Phys. Lett. B 847 (2023) 138266

Contents lists available at ScienceDirect

Physics Letters B

journal homepage: www.elsevier.com/locate/physletb

Accelerated discovery of machine-learned symmetries: Deriving the exceptional Lie groups G_2, F_4 and E_6

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

Institute for Fundamental Theory, Physics Department, University of Florida, Gainesville, FL, 32611, USA

ARTICLE INFO

Editor: J. Hisano

ABSTRACT

Recent work has applied supervised deep learning to derive continuous symmetry transformations that preserve the data labels and to obtain the corresponding algebras of symmetry generators. This letter introduces two improved algorithms that significantly speed up the discovery of these symmetry transformations. The new methods are demonstrated by deriving the complete set of generators for the unitary groups $U(n)$ and the exceptional Lie groups G_2, F_4, and E_6. A third post-processing algorithm renders the found generators in sparse form. We benchmark the performance improvement of the new algorithms relative to the standard approach. Given the significant complexity of the exceptional Lie groups, our results demonstrate that this machine-learning method for discovering symmetries is completely general and can be applied to a wide variety of labeled datasets.

1. Introduction

The beginning of the 20th century significantly changed theoretical physics. It became evident that the laws of nature are closely tied to principles of symmetry [1]. Emmy Noether taught us that a continuous symmetry within any physical system inherently implies a conservation law, providing profound insights about the system [2]. This concept has deepened our understanding of fundamental physics, from the simplest principles of classical mechanics to the intricacies of the Universe at large. In particle physics, these symmetries serve as the foundation that organizes the particles and their associated interactions. They are instrumental in guiding theoretical physicists in exploring possible extensions of the Standard Model [3,4].

Group theory has traditionally served as the framework for studying symmetries [5]. Among the numerous types of classical Lie groups, the special orthogonal groups $SO(n)$ and the special unitary groups $SU(n)$ are most commonly used in particle physics. While some classical Lie groups describe symmetries observed in nature, the exceptional Lie groups [6] open up new possibilities for theoretical physics, potentially enabling us to describe new kinds of gauge theories, including grand unified theories (GUTs) [7].

There exist five exceptional Lie groups — G_2, F_4, E_6, E_7, and E_8, all of which have found various applications in theoretical physics [8]. For example, the smallest among them, G_2, has been widely used in string theory, e.g., in the studies of seven-dimensional manifolds known as G_2-manifolds, used in M-theory compactifications [9–12]. The next exceptional group, F_4, plays an important role in Jordan algebra theory [13] and has also been used in the context of gauge theories [14,15]. The exceptional Lie group E_6 has been broadly employed in GUTs [16,17]. The two remaining exceptional groups, E_7 and E_8, have also been considered as potential GUT candidates [18,19].

The use of machine learning (ML) to uncover and recognize symmetries in datasets has recently sparked considerable interest [20–31]. Specific applications to group theory include calculating tensor products and branching rules of irreducible representations of Lie groups [32] or testing for the presence of a conjectured Lie group symmetry in the data [25,27]. More recent work has focused on discovering from first principles the Lie group generators reflecting symmetries in the data [28–31]. In this Letter, we introduce new algorithms that significantly enhance the symmetry discovery process outlined in [28–31]. After introducing the problem in Section 2, in Section 3 we describe the new methods and demonstrate their advantages on the unitary groups $U(n)$. Then in Sections 4, 5 and 6 we consecutively consider three of the five exceptional Lie groups: G_2, F_4, and E_6. We derive the complete set of generators, in sparse form, which preserve the respective polynomial invariants. Our approach is completely general, and could be

* Corresponding author.
E-mail address: matchev@ufl.edu (K.T. Matchev).
[1] All authors share equal contributions to this paper.

https://doi.org/10.1016/j.physletb.2023.138266
Received 14 July 2023; Received in revised form 19 September 2023; Accepted 16 October 2023
Available online 20 October 2023

easily extended to the remaining two exceptional Lie groups E_7 and E_8. Section 7 is reserved for our summary.

2. Problem description

The classical groups are the linear groups of transformations over the real numbers $\mathbb{R}$, the complex numbers $\mathbb{C}$, and quaternions $\mathbb{H}$. Consequently, a symmetry transformation operates on a feature vector $\mathbf{x} \equiv \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$, where $\mathbf{x} \in \mathbb{R}^n$ ($\mathbf{x} \in \mathbb{C}^n$) for real (complex) representations. To encapsulate the effect of a group transformation on $\mathbb{R}^n$ or $\mathbb{C}^n$, we examine a representative set of m points $\{\mathbf{x}\} \equiv \left\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\right\}$ sampled from a finite domain. The selection of a sampling distribution, as well as the size and location of the domain, are inconsequential. We adopt a standard normal distribution and pick m on the order of several hundred, depending on the complexity of the problem.

