arXiv:2302.05383v1 [hep-ph] 10 Feb 2023

利用机器学习发现李群的稀疏表示

Roy T. Forestano, [1], [*] Konstantin T. Matchev, [1], [†] Katia Matcheva, [1], [‡] Alexander Roman, [1], [§] Eyup B. Unlu, [1], [¶] 以及 Sarunas Verner[1], [**]

[1]基础理论研究所，物理系，佛罗里达大学，美国佛罗里达州盖恩斯维尔，邮编 *32611*

（日期：2023年2月10日）

摘要

近期研究利用深度学习推导出保持守恒量不变的对称变换，并获得相应的生成元代数。在这封信中，我们将这一技术拓展到推导任意李代数的稀疏表示。我们展示了该方法能够重现洛伦兹群生成元的标准（稀疏）表示，以及 $U(n)$ 和 $SU(n)$ 李群族的标准表示。这种方法具有完全的普适性，可用于求出任意李群的无穷小生成元。

[*] roy.forestano@ufl.edu

[†] matchev@ufl.edu

[‡] matcheva@ufl.edu[§] alexroman@ufl.edu

[¶] eyup.unlu@ufl.edu

[**] verner.s@ufl.edu

I. 引言

理论物理在 20th 世纪之交取得了变革性进展，人们认识到，对称性是自然的基本特征，它约束着可能的动力学定律 [1]。根据诺特定理 [2], ，问题中存在连续对称性，就意味着存在一条普遍适用、对于理解系统的行为和演化不可或缺的守恒定律。对称性存在于各个尺度——从标准模型中亚原子粒子的微观描述，到宇宙的大尺度结构。在粒子物理学中，对称性为观测到的粒子谱系及其相互作用提供了组织原则，并指导模型构建者寻找标准模型的可行扩展 [3]。

历史上，对称性的形式化研究一直是在群论 [4] 的框架内进行的。几乎所有类型的经典李群都在物理学的各个分支领域中有所应用，而粒子物理学中最常见的李群是特殊正交群 $SO(n)$ 和特殊酉群 $SU(n)$。具体而言，旋转群 $SO(3)$体现了空间各向同性以及所有基本运动定律；洛伦兹群 $SO(1,3)$ 是狭义相对论中时空的对称群；而$SU(2)$和 $SU(3)$分别是标准模型中负责弱相互作用和强相互作用的规范群。对于有志成为物理学家的研究生而言，高阶群论课程是必不可少的知识储备。

近年来，人们对应用机器学习（ML）发现和识别数据中的对称性产生了日益浓厚的兴趣 [5–16]。机器学习在群论中的相关应用包括计算李群不可约表示的张量积和分支规则 [17], ，以及检验和/或推导数据中存在的对称性的李群生成元 [11, 14–16]。在本文中，我们扩展 [15, 16] 的方法，以推导任意李代数的稀疏表示。我们将证明，通过对第II节中讨论的损失函数进行适当修改，可以利用机器学习从第一性原理推导出洛伦兹群生成元的规范（稀疏）形式（第III节），以及 $U(n)$ 和 $SU(n)$ 族（第IV节）的规范形式。如总结部分第V节所强调的，所述方法具有完全的普适性，可应用于任何其他经典李群。

II. 方法

我们首先简要总结参考文献 [15]中概述的对称性发现过程。经典群是在实数 $\mathbb{R}$、复数 $\mathbb{C}$和四元数 $\mathbb{H}$上定义的线性变换群。因此，对称变换作用于特征向量$\mathbf{x} \equiv \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$，其中，在我们的 $SO(n)$和 $SU(n)$示例中，$\mathbf{x} \in \mathbb{R}^n$ 和$\mathbf{x} \in \mathbb{C}^n$ 分别为。为了捕捉群变换对 $\mathbb{R}^n$ 或$\mathbb{C}^n$的影响，我们考虑从某个有限域中采样得到的 m 个点 $\{\mathbf{x}\} \equiv \{\mathbf{x}_1, \mathbf{x}_2, \ldots\ldots, \mathbf{x}_m\}$ 组成的代表性集合。采样分布的选择以及域的大小和位置并不重要。为明确起见，我们使用标准正态分布，并令$m = 100$。

经典群是根据其各自域上的多项式不变量定义的。例如， $O(n)$保持多项式预言机的值

$$\varphi_O(\mathbf{x}) \equiv |\mathbf{x}|^2 = \sum_{j=1}^{n} [x^{(j)}]^2, \quad x^{(j)} \in \mathbb{R}, \tag{1}$$

$n = 4$ 维空间中的洛伦兹群保持

$$\varphi_L(\mathbf{x}) \equiv \left(x^{(1)}\right)^2 - \left(x^{(2)}\right)^2 - \left(x^{(3)}\right)^2 - \left(x^{(4)}\right)^2, \quad x^{(j)} \in \mathbb{R}, \tag{2}$$

而 $U(n)$保持

$$\varphi_U(\mathbf{x}) \equiv \sum_{j=1}^{n} \left(x^{(j)}\right)^* x^{(j)}, \quad x^{(j)} \in \mathbb{C}. \tag{3}$$

