

$$\mathbb{P}\{(1-\varepsilon)\sqrt{n} \leq \|x\| \leq \frac{\sqrt{n}}{1-\varepsilon}\} \geq 1 - 2e^{-4} \geq 0.97$$

我们可以使用缩放系数为 $1\ 0_{\circ 97}$ 的条件密度，但我们不想为这个小细节费心，因此会直接使用标准高斯分布从环域中采样

$$(1-\varepsilon)\sqrt{n} < \|x\| < (1+\varepsilon)\sqrt{n}$$

在点 x 被采样之后，我们需要处理Kac-Rice公式中的条件期望。我们将使用 [14, 定理1.1] 以及标准的重要性抽样技巧来简化条件期望表达式。更精确地说，令 $A_i = B_i + \delta X_i$，其中 X_i 是具有独立元素、均值为零且方差为1的 $n \times n$高斯矩阵。这些 A_i 可以使用标准随机矩阵生成器进行采样。然而，这些随机量并不满足该表达式中的条件假设

$$\mathbb{E}\left(|det(Z^{'}(x))| \mid Z(x) = (c_1, c_2, \ldots, c_n)\right)\ p_Z(c_1, c_2, \ldots, c_n)$$

为了解决这一问题，我们将使用 [14, 定理1.1]。首先，我们选取采样向量的一个坐标 x_i （$x_1, x_2, \ldots, x_n$）。为确保 x_i既不太大也不太小，我们可以按如下方式选取x_i：

$$S_x = \{|x_j| : x_j^2 \geq \frac{1}{2}\ ,\ 1 \leq j \leq n\}$$

然后我们选取 x_i，使其成为满足 $|x_i|$ 是 S_x的中位数的元素。此后，我们来看我们的方程:

$$\begin{aligned} \|A_1 x\|^2 = c_i &\Leftrightarrow x^T A_1^T A_1 x = c_i \\ \|A_1 x\|^2 = c_i &\Leftrightarrow x_i^{-2}\left(c_i - x^T A_1^T A_1 x + a_{i,i}^1 x_i^2\right) = a_{i,i}^1 \end{aligned} \tag{19}$$

其中wit h滥用记号，我们用$a_{i,j}^k$来表示 $A_k^T A_k$的第 i, j个元素。现在，我们se t

$$g(A_j, c_j, x) := x_i^{-2}\left(c_j - x^T A_j^T A_j x + a_{i,i}^j x_i^2\right)$$

假设坐标 i 已经确定。因此，如果我们像之前一样对 A_1 的所有条目进行抽样，但 $a_{i\,i}^{(1)}$,然后利用方程 (19) 将 $a_{i\,i}^{(1)}$,条件于其余所有条目，就可以确保 $\|A_1 x\|^2 = c_1$ 得到满足。于是，我们进行如下更新:

$$a_{i,i}^j \leftarrow g(A_j, c_j, x) \text{ for } j = 1, 2, \ldots, n.$$

在这种情况下，重要性抽样中的密度比将由从已抽样的 $a_{i,i}^1$ 到 $g(A_1, c_1, x)$ 的更新所决定，即:

$$\exp\left(\frac{-\left(g(A_1, c_1, x) - b_{i,i}^1\right)^2 + \left(a_{i,i}^1 - b_{i,i}^1\right)^2}{2}\right)$$

A总而言之，我们在 Kac-Rice公式中的整个系统的密度更新为

$$p(G(x)) := \prod_{j=1}^{n} exp\left(\frac{-\left(g(A_j, c_j, x) - b_{i,i}^j\right)^2 + (a_{i,i}^j - b_{i,i}^j)^2}{2}\right)$$

现在定义 $G = (g(A_1,\ c_1,\ x)\ g(A_2,\ c_2,\ x) \ldots, g(A_n,\ c_n,\ x))$，并令 D_xG 表示该方程组在 x 处的雅可比矩阵。 [14, 定理 1.1] 表明，我们有以下等式：

$$\mathbb{E}\left(|det(Z^{'}(x))| \mid Z(x) = (c_1, c_2, \ldots, c_n)\right) p_Z(c_1, c_2, \ldots, c_n) = \int |\det(D_xG)|\ p(G(x))\ \mu_1\mu_2 \ldots \mu_n \tag{20}$$