The classical groups can be defined in terms of polynomial invariants over their respective fields. For example, the orthogonal group $O(n)$ preserves the values of the polynomial oracle,

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

the Lorentz group in $n = 4$ dimensions preserves

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

and the unitary group $U(n)$ preserves

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

A symmetry transformation $\mathbf{f}$ is a map $\mathbf{x}' = \mathbf{f}(\mathbf{x})$ that preserves the respective oracle (1)-(3) everywhere, or in our case, for each of the sampled m points:

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

The basic task is to find such a symmetry map $\mathbf{f}$ in parametric form. To focus on the *generators* of the symmetry transformation group, $\mathbf{f}$ is linearized by considering infinitesimal transformations $\delta\mathbf{f}$ in the vicinity of the identity transformation $\mathbb{I}$:

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

where ε is an infinitesimal parameter and $\mathbb{G}$ is an $n \times n$ matrix. After training with the invariance loss function $L_{\rm inv}(\mathbb{G}, \{\mathbf{x}\})$ defined in (A.1), the components of $\mathbb{G}$ evolve towards their *trained* values, thereby producing a valid symmetry generator [29]

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

As shown in the flowchart of Fig. 1, this idea was used in [29] to simultaneously train N_g generators, which were designed to be normalized, orthogonal to each other, and to form a closed algebra, by adding suitable terms to the loss function. However, as indicated in Fig. 1, the resulting set of N_g symmetry generators $\{\mathbb{J}_\alpha\}$ was not sparse. An improvement was implemented in [31], by adding an extra term to the loss function, Eq. (A.4), which encourages the learning of sparse generators in the canonical textbook form.

3. Iterative constructions of symmetry generators

The advantage of learning N_g symmetry generators in one go is that one can impose the closure condition in the loss function and thus guarantee that the learned set $\{\mathbb{J}_\alpha\}$ is closed. However, there are drawbacks to this approach as well. First, due to the large number of trainable parameters ($\sim n^2 \times N_g$), the training is challenging and becomes prohibitively slow for large groups (large N_g) and/or high dimensions (large n). For this reason, in Section 3.1 we explore an alternative "greedy" approach, which trains one generator at a time. Secondly, a closed algebra of generators only exists for certain values of N_g, which are a priori unknown, and with the previous algorithms, would have to be guessed by trial and error. In contrast, the greedy approach automatically finds the largest possible closed set of generators. Furthermore, we can use the closure condition to find additional generators directly, without any training. We refer to this last method as the "Lie bracket trick" (LBT), which is described in Section 3.2. In principle, both of the two new methods could accommodate the sparsity condition (A.4). However, we find that the training is more efficient when we look for nonsparse generators, therefore we postpone their sparsification to a postprocessing step which we describe in Section 3.3.

Algorithm 1: The greedy algorithm.

```
1  Parameters: $\lambda, L_{min}, N_{epochs}$;
2  $\{\mathbb{J}\} \leftarrow []$;
3  $\mathcal{W} \leftarrow \mathcal{W}_{initial} \sim \mathcal{N}$;
4  for $i$ from 1 to $N_{epochs}$ do
5      L $\leftarrow L_{\rm greedy}(\mathbb{G}(\mathcal{W}), \{\mathbb{J}\}, \mathbf{x})$ ;
6      if $L < L_{min}$ then
7          append $\mathbb{G}(\mathcal{W})$ to $\{\mathbb{J}\}$;
8          goto 3;
9      end
10     $\mathcal{W} \leftarrow \mathcal{W} - \lambda \nabla_{\mathcal{W}} L_{\rm greedy}$;
11 end
12 stop
```

3.1. Greedy algorithm

The basic idea of the greedy algorithm is illustrated in Fig. 1 and the corresponding pseudocode is shown in Algorithm 1. We use Eq. (6) to train one generator $\mathbb{G}(\mathcal{W})$ at a time, where $\mathcal{W}$ denotes the trainable parameters of the matrix $\mathbb{G}$. The generator $\mathbb{G}$ is required to i) preserve the oracle, ii) be normalized, and iii) be orthogonal to the set of generators $\{\mathbb{J}_\alpha\}$ already found so far. Therefore, the loss function is

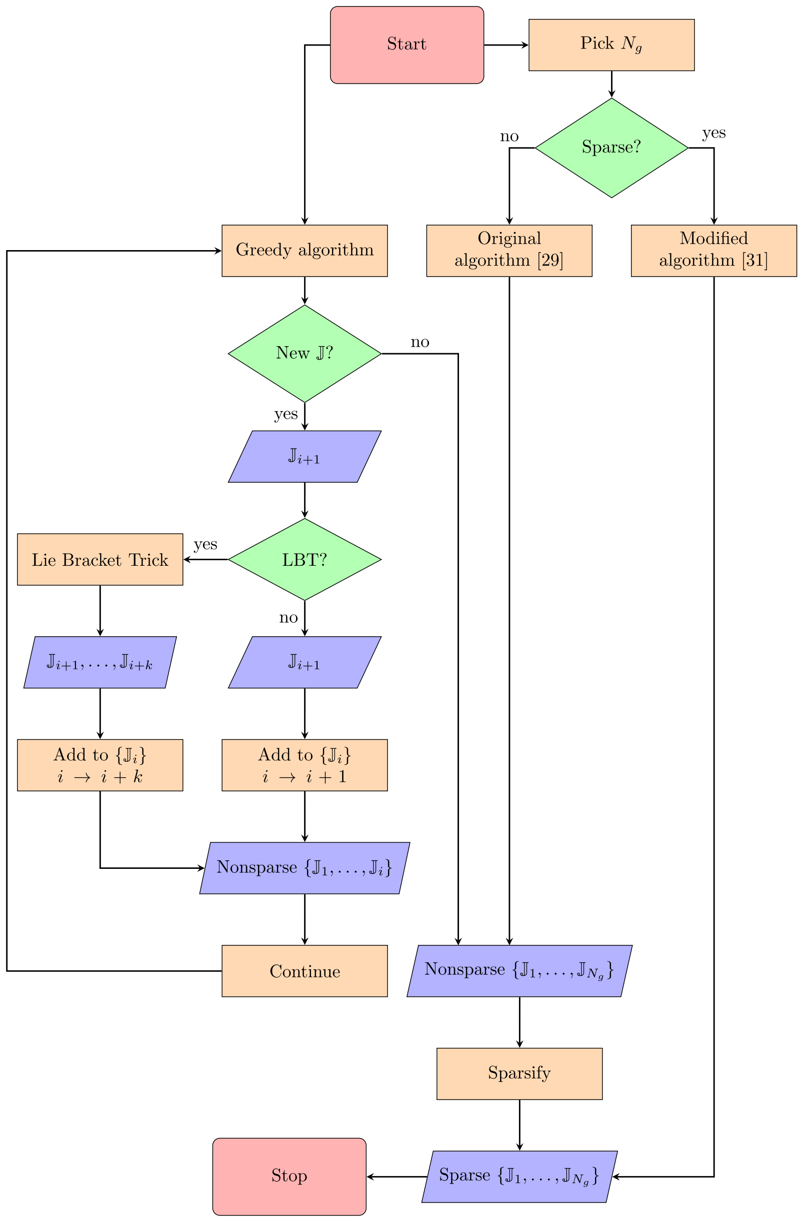
$$L_{\rm greedy}(\mathbb{G}(\mathcal{W}), \{\mathbb{J}\}, \{\mathbf{x}\}) = L_{\rm inv} + L_{\rm norm} + L_{\rm ortho}, \tag{7}$$