一个对称变换 $\mathbf{f}$ 是一个映射 $\mathbf{x}' = \mathbf{f}(\mathbf{x})$，它在所有位置都保持相应的预言机（1-3），在我们的情形中，则是在每个采样的 m 点上：

$$\varphi(\mathbf{x}'_i) \equiv \varphi(\mathbf{f}(\mathbf{x}_i)) = \varphi(\mathbf{x}_i), \quad \forall i = 1, 2, \ldots, m\,. \tag{4}$$

为了聚焦于对称变换群的生成元，通过考虑恒等变换$\mathbb{I}$附近的无穷小变换 δ，将$\mathbf{f}$ 线性化：

$$\delta\mathbf{f} \equiv \mathbb{I} + \varepsilon\,\mathbb{G}\,, \tag{5}$$

其中， ε 是一个无穷小参数， $\mathbb{G}$ 是一个 $n \times n$ 矩阵。通过使用合适的损失函数 $L(\mathbb{G}, \{\mathbf{x}\})$进行训练， $\mathbb{G}$ 的各个分量会逐渐达到其训练后的值，

从而得到一个对称生成元 [15]

$$\mathbb{J} \equiv \arg\min_{\mathbb{G}} \Big(L(\mathbb{G}, \{\mathbf{x}\}) \Big). \tag{6}$$

针对候选生成元分量，在不同初始条件下重复上述过程 N_g 次，或使用不同的随机种子和超参数，可得到一整组 N_g 生成元 $\mathbb{J}_\alpha$, $\alpha = 1, 2, \ldots, N_g$。损失函数的设计旨在确保得到的生成元具有以下性质：

1. 不变性，即对于所有采样的数据点 $\{\mathbf{x}\}$，在所有候选变换 $\mathbb{G}_\alpha$, $\alpha = 1, 2, \ldots, N_g$构成的集合 $\{\mathbb{G}\}$ 下，保持预言机值不变：

$$L_{\text{inv}}\big(\{\mathbb{G}\}, \{\mathbf{x}\}\big) = \frac{1}{m\varepsilon^2} \sum_{\alpha=1}^{N_g} \sum_{i=1}^{m} \Big[\varphi\big(\mathbf{x}_i + \varepsilon \mathbb{G}_\alpha \cdot \mathbf{x}_i\big) - \varphi(\mathbf{x}_i) \Big]^2, \tag{7}$$

其中，“$\cdot$”表示普通的张量乘法。

2. 正交性，确保得到的生成元彼此不同：

$$L_{\text{ortho}}\big(\{\mathbb{G}\}\big) = \Big[\sum_{\alpha=1}^{N_g-1} \sum_{\beta=\alpha+1}^{N_g} \text{Tr}\big(\mathbb{G}_\alpha \cdot \mathbb{G}_\beta^T\big) \Big]^2. \tag{8}$$

3. 归一化，确保变换不是平凡的：

$$L_{\text{norm}}\big(\{\mathbb{G}\}\big) = \sum_{\alpha=1}^{N_g} \Big[\text{Tr}\Big(\mathbb{G}_\alpha \cdot \mathbb{G}_\alpha^T\Big) - 2 \Big]^2. \tag{9}$$

4. 代数闭包。这用于检验候选生成元集合 $\{\mathbb{G}\}$ 是否构成具有某些结构常数 $a_{\alpha\beta}^\gamma$ 的闭代数 $\big[\mathbb{G}_\alpha, \mathbb{G}_\beta\big] = \sum_{\gamma=1}^{N_g} a_{\alpha\beta}^\gamma \mathbb{G}_\gamma$：

$$L_{\text{closure}}\big(\{\mathbb{G}\}, a_{\alpha\beta}^\gamma\big) = \sum_{\alpha=1}^{N_g-1} \sum_{\beta=\alpha+1}^{N_g} \text{Tr}\left(\mathbb{C}_{\alpha\beta} \cdot \mathbb{C}_{\alpha\beta}^T\right), \tag{10}$$

其中，闭包失配定义为

$$\mathbb{C}_{\alpha\beta}\big(\{\mathbb{G}\}, a_{\alpha\beta}^\gamma\big) \equiv \big[\mathbb{G}_\alpha, \mathbb{G}_\beta\big] - \sum_{\gamma=1}^{N_g} a_{\alpha\beta}^\gamma \mathbb{G}_\gamma. \tag{11}$$

[15] 中的损失函数是这四项之和，并采用了适当的权重因子（超参数） h_{ortho}、 h_{norm} 和 h_{closure}：

$$L_4 = L_{\text{inv}} + h_{\text{ortho}}\, L_{\text{ortho}} + h_{\text{norm}}\, L_{\text{norm}} + h_{\text{closure}}\, L_{\text{closure}}. \tag{12}$$

[15] 证明，该方法无需任何先验假设，即可完全以无偏的方式学习由预言机（1）和（2）生成的对称群（及其子群）。然而，由于该方法具有完全的通用性，[15] 中学到的表示并未与特征空间的坐标轴恰当地对齐，因此不同于教科书中漂亮而传统的结果。为了解决这一问题，我们在此引入一个旨在促进稀疏性的额外损失项：

$$L_{\text{sparsity}}\big(\{\mathbb{G}\}\big) = \sum_{\alpha=1}^{N_g}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{j'=1}^{n}\sum_{k'=1}^{n}\Big|\mathbb{G}_\alpha^{(jk)}\mathbb{G}_\alpha^{(j'k')}\Big|\Big(1-\delta_{jj'}\delta_{kk'}\Big), \tag{13}$$