其中，μ_1，μ_2，……，μ_n 表示用于对A_1，A_2，……，A_n 进行初始采样的独立高斯分布。

注 4.1。严格来说，[14, 定理 1.1] 表明，方程（20）对于有界区间上的均匀分布成立，而非我们采用的高斯分布。不过，完全可以照此重复证明，得到上述等式。为避免增加更多技术细节，我们不再进行这一推导；但欢迎持怀疑态度的读者自行推导。

因此，我们的 Kac-Rice公式现在如下所示：

$$\mathbb{E}N(\mathcal{B}, c, \delta) = \int_{\mathbb{R}^n} \mathbb{E}\left(|det(Z^{'}(x))| \mid Z(x) = (c_1, c_2, \ldots, c_n)\right) p_Z(c_1, c_2, \ldots, c_n)\ dx$$

$$\mathbb{E}N(\mathcal{B}, c, \delta) = \int_{\mathbb{R}^n} \int |\det(D_xG)|\ p(G(x))\ \mu_1\mu_2 \ldots \mu_n\ dx$$

其中，μ_1，μ_2，……，μ_n 表示用于对随机矩阵 A_1，A_2，……，A_n进行采样的独立高斯分布。

因此，我们的蒙特卡洛近似 $\mathbb{E}N(\mathcal{B},\ c,\ \delta)$，其中 $0 < \delta < \frac{1}{n}$ 将如下进行：

1. 抽取 M个 n-矩阵元组（$A_1(t)\ A_2(t). \cdots,\ A_n(t))_{t=1}^{M}$。 M由用户选择，我们建议使用 $M > n^2$。

2. 从 $\mathbb{R}^n$ 上的标准高斯测度中独立抽取 N个数据点 x_1，x_2，……，$x_N \in \mathbb{R}^n$。其中 N 由用户确定，它将决定近似的质量。我们建议使用 $N > n^4$。

3. 对于 $t = 1$，……，M中的每个矩阵元组，计算以下增量和

$$\frac{1}{N}\sum_{i=1}^{N} |\det(D_{x_i}G(A_t, c_t))|\ \ p(G(A_t, c_t, x_i))$$

4. 将以下结果作为对 Kac-Rice公式的估计返回

$$\frac{1}{M}\sum_{t=1}^{M}\left(\frac{1}{N}\sum_{i=1}^{N} |\det(D_{x_i}G(A_t, c_t))|\ p(G(A_t, c_t, x_i))\right)$$

现在让我们明确写出 D_xG：

$$D_xG = \begin{bmatrix} \nabla g(A_1, c_1, x) \\ \nabla g(A_2, c_2, x) \\ \vdots \\ \nabla g(A_n, c_n, x) \end{bmatrix}$$

其中， ∇ 表示梯度。请注意，

$$\nabla g(A_j, c_j, x) = -2x_i^{-2} A_j^T A_j x + (c_j - x^T A_j^T A_j x)\nabla x_i^{-2}$$

其中 $\nabla x_i^2 =$(0，0，.....，0，$-2x_i^{-3}$，0，.....，0)。注意，$\nabla x_i^{-2} = -2x_i^{-2}\nabla \log(x_i)$。因此，我们有

$$D_x G = -2x_i^{-2}\left(\begin{bmatrix} A_1^T A_1 x \\ A_2^T A_2 x \\ \vdots \\ A_n^T A_n x \end{bmatrix} + \begin{bmatrix} (c_1 - x^T A_1^T A_1 x)\nabla \log(x_i) \\ (c_2 - x^T A_2^T A_2 x)\nabla \log(x_i) \\ \vdots \\ (c_n - x^T A_n^T A_n x)\nabla \log(x_i) \end{bmatrix}\right)$$