where $L_{\rm inv}$, $L_{\rm norm}$ and $L_{\rm ortho}$ are defined in (A.1), (A.2) and (A.3), respectively.

As shown in Fig. 1, at the start of the algorithm the learned set $\{\mathbb{J}\}$ is empty ($i = 0$). At this point, the orthogonality loss $L_{\rm ortho}$ is not applicable and is turned off. The minimization in Eq. (6) will yield a single new symmetry generator $\mathbb{J}_{\rm new} = \mathbb{J}_1$, the first to be added to the learned set $\{\mathbb{J}\}$. Now the orthogonality loss $L_{\rm ortho}$ is turned on and the process continues until the algorithm fails to find a new valid symmetry generator $\mathbb{J}_{\rm new}$. The resulting set $\{\mathbb{J}_1, \ldots, \mathbb{J}_{N_g}\}$ is the largest orthogonal set of nontrivial symmetry generators. For the orthogonal groups $O(n)$ with the oracle (1), this procedure terminates when $N_g = n(n-1)/2$, while for the unitary groups $U(n)$ with the oracle (3), it stops at $N_g = n^2$.

3.2. Lie bracket trick

Algorithm 2 shows the pseudocode for the Lie bracket trick algorithm, which leverages the existing group structure among the symmetry generators. As shown with the left side branch in Fig. 1, the LBT can be (optionally) applied in conjunction with the greedy algorithm, once two orthogonal generators, $\mathbb{J}_1$ and $\mathbb{J}_2$, have been found. At that point, one can compute their commutator (Lie bracket) $\left[\mathbb{J}_1, \mathbb{J}_2\right]$, with three possible outcomes. First, if the commutator is zero, we gain nothing from the LBT and must return to the greedy algorithm. If, however, the chosen pair does not commute, the result may include a component outside the span of the generators discovered so far, which would lead to a new valid symmetry generator that we can extract for free. To isolate this component, we employ Gram-Schmidt orthogonalization (line 11 in Algorithm 2). If the result is zero, we are again out of luck and must return to the greedy method. However, if the out-of-span component is nonzero, it becomes (after the normalization in line 13) a new



Start
Pick N_g
Sparse?
no
yes
Greedy algorithm
Original algorithm [29]
Modified algorithm [31]
New $\mathbb{J}$?
no
yes
$\mathbb{J}_{i+1}$
LBT?
yes
Lie Bracket Trick
no
$\mathbb{J}_{i+1},\ldots,\mathbb{J}_{i+k}$
$\mathbb{J}_{i+1}$
Add to $\{\mathbb{J}_i\}$ $i \to i+k$
Add to $\{\mathbb{J}_i\}$ $i \to i+1$
Nonsparse $\{\mathbb{J}_1,\ldots,\mathbb{J}_i\}$
Continue
Nonsparse $\{\mathbb{J}_1,\ldots,\mathbb{J}_{N_g}\}$
Sparsify
Stop
Sparse $\{\mathbb{J}_1,\ldots,\mathbb{J}_{N_g}\}$

Fig. 1. Flowchart illustrating the algorithms discussed in Section 3 (left side) and the algorithms from Refs. [29] and [31] (right side).

generator which can be added to the set $\{J\}$. This process can be repeated for every pair of known generators, including the newly found ones via the LBT method. The LBT algorithm terminates when all possible Lie brackets close in the current set $\{\mathbb{J}\}$. We note that when using the two Algorithms 1 and 2 together, one should choose judiciously the respective values of the loss thresholds L_{min}.

3.3. Sparsification

Suppose we have already found a set of N_g generators J_α, $\alpha = 1,2,\ldots,N_g$. We can transform them to a new sparse basis, $\tilde{J}_\alpha$, by rotating with an orthogonal $N_g \times N_g$ matrix O,

$$\tilde{J}_\alpha(O) = O_{\alpha\beta} J_\beta. \tag{8}$$

In analogy to (A.4), the loss function for the sparsification of the generators can be defined as

$$L_{\rm sp}(O) = \sum_{\alpha=1}^{N_g} \sum_{\substack{j,j'=1\\ k,k'=1}}^{n} \left| \tilde{J}_\alpha^{(jk)}(O) \tilde{J}_\alpha^{(j'k')}(O) \right| \left(1 - \delta_{jj'}\delta_{kk'}\right). \tag{9}$$

It takes into account all possible pairs of entries in each transformed generator $\tilde{J}_\alpha$. We minimize over this loss to obtain the desired orthogonal transformation O, and subsequently, apply this transformation to the original generators J_α to find the sparse generators $\tilde{J}_\alpha$ as in (8).