其中 $\mathbb{G}_\alpha^{(jk)}$表示 $\mathbb{G}_\alpha$的 jk分量。因此，完整的损失函数为

$$L\big(\{\mathbb{G}\}, a_{\alpha\beta}^{\gamma}\big) = L_4\big(\{\mathbb{G}\}, a_{\alpha\beta}^{\gamma}\big) + h_{\text{sparsity}}\, L_{\text{sparsity}}\big(\{\mathbb{G}\}\big) \tag{14}$$

并使用学习率介于 0.0001 和 0.01 之间的 Adam 优化器，对所有生成元分量 $\mathbb{G}_\alpha^{(jk)}$和结构常数 $a_{\alpha\beta}^{\gamma}$进行最小化。除非另有说明，超参数 h 均固定为 1。

III. 洛伦兹群 $SO(1,3)$

作为参考基准，图1 左栏展示了 [15] 针对洛伦兹群的$N_g = 6$ 生成元所得的结果，使用损失函数（12）和预言机（2）。在此图及后续所有同类图中，每个 $n \times n$ 面板表示一个学习得到的生成元 $\mathbb{J}_\alpha$ 的矩阵形式。各矩阵元的数值由色条标示。图1底部面板中的每一行（标记为 $\alpha\beta = 12$、13、.....、56）代表 15 个可能的唯一对易子之一 $[\mathbb{J}_\alpha, \mathbb{J}_\beta]$，而各列（标记为 $\gamma = 1$、2、.....、6）代表找到的生成元 $\mathbb{J}_\gamma$。每个单元格代表一个结构常数 $a_{\alpha\beta}^{\gamma}$，其数值由色条标示。

尽管在 [15]中找到的六个生成元确实满足上一节讨论的四项标准 1-4，但它们并不容易解释，因为这些生成元最终是熟悉的 boost 生成元 $\mathbb{K}_k$ 和旋转生成元 $\mathbb{L}_k$的通用线性组合，$k = 1, 2, 3$。换言之，图1 左上方各面板中的表示并不具有稀疏性。在将稀疏性项（13）纳入损失函数（14）后，我们得到图1右侧各面板所示的相应结果。我们很容易识别出 $\mathbb{J}_1 = -\mathbb{K}_2$、$\mathbb{J}_3 = -\mathbb{K}_3$ 和 $\mathbb{J}_4 = \mathbb{K}_1$