5 实验

为了能够使用 Julia Homotopy 验证智能体的输出，我们使用了规模较小的 $n = 10$ 矩阵。我们训练了回合长度各异的智能体： $L = 10$，15，20。所有实验的超参数 $\hat{}a$ 均为 0.01：这个较小的步长是为了确保我们确实在测试智能体的能力，看看它们即使无法远离初始起始点，也能否在景观中导航。我们将训练出的三个智能体与随机采样进行比较。每个智能体都从一个均匀生成的矩阵系统出发，然后使用其策略在 20 步内反复更新该矩阵系统；这一过程重复 20 次，使每个智能体进行 20 次、每次 20 步的测试运行。表 2 给出了通过智能体策略运行所有更新后矩阵系统所得的平均实数解数量。表 3 和表 4 分别给出了 20 次测试运行中，更新后的矩阵系统实数解数量超过潮流建模关注的不同解数量的运行次数，以及智能体达到该数量所用步数的中位数。

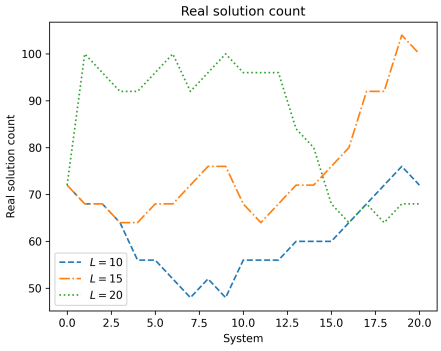
图 2 展示了训练后的智能体进行的三个测试运行示例，每个智能体都从同一个随机生成的系统开始。图 2c 显示，智能体稳步改进初始系统，尤其是 $L = 10$*20* 个智能体。不过，我们的智能体似乎并非总能实现稳步改进；例如，参见图 2a 中的 $L = 20$ 或图 2b 中的 $L = 10$，智能体在快速达到 > 80 实数解数量范围后，难以维持在该范围内。最具代表性的行为似乎是图 2b 中的 $L = 15$： 总体趋势向上，但每一步的实数解数量都有大幅增加和减少。

随机	$L = 10$	$L = 15$	$L = 20$
49.36	66.42	71.85	70.12

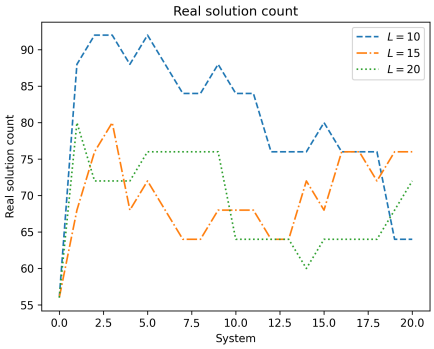
表 2：平均实数解数量

数量	随机	$L = 10$	$L = 15$	$L = 20$
80	8	9	11	11
90	4	6	7	7
100	0	2	5	2

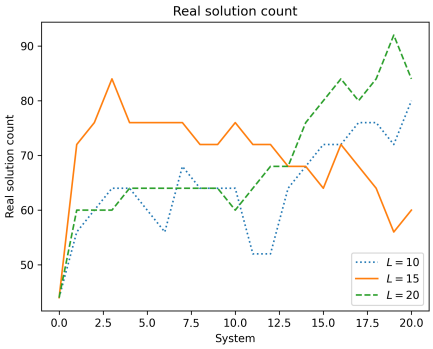
表 3：超过解数量的测试运行



(a) 测试运行 1



(b) 测试运行 2



(c) 测试运行 3

图 2： $L = 10$，15，20 次智能体测试运行

解的数量	随机	$L = 10$	$L = 15$	$L = 20$
80	12.0	7.0	6.0	1.0
90	15.5	3.5	9.0	2.0
100	N/A	14.0	11.0	3.5