3.4. Timing tests

The main advantage of the two new algorithms is that they significantly speed up the training procedure. To quantify this improvement, in Fig. 2 we present the results from timing tests on a personal laptop for different $U(n)$ groups, using the three approaches discussed earlier: the standard algorithm [31] (blue squares), the greedy Algorithm 1 (orange diamonds) and the LBT Algorithm 2 (green circles). The plot shows the time in seconds that it took to learn all n^2 generators for the $U(n)$ group,

Algorithm 2: The Lie bracket trick (LBT) algorithm.

```
Input: $\{J_1, \dots, J_i\}$: known algebra; $J_{i+1}$: new generator;
append $J_{i+1}$ to $G$;
repeat
    $k \leftarrow |G|$;
    $i \leftarrow |J|$;
    append G to J;
    clear G;
    for $p$ from $1$ to $i$ do
        for $q$ from $i+1$ to $i+k$ do
            $C \leftarrow J_p J_q - J_q J_p$;
            $C \leftarrow C - \sum_{g \in J} \frac{g}{||g||} \times (C \cdot g)$;
            if $||C|| \neq 0$ then
                $C \leftarrow \frac{C}{||C||}$;
                if $L_{inv}(C, \mathbf{x}) < L_{min}$ then
                    append C to G;
                end
            end
        end
    end
until $|G| = 0$;
```

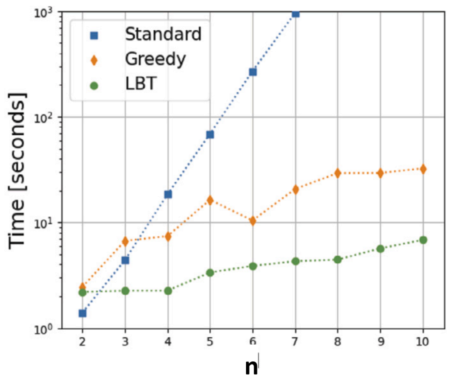


Fig. 2. Comparison of the running times of the different algorithms in finding the full algebras of the $U(n)$ family: the standard algorithm [31] (blue squares), the greedy Algorithm 1 (orange diamonds) and the Lie bracket trick Algorithm 2 (green circles).

as a function of the dimension n, at a learning rate 2.5×10^{-2}. The generators were found in sparse form by applying the post-processing step from Section 3.3, whose duration was included in the total time shown in Fig. 2. We observe that for small n the performance of all three methods is comparable, but for large n the new methods offer significant improvement. In particular, for $n \sim 10$ the standard method would require training for days, while the new methods reduce the training time to less than a minute.

4. The exceptional group G_2

The ML approach described in the previous two sections can be used to discover the orthogonal groups $O(n)$ [29] and the unitary groups $U(n)$ [31]. The method can also be generalized to the case of vector (i.e., multicomponent) oracles [30]. We shall now apply it to exceptional (non-classical) Lie algebras, which have relatively large number of generators, and would benefit from the speed-up offered by the new algorithms. We consider three of the five exceptional Lie groups: G_2 in this section, F_4 in Section 5, and E_6 in Section 6. These exceptional Lie groups have found various applications in high energy physics in the context of gauge theories and model building [8,14,16,33–39]. Given the significant mathematical complexity of these groups, our ability to successfully derive their algebras attests to the robustness and generality of our ML techniques.

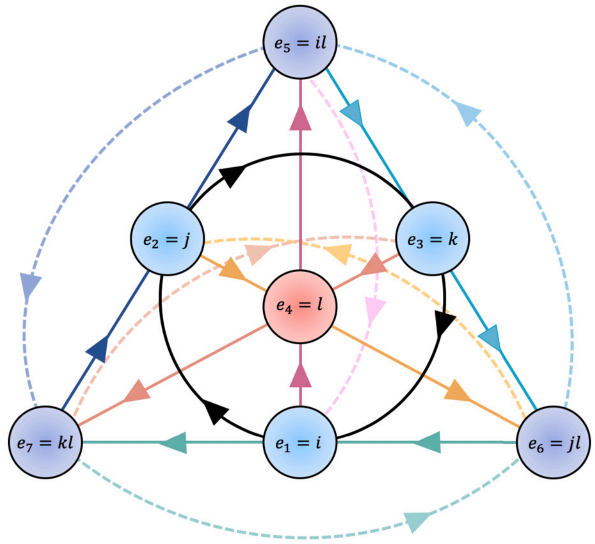


Fig. 3. Fano plane illustrating the multiplication rules for the imaginary unit octonions $e_1, \dots, e_7$ in our conventions. For each triple e_i, e_j and e_k connected by a solid line, the result of the multiplication $e_i e_j$ is equal to $+e_k$ ($-e_k$) when going along (against) the arrows. The dashed lines have been added to guide the eye in following each triplet cycle.