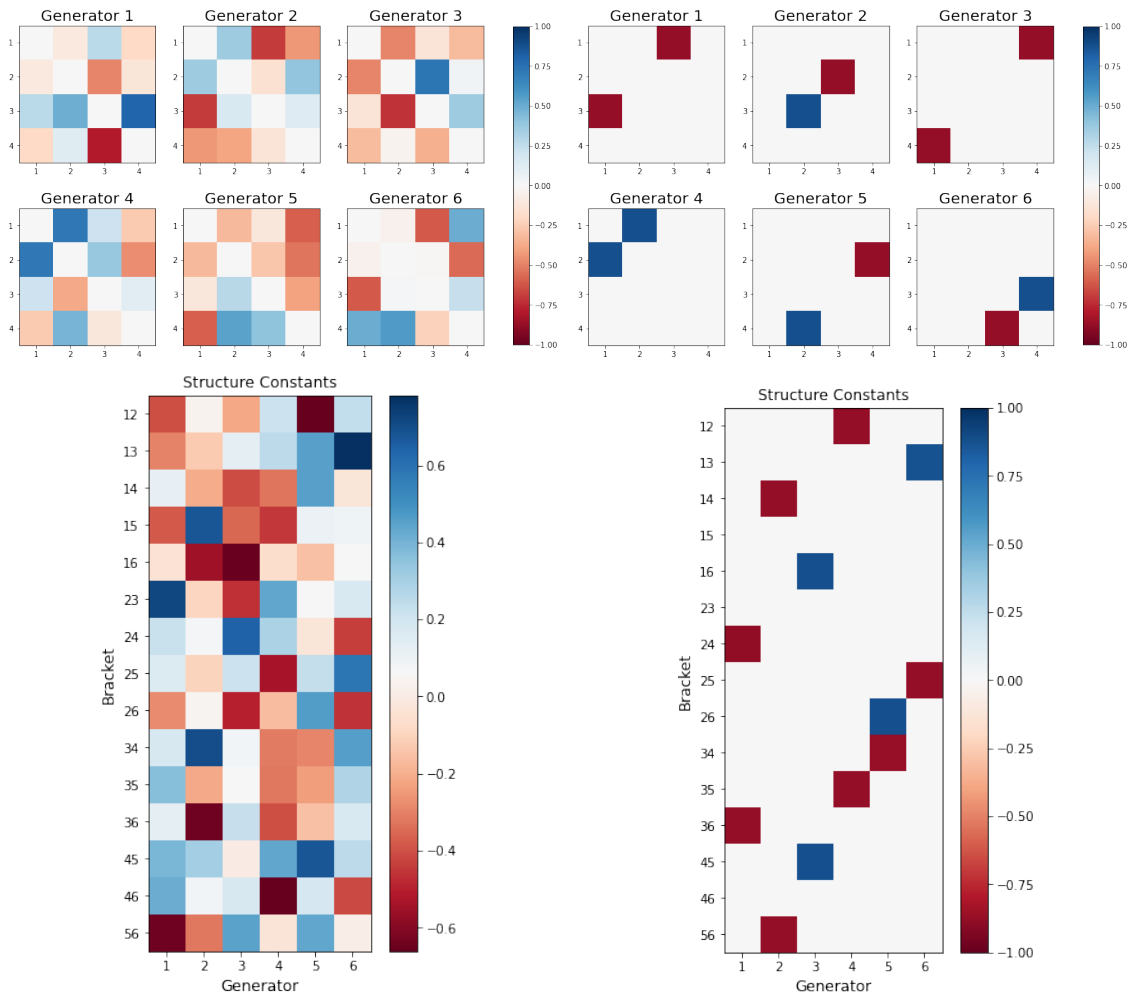

图 1. 左：参考文献 [15] 中通过使用损失函数（12）和预言机（2）（上方图）得到的洛伦兹群生成元，以及相应的结构常数（下方图）。右：加入稀疏性项（13）后得到的相应结果，其中 $h_{\text{sparsity}} = 1$。

分别是标准 boost 生成元，而 $\mathbb{J}_2 = \mathbb{L}_3$、$\mathbb{J}_5 = -\mathbb{L}_2$ 和 $\mathbb{J}_6 = -\mathbb{L}_1$ 是标准旋转生成元。

与任何机器学习方法一样，我们的方法依赖于损失函数的最小化，这涉及一定的超参数调优。为了获得最佳结果，我们确实需要成功平衡损失函数中不同项的相对权重（14）。图 2 展示了 $N_g = 4$洛伦兹群的子代数对应的学习得到的生成元随稀疏性超参数 h_{sparsity} 的变化关系，该参数的扫描范围为 0 到 10。随着稀疏性项逐渐启用，学习得到的表示变得更加稀疏；在 h_{sparsity}取某一范围的值时，我们恢复出一组规范集合 $\mathbb{J}_1 = \mathbb{L}_3$、$\mathbb{J}_2 = \mathbb{K}_2 + \mathbb{L}_1$、$\mathbb{J}_3 = \mathbb{K}_1 - \mathbb{L}_2$以及

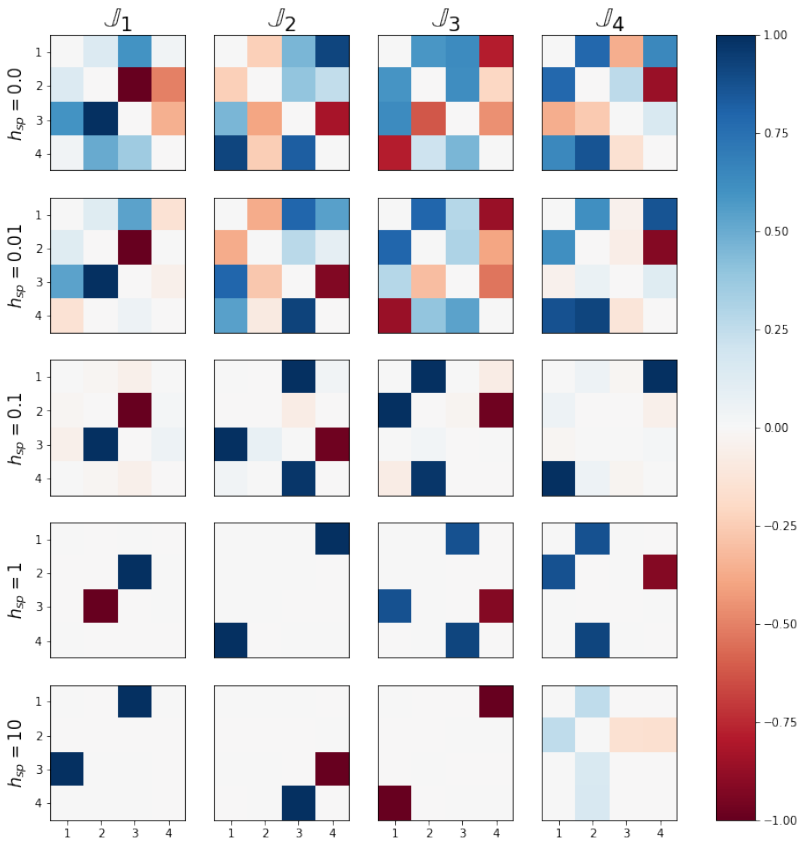

图 2. 不同超参数 h_{sparsity} 取值下，洛伦兹群的 $N_g = 4$ 子代数学习得到的生成元： $h_{\text{sparsity}} = 0$、0.01、0.1、1、10（从上到下）。每次训练过程均进行了 3,000 个训练轮次。

$\mathbb{J}_4 = \mathbb{K}_3$。然而，进一步增大 h_{sparsity} 会导致解变得非常稀疏，但代价是违反其他条件之一；在本例中是归一化条件——注意，图 2 最后一行中的 $\mathbb{J}_4$ 本质上是平凡的。

IV. $U(n)$与 $SU(n)$族

现在，我们将注意力转向 $U(n)$族和 $SU(n)$族，这些族在[15]或当前语境的其他地方都未曾考虑过。这两个族的关系为 $U(n) = SU(n) \times U(1)$，其中 $U(1)$因子用于满足 $\det U = 1$ 条件。由于不变多项式(3)定义在复数域上，我们需要对第II节中的过程做一些调整：i) 创建数据集 $\{\mathbf{x}\}$，其中现在包含 m n维复向量；ii) 在损失函数中定义的函数里使用复数形式，方法是