表 4：智能体在测试运行中超过目标解数量所采取步数的中位数

6 讨论

我们利用一种新颖的奖励函数，将寻找具有多个解的潮流方程表述为一个强化学习问题。我们严格推导了平均情形行为，并将其用作基准。我们成功整合了多种数学洞见、凸优化算法和蒙特卡洛实现技巧，从而能够实现该奖励函数。训练后的智能体发现了比这项工作之前所能找到的解更多得多的方程；这既展现了强化学习在潮流方程建模中的潜力，也体现了它在一般情况下探索复杂非线性几何结构的潜力。我们预计，这将为潮流建模领域的进一步实验以及实代数几何中猜想的检验打开新的大门。

7 致谢

A.E. 和 V.M. 获得 NSF CCF 2414160 的资助。A.E. 感谢 Josue Tonelli Cueto 提供有益的讨论，也感谢 Cameron Khanpour 给予额外动力，促使其完成论文。A.E. 和 V.M. 还感谢 Early Birds 咖啡馆，本研究的大部分工作都在那里完成。

参考文献

[1] Taoridi Ademoye 和 Ali Feliachi。强化学习调优的电力系统分散协同控制。《电力系统研究》，86:34–40，2012。[2] Jean-Marc Aza¨ıs 和 Mario Wschebor。随机过程与场的水平集和极值。John Wiley & Sons，2009。[3] Alexander Barvinok。配分函数的组合学与复杂性，第 30 卷。Springer，2016。[4] Paul Breiding 和 Sascha Timme。Homotopycontinuation.jl——用于求解 Julia 中多项式方程组的软件包。*CoRR*，abs/1711.10911，2017。[5] T. Brysiewicz、C. Fevola 和 B. Sturmfels。实三维空间中的切二次曲面。《*Matematiche*》（卡塔尼亚），76(2):355–367，2021。[6] T. Brysiewicz、K. Kozhasov 和 M. Kummer。五次谱超曲面上的节点。《*Matematiche*》（卡塔尼亚），76(2):415–430，2021。[7] Javier Cardo-Miota、Hector Beltran、Emilio P´erez、Shafi Khadem 和 Mohamed Bahloul。基于深度强化学习的策略：在多电力市场中最大化可再生能源和储能系统的收益。应用能源，388:125561，2025。[8] Tianran Chen、Evgeniia Korchevskaia 和 Julia Lindberg。关于 Kuramoto 方程的典型解与非典型解。*arXiv*，2022。[9] H.-C. Chiang。《电力系统稳定性的直接方法、理论基础、*BCU* 方法与应用》。John Wiley and Sons，2011。[10] H.-D. Chiang、F. F. Wu 和 P. Varaiya。用于电力系统暂态稳定性直接分析的 BCU 方法。*IEEE*电力系统汇刊，9(3):1194–1208，1994。[11] Steven de Jongh、Frederik Gielnik、Felicitas Mueller、Loris Schmit、Michael Suriyah 和 Thomas Leibfried。用于电力系统推断任务的物理信息几何深度学习。《电力系统研究》，211:108362，2022。[12] P. Dietmaier。一般几何的 Stewart-Gough 平台可以具有 40 个实位姿。在《机器人运动学进展：分析与控制（萨尔茨堡，*1998*）》中，第 7–16 页。Kluwer Acad. Publ.，多德雷赫特，1998。