The smallest exceptional Lie group is the G_2 group, which has rank 2 and dimension 14. This group emerges as the automorphism group of the octonion algebra [40]. An octonion $\mathbf{o}$ is a linear combination

$$\mathbf{o} = \sum_{i=0}^{7} x^{(i)} e_i \tag{10}$$

of the unit octonions e_i with real coefficients $x^{(i)}$. Here e_0 is the real element which obeys $e_0^2 = +1$ and can be identified with the real number 1. The remaining $e_1, \dots, e_7$ are the seven imaginary unit octonions which obey $e_i^2 = -1$. Their multiplication rules can be visualized in the Fano plane of Fig. 3. The figure shows only one of 480 possible definitions for octonion multiplication with $e_0 = 1$; the other definitions are isomorphic and can be obtained by permuting and/or changing the signs of the imaginary basis elements.

The group G_2 has a fundamental representation of dimension 7, hence in this section $\mathbf{x} \in \mathbb{R}^7$. The components of the feature vector $\mathbf{x}$ will be identified with the coefficients $x^{(i)}$, $i = 1, \dots, 7$, of the *imaginary* unit octonions in (10). Correspondingly, the generators $\{J\}$ will be real 7×7 matrices.

In general, the exceptional groups preserve K vector oracles $\varphi^{(1)}, \dots, \varphi^{(K)}$, where each component $\varphi^{(i)}$ represents an invariant polynomial characteristic of the group. The group G_2 preserves $K = 2$ such polynomials. The first one is the norm of a purely imaginary octonion,

$$\varphi^{(1)}_{G_2}(\mathbf{x}) = \sum_{i=1}^{7} \left(x^{(i)}\right)^2. \tag{11}$$

To define the second oracle, we need to introduce the real part of the product of three octonions $\mathbf{o}_1$, $\mathbf{o}_2$ and $\mathbf{o}_3$

$$\mathrm{Re}\left(\mathbf{o}_1 \mathbf{o}_2 \mathbf{o}_3\right) = \sum_{i,j,k=0}^{7} \mathcal{D}_{ijk}\, x_1^{(i)} x_2^{(j)} x_3^{(k)}. \tag{12}$$

Using the multiplication rules from Fig. 3, the components of the rank three tensor $\mathcal{D}_{ijk}$ can be easily derived and are shown in Fig. 4. The group G_2 preserves the real component of the product of three *purely imaginary* octonions, hence the second G_2 oracle is (note that the sums start from 1 instead of 0)

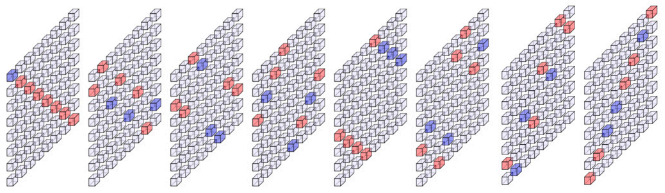


Fig. 4. A pictorial visualization of the rank three tensor $\mathcal{D}_{ijk}$ defined in Eq. (12). Each plane represents a slice at a fixed $i = 0, 1, \ldots, 7$ (from left to right). The blue, red and grey boxes indicate coefficient values of $+1$, -1 and 0, respectively.

$$\varphi^{(2)}_{G_2}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = \sum_{i,j,k=1}^{7} \mathcal{D}_{ijk}\, x_1^{(i)} x_2^{(j)} x_3^{(k)}. \tag{13}$$

For the training of the G_2 generators, we use $m = 900$ samples (7-dimensional vectors in $\mathbb{R}^7$) and fix $\varepsilon = 10^{-3}$. For the computation of the second oracle (13) we split the sample into three equally sized groups of 300, from which we draw the three vectors $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$. The training was done with the Adam optimizer [41] for 1,000 epochs and with learning rate of 2.5×10^{-2}.

Our results for the learned G_2 generators are shown in Fig. 5. The panels in the top two rows depict the 14 generators found by the greedy algorithm (as expected, the algorithm failed to find a valid 15th generator, where even after 10,000 epochs, the loss stayed of order 1). The panels in the bottom two rows show the corresponding results after applying the sparsification procedure of Section 3.3. Note that all found generators are antisymmetric (G_2 is a subgroup of $SO(7)$), and that they can be organized into 7 pairs which share common matrix elements: $\mathbb{J}_1$ and $\mathbb{J}_9$, $\mathbb{J}_2$ and $\mathbb{J}_7$, $\mathbb{J}_3$ and $\mathbb{J}_4$, $\mathbb{J}_5$ and $\mathbb{J}_{10}$, $\mathbb{J}_6$ and $\mathbb{J}_{12}$, $\mathbb{J}_8$ and $\mathbb{J}_{11}$, $\mathbb{J}_{13}$ and $\mathbb{J}_{14}$. Note that in each pair, one generator has six non-zero elements, while the other has only four. Therefore our version is sparser than the conventional textbook representation, in which each generator has six nonvanishing elements.