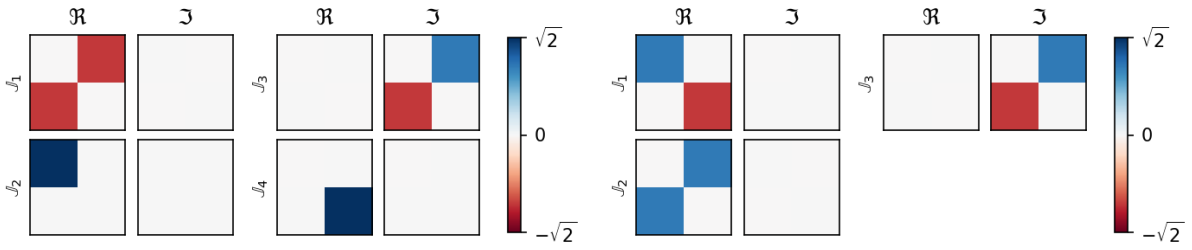

FIG. 3. $U(2)$（左）和 $SU(2)$（右）情况下学得的稀疏生成元 .

将 $\mathbb{G}\mathbb{G}$ 的任意乘积替换为 $\mathbb{G}\mathbb{G}^*$；以及 iii) 在(5)中的 ε 前插入惯用的虚数单位因子 $\delta\mathbf{f} = \mathbb{I} + i\varepsilon\,\mathbb{G}$。使用 n^2 生成元进行训练后，我们将得到 $U(n)$的代数。为了找到 $SU(n)$的代数，我们利用该方法能够找到构成闭代数的生成元集合这一优势。因此，我们只需将生成元的数量减少到 n^2-1，然后像之前一样继续即可。

A. $U(2)$和 $SU(2)$的情况

我们关于 $n = 2$ 情况的结果如图3所示。由于生成元矩阵的分量现在是复数，我们分别展示其实部（$\mathfrak{R}$）和虚部（$\mathfrak{I}$）面板。 $U(2)$ 的生成元由 $SU(2)$ 的生成元构成，通常取为泡利矩阵

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{15}$$

再加上 2×2 单位矩阵 $\sigma_0 = \mathbb{I}_2$。我们看到，对于图 3 左侧所示的 $N_g = 4$ $U(2)$情形，该方法正确地找到了 $\mathbb{J}_1 = -\sigma_1$、 $\mathbb{J}_3 = -\sigma_2$、$\mathbb{J}_2 = (\sigma_0+\sigma_3)/\sqrt{2}$、 $\mathbb{J}_4 = (\sigma_0-\sigma_3)/\sqrt{2}$。算法选择最后两个线性组合，是因为它们比规范集合 σ_0 和 σ_3更稀疏。请注意，当一个生成元矩阵只有一个非零元素时，根据归一化条件（9），其值被固定为 $\pm\sqrt{2}$ 。当使用 $N_g = 3$ 重复这一操作时，即$SU(2)$的情形，我们得到图 3 右侧所示的结果。我们识别出 $\mathbb{J}_1 = \sigma_3$、$\mathbb{J}_2 = \sigma_1$、 $\mathbb{J}_3 = -\sigma_2$，这正是除符号和重新标记外的规范 $SU(2)$代数。