[13] Martin Dyer、Peter Gritzmann 和 Alexander Hufnagel。关于混合体积计算的复杂性。*SIAM*计算期刊，27(2):356–400，1998。[14] Elisenda Feliu 和 AmirHosein Sadeghimanesh。Kac-Rice公式与参数化多项式方程组的解的数量。计算数学，91(338):2739–2769，2022。[15] Scott Fujimoto、Herke Hoof 和 David Meger。解决演员-评论家方法中的函数逼近误差问题。在国际机器学习会议上，第 1587–1596 页。PMLR，2018。[16] Mevludin Glavic、Rapha¨el Fonteneau 和 Damien Ernst。电力系统决策与控制的强化学习：过去的思考与展望。*IFAC*论文在线，50(1):6918–6927，2017。第 20 届 IFAC世界大会。[17] Gurvits 和 Samorodnitsky。用于逼近混合判别式和混合体积的确定性算法及其组合学推论。离散与计算几何，27(4):531–550，2002 年 1 月。[18] Akanksha Jain 和 S.C. Gupta。使用机器学习算法进行峰值负荷预测。在*2023 IEEE*可再生能源与可持续电动交通会议（*RESEM*）上，第 1–4 页，2023。[19] Hooman Khaloie、Mih´aly Dol´anyi、Jean-Fran¸cois Toubeau 和 Fran¸cois Vall´ee。最优潮流机器学习技术综述。应用能源，388:125637，2025。[20] R. Klump 和 T. Overbye。寻找低压潮流解的新方法。在*2000*年电力工程学会夏季会议（目录号：*00CH37134*）上，第 1 卷，第 593–597 页，2000。[21] L. Lee、Y. Min、F. Xu 和 K. Wang。计算电力系统暂态稳定性分析相关不稳定平衡点的基于延拓的方法。*IEEE*电力系统汇刊，24(1):165–172，2009。[22] Bernard Lesieutre、Julia Lindberg、Alisha Zachariah 和 Nigel Boston。关于潮流方程实值解的分布。在*2019*年第*57*届*Allerton*通信、控制与计算年会（*Allerton*）上，第 165–170 页，2019。[23] Bernard Lesieutre 和 Dan Wu。定位所有潮流解的高效方法 —— 再探。在 *2015* 年第 *53* 届 *Allerton* 通信、控制与计算年会（*Allerton*）上，第 381–388 页，2015。[24] J. Lindberg、N. Boston 和 B. C. Lesieutre。利用单值性研究潮流方程中的对称性。*ACM*计算机代数通讯，54(3):100–104，2021 年 3 月。

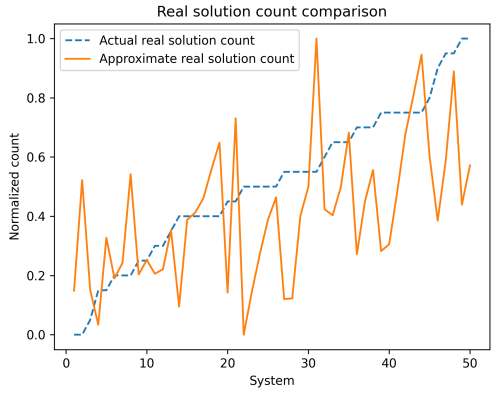
[25] Julia Lindberg、Alisha Zachariah、Nigel Boston和Bernard Lesieutre。潮流方程实数解的几何学。载于*2018*年第*56*届*Allerton*通信、控制与计算年会，第596–603页，2018。[26] Julia Lindberg、Alisha Zachariah、Nigel Boston和Bernard Lesieutre。潮流方程实数解个数的分布。*IEEE*电力系统汇刊，38(2):1058–1068，2023。[27] C. Liu、C. Chang、J. Jiang和G. Yeh。面向一种基于CPFLOW的算法，以计算电力系统中的全部1型潮流解。*IEEE*电路与系统汇刊-*I*：正规论文，52(3):625–630，2005年。[28] W. Ma和J. S. Thorp。一种定位全部潮流解的高效算法。*IEEE*电力系统汇刊，8(3):1077–1083，1993年。[29] Andrew McLennan。多齐次多项式方程组实根的期望个数。《美国数学杂志》，124(1):49–73，2002。[30] D. Mehta、D. K. Molzahn和K. Turitsyn。潮流方程计算方法的最新进展。美国控制会议（*ACC*），第1753–1765页，2016年。[31] D. Mehta、H. D. Nguyen和K. Turitsyn。用于定位全部潮流解的数值多项式同伦延拓法。*IEEE*电力系统汇刊，10(12):2972–2980，2016年。[32] Tayeb Meridji、Geza Jo´os和Jose Restrepo。一种使用机器学习的电力系统稳定性评估框架。《电力系统研究》，216:108981，2023。[33] D. K. Molzahn、B. C. Lesieutre和H. Chen。对一种用于寻找全部潮流解的基于延拓的算法的反例。*IEEE*电力系统汇刊，28(1):564–565，2013年。[34] J. Nocedal和S. Wright。《数值优化》。Springer运筹学与金融工程丛书。Springer纽约，2006年。[35] M.A. Pai。《电力系统稳定性：李雅普诺夫直接法分析》。North Holland出版公司，1981年。[36] Antonin Raffin、Ashley Hill、Adam Gleave、Anssi Kanervisto、Maximilian Ernestus和Noah Dormann。Stable-baselines3：可靠的强化学习实现。《机器学习研究杂志》，22(268):1–8，2021。[37] J. Maurice Rojas。论某些随机稀疏多项式系统实根的平均个数。载于《数值分析数学》（犹他州帕克城，*1995*年），《应用数学讲义》第32卷，第689–699页。美国数学学会，罗得岛州普罗维登斯，1996年。