5. The exceptional group F_4

In this and the next section we follow the notation of refs. [42] and [43], where the generators for F_4 and E_6 have been explicitly derived. The exceptional group F_4 has rank 4 and dimension 52. It is the automorphism group of the Jordan algebra

$$\mathfrak{h}_3 = \begin{pmatrix} r_1 & \mathbf{o}_1 & \mathbf{o}_2 \\ \mathbf{o}_1^* & r_2 & \mathbf{o}_3 \\ \mathbf{o}_2^* & \mathbf{o}_3^* & r_3 \end{pmatrix}, \tag{14}$$

where r_a are three real numbers ($a = 1, 2, 3$), and $\mathbf{o}_a$ are three octonions [42]. The asterisk notation in (14) stands for octonion conjugation

$$\mathbf{o}^* = x^{(0)} e_0 - \sum_{i=1}^{7} x^{(i)} e_i. \tag{15}$$

The fundamental representation is of dimension 26. Following [42,43], we find it convenient to work in $\mathbb{R}^{27}$ instead and map the components of the feature vector $\mathbf{x}$ onto r_a and $\mathbf{o}_a$ as follows

$$r_1 = x^{(1)}, \qquad \mathbf{o}_1 = \sum_{i=0}^{7} x^{(2+i)} e_i, \tag{16a}$$

$$r_2 = x^{(18)}, \qquad \mathbf{o}_2 = \sum_{i=0}^{7} x^{(10+i)} e_i, \tag{16b}$$

$$r_3 = x^{(27)}, \qquad \mathbf{o}_3 = \sum_{i=0}^{7} x^{(19+i)} e_i. \tag{16c}$$

The group F_4 preserves $K = 3$ different oracles (invariant polynomials), which can be expressed in terms of the variables (16) as follows:

$$\varphi^{(1)}_{F_4}(\mathbf{x}) = \mathrm{Tr}\, \mathfrak{h}_3 = \sum_{a=1}^{3} r_a = x^{(1)} + x^{(18)} + x^{(27)}, \tag{17}$$

$$\begin{aligned} \varphi^{(2)}_{F_4}(\mathbf{x}) &= \mathrm{Tr}\, \mathfrak{h}_3^2 = \sum_{a=1}^{3} \left(r_a^2 + 2\, |\mathbf{o}_a|^2 \right) \\ &= 2 \sum_{i=1}^{27} \left(x^{(i)} \right)^2 - \sum_{i \in \{1,18,27\}} \left(x^{(i)} \right)^2, \end{aligned} \tag{18}$$

$$\begin{aligned} \varphi^{(3)}_{F_4}(\mathbf{x}) &= \det \mathfrak{h}_3 = r_1 r_2 r_3 - \sum_{a=1}^{3} r_a\, |\mathbf{o}_{4-a}|^2 + 2\, \mathrm{Re}\left(\mathbf{o}_3 \mathbf{o}_2^* \mathbf{o}_1 \right) \\ &= x^{(1)} x^{(18)} x^{(27)} - x^{(1)} \sum_{i=19}^{26} \left(x^{(i)} \right)^2 \\ &\quad - x^{(18)} \sum_{i=10}^{17} \left(x^{(i)} \right)^2 - x^{(27)} \sum_{i=2}^{9} \left(x^{(i)} \right)^2 \\ &\quad + 2 \sum_{i,j,k=0}^{7} \mathcal{D}_{ijk}\, x^{(19+i)} x^{(10+j)} x^{(2+k)} (2\delta_{j,10} - 1), \end{aligned} \tag{19}$$

where the tensor $\mathcal{D}_{ijk}$ was defined in Eq. (12) and pictorially illustrated in Fig. 4. The additional factor of $(2\delta_{j,10} - 1)$ in the last line flips the sign of the $x^{(11)}, x^{(12)}, \ldots, x^{(17)}$ factors in the sum and thus accounts for the conjugation of $\mathbf{o}_2$ in the triple product $\mathbf{o}_3 \mathbf{o}_2^* \mathbf{o}_1$.

The results from the training of 52 F_4 generators with the greedy algorithm and with the oracles (17)-(19) are shown in Fig. 6. We used $m = 900$ samples of the $n = 27$ input features $x^{(i)}$, organized as in (16). The training ran over up to 26,000 epochs with the Adam optimizer, $\varepsilon = 10^{-4}$ and learning rate of 5×10^{-4}. A 53rd generator was not found after more than 250,000 epochs.

The numerically derived generators in Fig. 6 can be compared against explicit analytical constructions of the F_4 generators in the literature. For example, we have verified that the set of generators in Fig. 6 is isomorphic (in the sense that the matrix O relating the two sets as in (8) is orthogonal) to the set of fifty two 27×27 matrices $\mathbb{C}_\alpha$ listed in Appendix C of [42]. This comparison is rather nontrivial in light of the following differences between the two studies:

- *Octonion multiplication table.* The multiplication rules satisfied by the imaginary unit octonions e_i' in [42] are different from those of Fig. 3. The two bases are related as $e_1' = e_1$, $e_2' = e_2$, $e_3' = e_4$, $e_4' = e_3$, $e_5' = e_6$, $e_6' = -e_7$, $e_7' = e_5$. As a result, the two sets of generators appear visually different, as the non-vanishing components of the respective matrices are located in different rows and/or columns. As it turns out, the conventions of Fig. 3 happen to produce an aesthetically more pleasing result — note how all non-vanishing entries in Fig. 6 are nicely lined up diagonally, while the corresponding patterns in Appendix C of [42] appear more scattered and chaotic.
- *Normalization.* The F_4 generators $\mathbb{C}_\alpha$ in [42] are normalized as $\mathrm{Tr}(\mathbb{C}_\alpha \mathbb{C}_\beta) = -6\, \delta_{\alpha\beta}$, while those in Fig. 6 are normalized as $\mathrm{Tr}(\mathbb{J}_\alpha \mathbb{J}_\beta^T) = \delta_{\alpha\beta}$ as per (A.2) and (A.3).
- *Basis of* $\mathbb{R}^{27}$. Since the first oracle (17) is linear in $\mathbf{x}$, one can apply an orthogonal rotation to r_1, r_2 and r_3, so that one of the coordinates in the new basis, say the last one, is directly proportional to $\varphi^{(1)}_{F_4}$ [42]. This choice has the advantage that the last row and the last column of each generator matrix are filled with zeros, as in [42], confirming that the fundamental representation of F_4 is 26-dimensional. If we apply the same rotation to the learned generators in Fig. 6, the resulting matrices all have zeros in their last rows and columns as well.