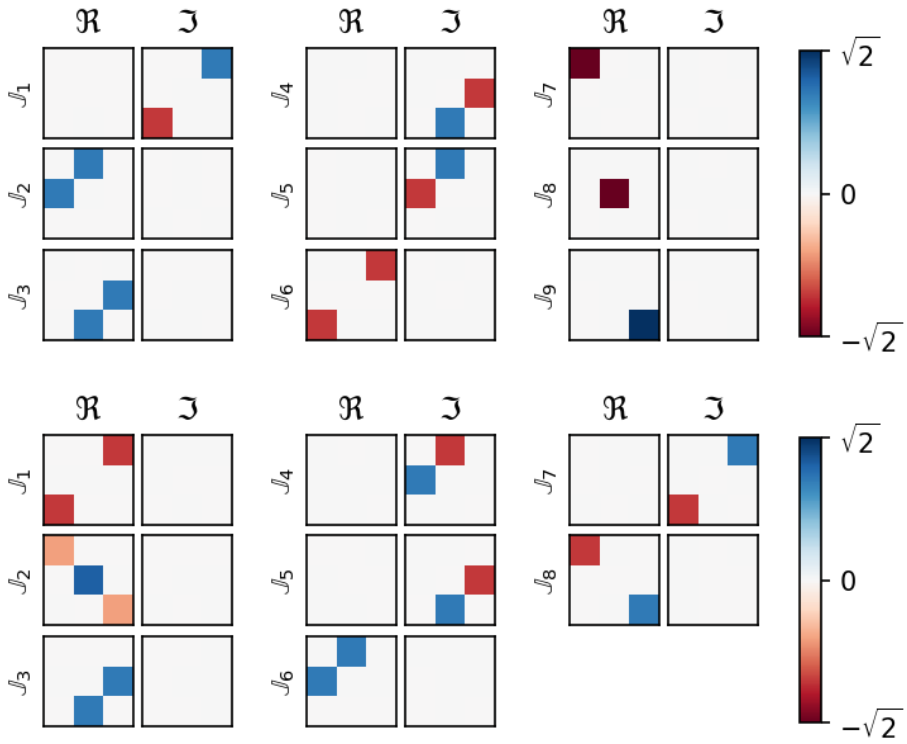

图 4. 与图 3相同，但对应于 $U(3)$（上方各图）和 $SU(3)$（下方各图）。

B. $U(3)$和 $SU(3)$的情形

现在我们继续讨论 $n = 3$的情形。 $SU(3)$ 的生成元是盖尔曼矩阵，它们是泡利矩阵在 $SU(3)$ 中的对应物（15）

$$\lambda_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \lambda_2 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \lambda_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$
$$\lambda_4 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \lambda_5 = \begin{pmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{pmatrix}, \tag{16}$$
$$\lambda_6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \lambda_7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix}, \quad \lambda_8 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}.$$

我们的结果如图 4所示。检查结果可见，生成元已成功复现。特别是，对于 $SU(3)$，算法找到了 $\mathbb{J}_1 = -\lambda_4$， $\mathbb{J}_2 =$

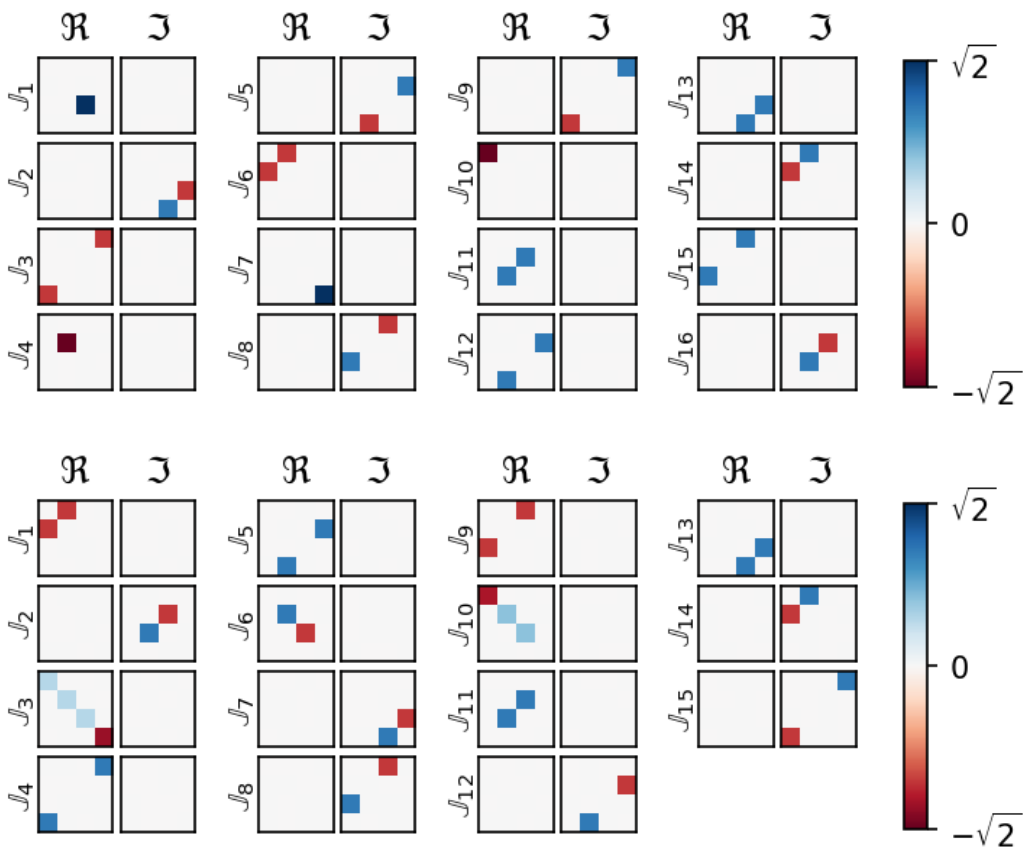

图 5. 与图 4相同，但针对 $U(4)$（上排图）和 $SU(4)$（下排图）。

$(\lambda_8 - \sqrt{3}\lambda_3)/2$, $\mathbb{J}_3 = \lambda_6$, $\mathbb{J}_4 = \lambda_2$, $\mathbb{J}_5 = \lambda_7$, $\mathbb{J}_6 = \lambda_1$, $\mathbb{J}_7 = -\lambda_5$, $\mathbb{J}_8 = -(\lambda_3 + \sqrt{3}\lambda_8)/2$.