[38] F. Salam、L. Ni、S. Guo和X. Sun。电力系统潮流的并行处理：方法与应用。载于第*28*届*IEEE*决策与控制会议会议录，第2173–2178页，1989年。[39] Abhishek Saxena、Ravi Shankar、Ehab F. El-Saadany、Manish Kumar、Omar Al Zaabi、Khalifa Al Hosani和Utkal Ranjan Muduli。面向增强电网可靠性的智能负荷预测与可再生能源集成。*IEEE*工业应用汇刊，60(6):8403–8417，2024年。[40] M. Shub和S. Smale。贝祖定理的复杂性。II. 体积与概率。载于计算代数几何（尼斯，*1992*年），《数学进展》第109卷，第267–285页。Birkh¨auser波士顿，波士顿，马萨诸塞州，1993年。[41] Tong Su、Tong Wu、Junbo Zhao、Anna Scaglione和Le Xie。现代电力系统安全强化学习方法综述。*IEEE*会议录，113(3):213–255，2025。[42] C. J. Tavora和O. J. M. Smith。电力系统的稳定性分析。*IEEE*电力设备与系统汇刊，3(3):1138–1144，1972年。[43] J.S. Thorp和S.A. Naqavi。潮流分形。载于第*28*届*IEEE*决策与控制年会（*CDC*），1989年12月。[44] R.T. Treinen、V. Vittal和T. Oh。一种确定电力系统中控制不稳定平衡点的改进技术。*IEEE*电路与系统汇刊*I*：基础理论，43(3)，1996年4月。[45] N. Tsolas、A. Arapostathis和P. Varaiya。一种用于电力系统暂态稳定性分析的保结构能量函数。*IEEE*电路与系统汇刊，32(10):1041–1049，1985年。[46] Pascal Van Hentenryck。面向最优潮流的机器学习。运筹学教程：新兴优化方法与建模技术及其应用，第62–82页，2021。[47] Jos´e R. V´azquez-Canteli和Zolt´an Nagy。面向需求响应的强化学习：算法与建模技术综述。应用能源，235:1072–1089，2019。[48] D. Wu。电力电网模型的代数集保持映射及其应用。威斯康星大学麦迪逊分校电气与计算机工程博士论文，2017。[49] J. Zaborszky、G. Huang、B. Zheng和T.-C. Leung。论一类大型非线性动力系统（如电力系统）的相图。*IEEE*自动控制汇刊，33(1):4–15，1988年。[50] Alisha Zachariah、Zachary Charles、Nigel Boston和Bernard Lesieutre。网络潮流方程解个数的分布。载于*2018*年*IEEE*国际电路与系统研讨会，第1–5页，2018。