6. The exceptional group E_6

The exceptional group E_6 is of rank 6 and dimension 78. Following [43], we work in the real case, which results in the split form of the E_6 algebra, with signature $(52, 26)$. (By multiplying the 26 added generators by i, the algebra remains real, and the Killing form becomes the compact one. For details, we refer the interested reader to [43].) The

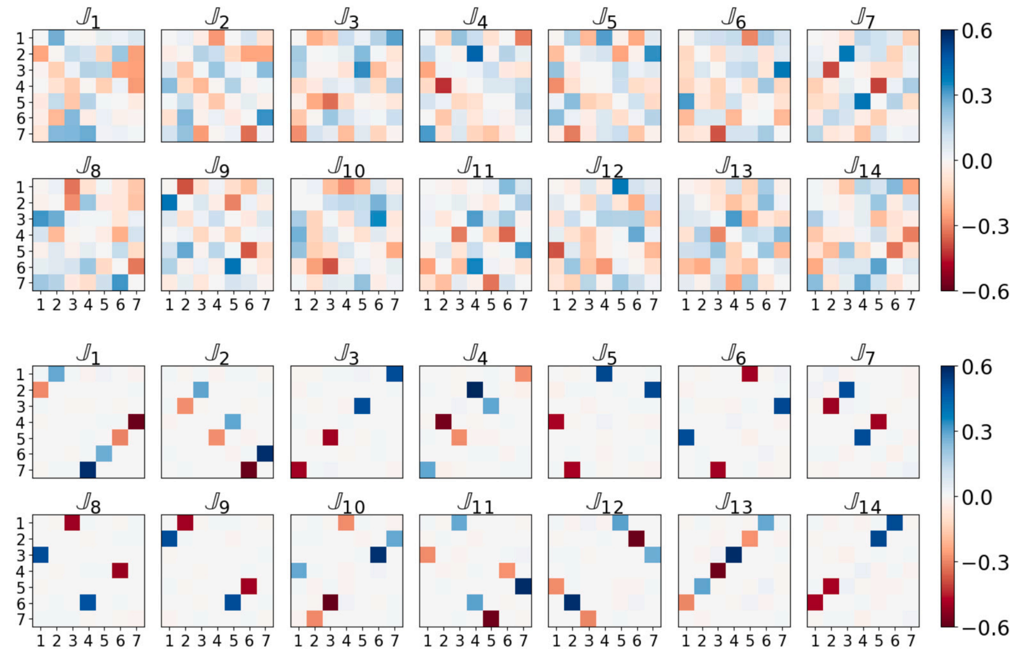


Fig. 5. The fourteen G_2 generators learned with the greedy method (top panels) and the result from their sparsification (bottom panels). In this and all subsequent such figures, each panel represents a learned generator $\mathbb{J}_\alpha$ in matrix form, where the values of the individual elements of the matrix are indicated by the color bar.

relevant 27 variables are those in (16). In this case, we require invariance with respect to only the last oracle, (19), but not the first two, (17) and (18). This allows for the presence of $78 - 52 = 26$ additional generators beyond those of Fig. 6. Our goal in this section will be to derive those additional 26 generators which are not contained in the F_4 subalgebra of E_6.

The straightforward approach to deriving all E_6 generators would be to apply the greedy algorithm and learn from scratch 78 generators preserving the oracle (19). However, we can save a significant amount of work by leveraging the learned F_4 generators from the previous section which already satisfy (19) by definition. In other words, given the 52 already discovered F_4 generators, we are looking to find the additional 26 which complete F_4 to E_6. This is the perfect setup for applying the Lie bracket trick — treat the F_4 generators as the starter set $\{\mathbb{J}_1, \dots, \mathbb{J}_{52}\}$, learn *a single* new non-sparse generator $\mathbb{J}_{53}$ with the greedy algorithm, and then hand those $52 + 1$ generators over to the LBT algorithm and let it do its job. The result from this procedure (after the corresponding sparsification of the 26 new generators among themselves) is shown in Fig. 7. Notably, the LBT algorithm was able to find *all* of the missing 25 generators from a single non-sparse seed $\mathbb{J}_{53}$. The key to this was that the seed $\mathbb{J}_{53}$ was non-sparse, and therefore a linear combination involving a large number of the canonical sparse E_6 generators.

The machine-learned generators shown in Fig. 7 can be verified against analytically derived results in the literature. We have compared our results to the additional 26 E_6 generators $\mathbb{C}_{53}, \dots, \mathbb{C}_{78}$ listed in Appendix C of [43] and found perfect agreement, once we account for the differences in our conventions (see discussion at the end of Section 5). For example, the LBT algorithm correctly finds exactly two diagonal generators, namely $\mathbb{J}_{66}$ and $\mathbb{J}_{72}$ (respectively proportional to $\mathbb{C}_{53}$ and $\mathbb{C}_{70}$ in [43]). This is expected, since the rank of E_6 is larger than the rank of F_4 by 2. Also note that all generators are antisymmetric in the octonionic indices $2-17, 19-26$. Once again, we find regular linear patterns in the locations of the nonzero matrix elements in the generators in Fig. 7.