C. $U(4)$ 和 $SU(4)$ 的情形

图 5 展示了下一个最复杂情形 $n = 4$的结果。 $SU(4)$ 的生成元是盖尔曼矩阵（16）的相应推广，由 6 个具有实数非对角分量的对称矩阵、6 个具有纯虚数非对角分量的反对称矩阵，以及 3 个具有实数对角分量的矩阵组成。这正是图 5下方各面板所显示的结果。 $U(4)$ 代数还多了一个对角生成元，可取为单位矩阵 $\mathbb{I}_4$。再次地，损失函数中的稀疏性项会迫使 $U(4)$ 中具有实数对角分量的生成元仅有一个非零元素，见 $\mathbb{J}_1$、 $\mathbb{J}_4$、 $\mathbb{J}_7$和 $\mathbb{J}_{10}$。

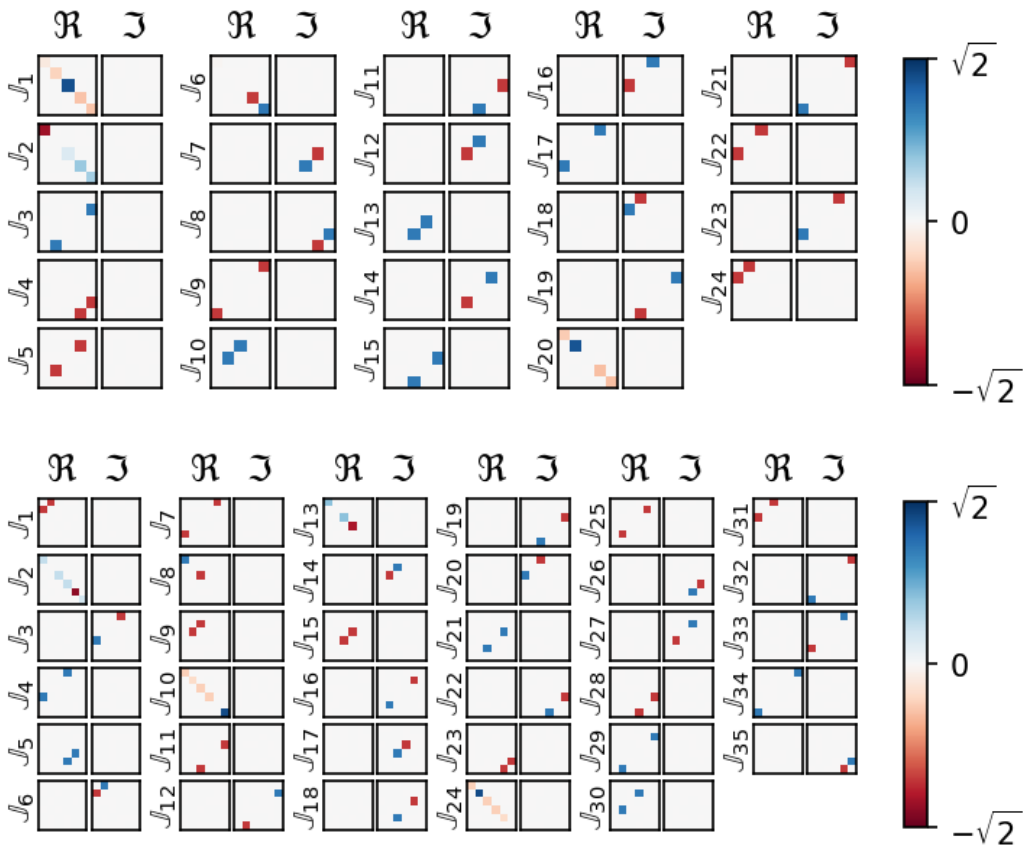

图 6. $SU(5)$（上方图）和 $SU(6)$（下方图）的学习所得稀疏生成元。

D. $SU(5)$ 及更高阶情形

现在我们转而讨论 $n = 5$ 和 $n = 6$。$SU(5)$ 和$SU(6)$ 的学习得到的生成元如图 6 所示。对于 $SU(5)$，我们有 4 个具有实对角元素的生成元（$\mathbb{J}_1$、$\mathbb{J}_2$、$\mathbb{J}_6$ 和 $\mathbb{J}_{20}$），10 个具有实数非对角分量的对称矩阵（$\mathbb{J}_\alpha$ 及 $\alpha \in \{3, 4, 5, 9, 10, 13, 15, 17, 22, 24\}$），以及 10 个具有虚数非对角分量的反对称矩阵（$\mathbb{J}_\alpha$ 及 $\alpha \in \{7, 8, 11, 12, 14, 16, 18, 19, 21, 23\}$）。对于$SU(6)$，在总计 35 个生成元中，我们有 5 个具有实对角元素的生成元（$\mathbb{J}_\alpha$ 至 $\alpha \in \{2, 8, 10, 13, 24\}$），15 个具有实数非对角分量的对称矩阵（$\mathbb{J}_\alpha$ 至 $\alpha \in \{1, 4, 5, 7, 9, 11, 15, 21, 23, 25, 28, 29, 30, 31, 34\}$），以及 15 个具有虚数非对角分量的反对称矩阵（$\mathbb{J}_\alpha$ 至 $\alpha \in \{3, 6, 12, 14, 16, 17, 18, 19, 20, 22, 26, 27, 32, 33, 35\}$）。在图中成功发现 $SU(5)$ 和 $SU(6)$ 的生成元后——

图 6，我们相信，我们的方法已证明完全适用于任意更高维群 $SU(n)$ 且具有 $n > 6$。

V. 总结与展望

在这封信中，我们展示了如何利用机器学习推导任意李代数生成元的稀疏表示。关键的观察是，李群在向量空间上的群作用会保持某个不变多项式（对于本文考察的例子，该多项式为预言机函数（1-3））。如 [15], 所示，可以为生成元分量设计合适的损失函数，使损失函数的最小化得到一组有效生成元。在这封信中，我们扩展了[15] 的方法，通过在损失函数中添加鼓励寻找稀疏表示的项。我们在洛伦兹群（第III节）以及 $U(n)$ 和 $SU(n)$ 李群族（第 IV 节）上验证了该方法。虽然本文未展示相关结果，我们还在 $SO(n)$ 中考察的所有 [15]例子上验证了这一流程。