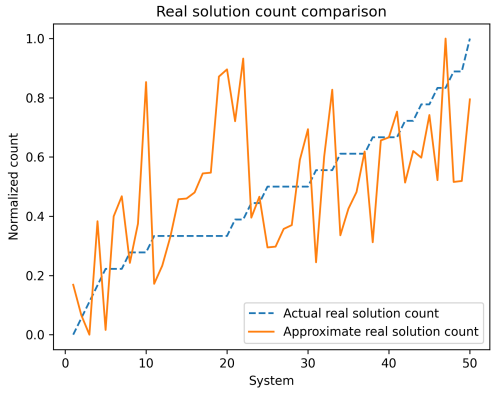
[51] Wentao Zhou、Ning Zhang、Zhenbo Cao、Yi Chen、Mingjuan Wang 和 Youbo Liu。《基于深度强化学习算法的配电网储能系统电压调节》。载于*2021*年第四届能源、电气与电力工程国际会议（*CEEPE*），第 892–896 页，2021 年。

A 蒙特卡洛实现超参数实验

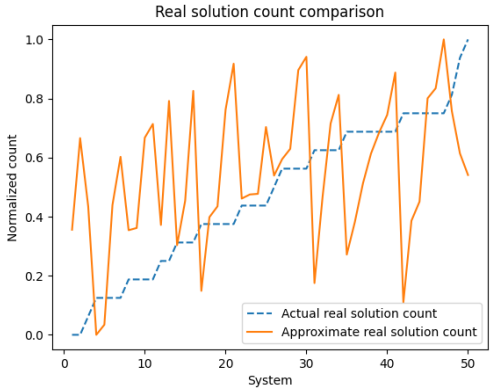
图 3 展示了蒙特卡洛近似器在 50 个系统上的超参数实验，每个系统包含 10 个独立元素以零为均值、方差为 1 的随机高斯矩阵。每次实验中，我们采样了 $N = 100000$ 个数据点和 $M = 2500$ 个矩阵元组，然后改变 δ 的取值。我们使用 Julia Homotopy [4] 计算每个系统的真实实数解数量，并用蒙特卡洛方法对其进行近似。随后，根据真实实数解数量对真实计数和近似计数进行归一化并排序。对于图 3a 中的 $\delta = 0.01$ ，前 10 个系统中有 3 个的近似值大于最后 20 个系统中的最小近似值。与之相比，在$\delta = 0.08$ 中，根据近似值无法区分实数解数量较少和较多的系统。



(a) $\delta = 0.01$



(b) $\delta = 0.05$



(c) $\delta = 0.08$

图 3： $N = 100000$, $M = 2500$ 的奖励函数对比

B 实验细节

我们的代码仓库可在此处获取。状态-行动者空间的问题实例规模经过选择，使得传递给蒙特卡洛近似器的矩阵均为 10×10,

这意味着所有智能体都可以复用蒙特卡洛超参数 $N = 100000$、$M = 2500$ 和 $\delta = 0.05$。因此，我们还引入了另外两个超参数：回合长度 L 和动作空间上限 $\hat{a}$。回合长度在 $L = 10$，15，20 之间变化；如第 5 节所述，所有智能体的 $\hat{a}$ 均为 0.01。其他 TD3 超参数均沿用 Stable Baselines3 [36] 中的 TD3 实现，没有进行调整。TD3 中使用的神经网络架构见表 5，图 4 展示了训练得到的三个智能体的训练曲线。

电网	**架构**
演员 $\pi_\phi(s)$	全连接：$D \to 500 \to 400 \to 300 \to D$，ReLU 激活函数。
评论家 $Q_\theta(s,\ a)$	全连接：$D \to 500 \to 400 \to 300 \to 1$，ReLU 激活函数。

表 5：所用的神经网络架构，其中 $D = 10 \times 10 \times 10$。

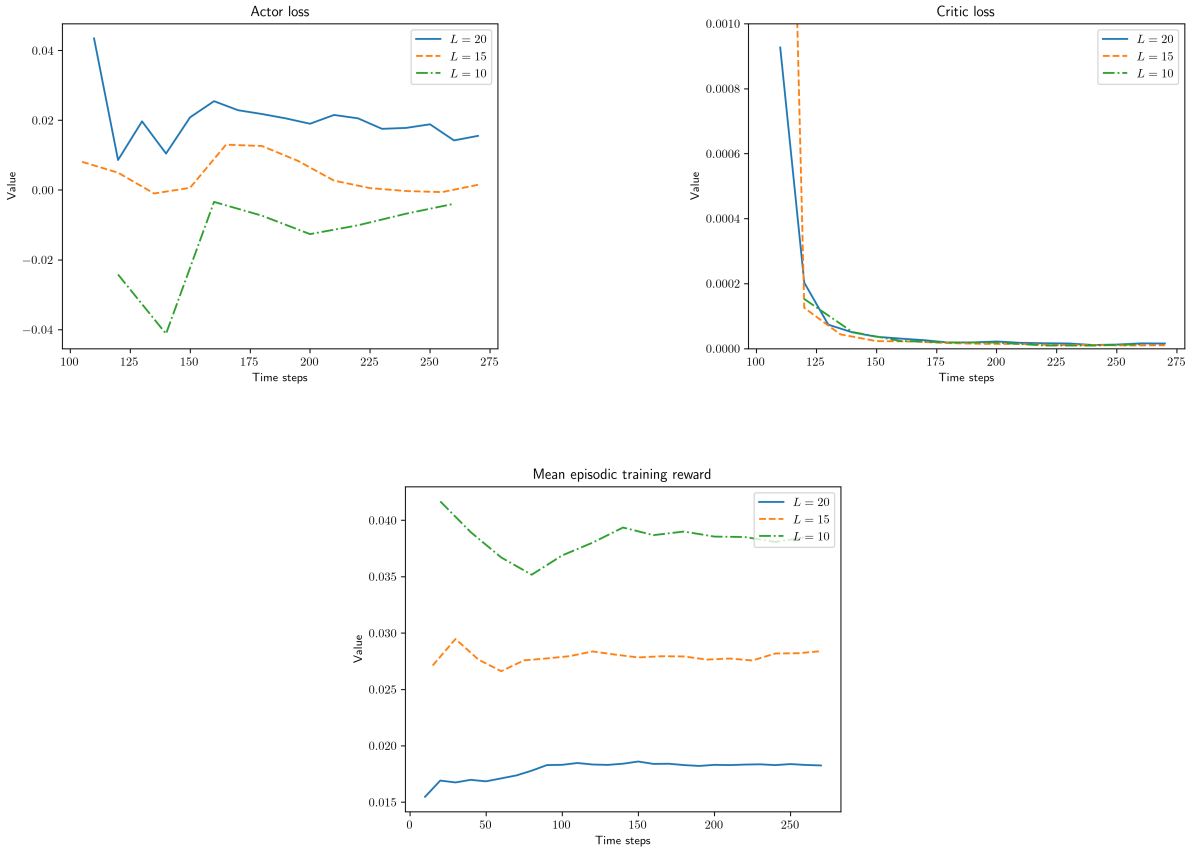


图 4：$L = 10$，15，20 个矩阵形式智能体的训练曲线图 .

C 例1.1 中潮流方程的矩阵形式

这些方程可以写成矩阵形式:

$$P_1 = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{b_{01}}{2} & 0 \\ 0 & 0 & 0 & -\frac{b_{01}}{2} & 0 & -\frac{b_{12}}{2} \\ 0 & 0 & 0 & 0 & \frac{b_{12}}{2} & 0 \\ 0 & -\frac{b_{01}}{2} & 0 & 0 & 0 & 0 \\ \frac{b_{01}}{2} & 0 & \frac{b_{12}}{2} & 0 & 0 & 0 \\ 0 & -\frac{b_{12}}{2} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{b_{12}}{2} \\ 0 & 0 & 0 & 0 & -\frac{b_{12}}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{b_{12}}{2} & 0 & 0 & 0 \\ 0 & \frac{b_{12}}{2} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

$$1 = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

$$1 = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}^T \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

$$1 = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

$$1 = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ y_0 \\ y_1 \\ y_2 \end{bmatrix}$$