需要指出的是，我们所有的数值计算都是在个人笔记本电脑上完成的，并未使用任何高性能计算设施。所有模型均训练了 5,000 个训练轮次；对于 $SU(6)$，典型训练时间从几分钟到几小时不等。尽管该方法适用于任意大的群，但在实践中，此类情况下的训练可能会相对缓慢，并可通过并行化、进一步的代码优化、超参数调优以及更深入地理解损失景观来改善。像例外李群 [18], 这样规模大得多的群，将在未来的工作中进行探索。

致谢

感谢 A. Davis、S. Gleyzer、R. Houtz、K. Kong、S. Mrenna、H. Prosper 和 P. Shyam- sundar 提供有益的讨论。感谢 P. Ramond 分享群论见解并给予启发。本研究部分由美国能源部资助，资助编号为

DE-SC0022148.

[1] David J. Gross, 《对称性在基础物理学中的作用》, 《美国国家科学院院刊》93, 14256–14259（1996）。[2]E. Noether, 《不变变分问题》, Nachrichten von der Gesellschaft der Wis- senschaften zu G¨ottingen, 数学物理学部 1918, 235–257（1918）。[3] Csaba Cs´aki、Salvator Lombardo 和 Ofri Telem, 《非超对称标准模型扩展模型 TASI讲座》, 载于《理论高级研究院基本粒子物理学研究所论文集：展望粒子物理学的下一步发现（TASI 2016）》, 美国科罗拉多州博尔德, 2016年6月6日至7月1日, 由 Rouven Essig 和 Ian Low 编（WSP, 2018）, 第501–570页, arXiv:1811.04279 [高能物理学-现象学]。[4] E.P. Wigner、J.J. Griffin 和 J.J. Griffin, 《群论及其在原子光谱量子力学中的应用》, 《纯粹与应用物理学：专著与教科书系列》（学术出版社, 1959）。[5]Raban Iten、Tony Metger、Henrik Wilming、L´ıdia del Rio 和 Renato Renner, 《利用神经网络发现物理概念》, 《物理评论快报》124, 010508（2020）, arXiv:1807.10300 [量子物理学]。[6] Sebastian J. Wetzel、Roger G. Melko、Joseph Scott、Maysum Panju 和 Vijay Ganesh, 《通过解读孪生神经网络发现对称不变量和守恒量》, 《物理评论研究》2, 033499（2020）, arXiv:2003.04299 [计算物理学]。[7] Sven Krippendorf 和 Marc Syvaeri, 《利用神经网络检测对称性》（2020）, arXiv:2003.13679 [计算物理学]。[8] Ziming Liu 和 Max Tegmark, 《从轨迹中通过机器学习发现守恒定律》, 《物理评论快报》126, 180604（2021）, arXiv:2011.04698 [计算机科学-机器学习]。[9] Gabriela Barenboim、Johannes Hirn 和 Veronica Sanz, 《对称性遇上人工智能》, 《SciPost物理学》11, 014（2021）, arXiv:2103.06115 [计算机科学-机器学习]。[10] Barry M. Dillon、Gregor Kasieczka、Hans Olischlager、Tilman Plehn、Peter Sorrenson 和 Lorenz Vogel, 《对称性、安全性与自监督》, 《SciPost物理学》12, 188（2022）, arXiv:2108.04253

[hep-ph].

[11] Ziming Liu 和 Max Tegmark，《机器学习隐藏对称性》，《物理评论快报》128，180201（2022），arXiv:2109.09721 [计算机科学-机器学习].[12] Krish Desai、Benjamin Nachman 和 Jesse Thaler，《利用深度学习发现对称性》，《物理评论D》105，096031（2022），arXiv:2112.05722 [高能物理学-现象学].[13] Sean Craven、Djuna Croon、Daniel Cutting 和 Rachel Houtz，《机器学习流形》，《物理评论D》105，096030（2022），arXiv:2112.07673 [高能物理学-现象学].[14] Artem Moskalev、Anna Sepliarskaia、Ivan Sosnovik 和 Arnold Smeulders，《Liegg：研究学习得到的李群生成元》，（2022），arXiv:2210.04345 [计算机科学-机器学习].[15] Roy T. Forestano、Konstantin T. Matchev、Katia Matcheva、Alexander Roman、Eyup Unlu 和 Sarunas Verner，《深度学习对称性及其李群、李代数和子代数：从第一性原理出发》，（2023），arXiv:2301.05638 [高能物理学-现象学].[16] Alexander Roman、Roy T. Forestano、Konstantin T. Matchev、Katia Matcheva 和 Eyup B. Unlu，《保持预言机的潜流》，（2023），arXiv:2302.00806 [计算机科学-机器学习].[17] Heng-Yu Chen、Yang-Hui He、Shailesh Lal 和 Suvajit Majumder，《机器学习李结构及其在物理学中的应用》，《物理快报B》817，136297（2021），arXiv:2011.00871[高能物理学-理论].[18] Pierre Ramond，《例外李群与李代数导论》，技术报告 CALT-68-577（加州理工学院，帕萨迪纳，加利福尼亚州 91125，1976年）。