7. Summary

Discovering symmetries in data is crucial for both fundamental theory and data science. Recent advancements in ML algorithms, coupled with the growth in computational resources, have facilitated progress in this area. However, when dealing with highly complex and multidimensional problems, training machine-learning models remains a significant bottleneck. This paper introduces two novel algorithms that greatly accelerate the symmetry discovery process. The new methods were rigorously tested and showcased using examples of symmetries from the exceptional groups G_2, F_4, and E_6 (the generators for the remaining two exceptional groups, E_7 and E_8, can be learned in a similar fashion, since their polynomial invariants are known as well [44]). Remarkably, the symmetry generators were learned accurately and in a nicer form than the conventional representations found in textbooks. Furthermore, as demonstrated in a follow up paper [45], the method can also reveal the existing subalgebra structures within the set of learned generators. With these groundbreaking techniques at our disposal, we can now confidently tackle even more intricate and challenging problems in mathematical physics, particle phenomenology, and data analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

We are greatly indebted to P. Ramond for setting us on this exceptional path by suggesting this project. We thank S. Gleyzer, R. Houtz, K. Kong, S. Mrenna, H. Prosper and P. Shyamsundar for useful discussions. This work is supported in part by US Department of Energy award DE-SC0022148.

Appendix A. Loss function

In this appendix we adapt the loss functions from [29,31] to the case where we train a single generator $\mathbb{G}$. The loss function is chosen to ensure that $\mathbb{G}$ has the following properties:

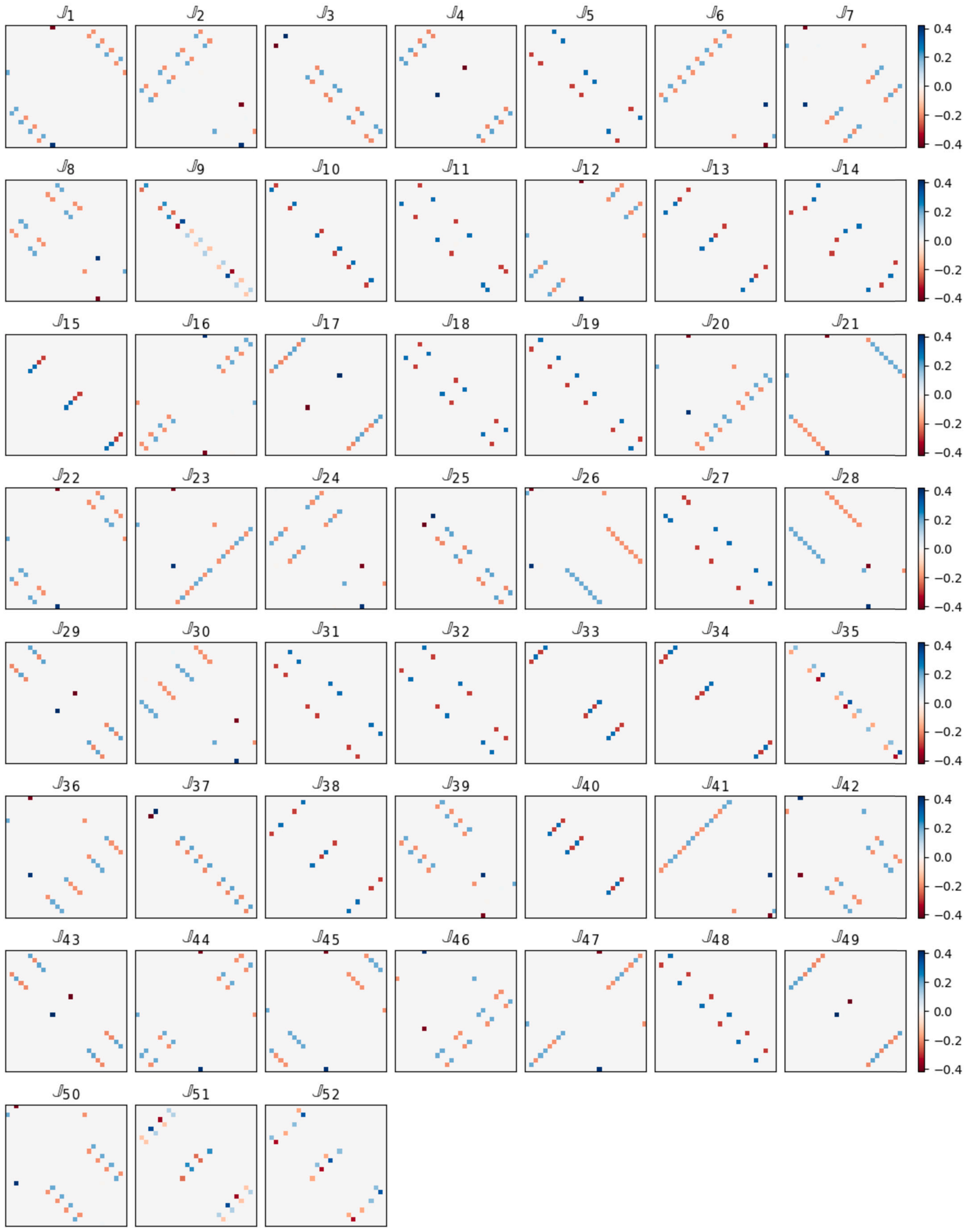


Fig. 6. The learned 27×27 sparse generators J_α, $\alpha = 1, \dots, 52$, for the case of F_4.

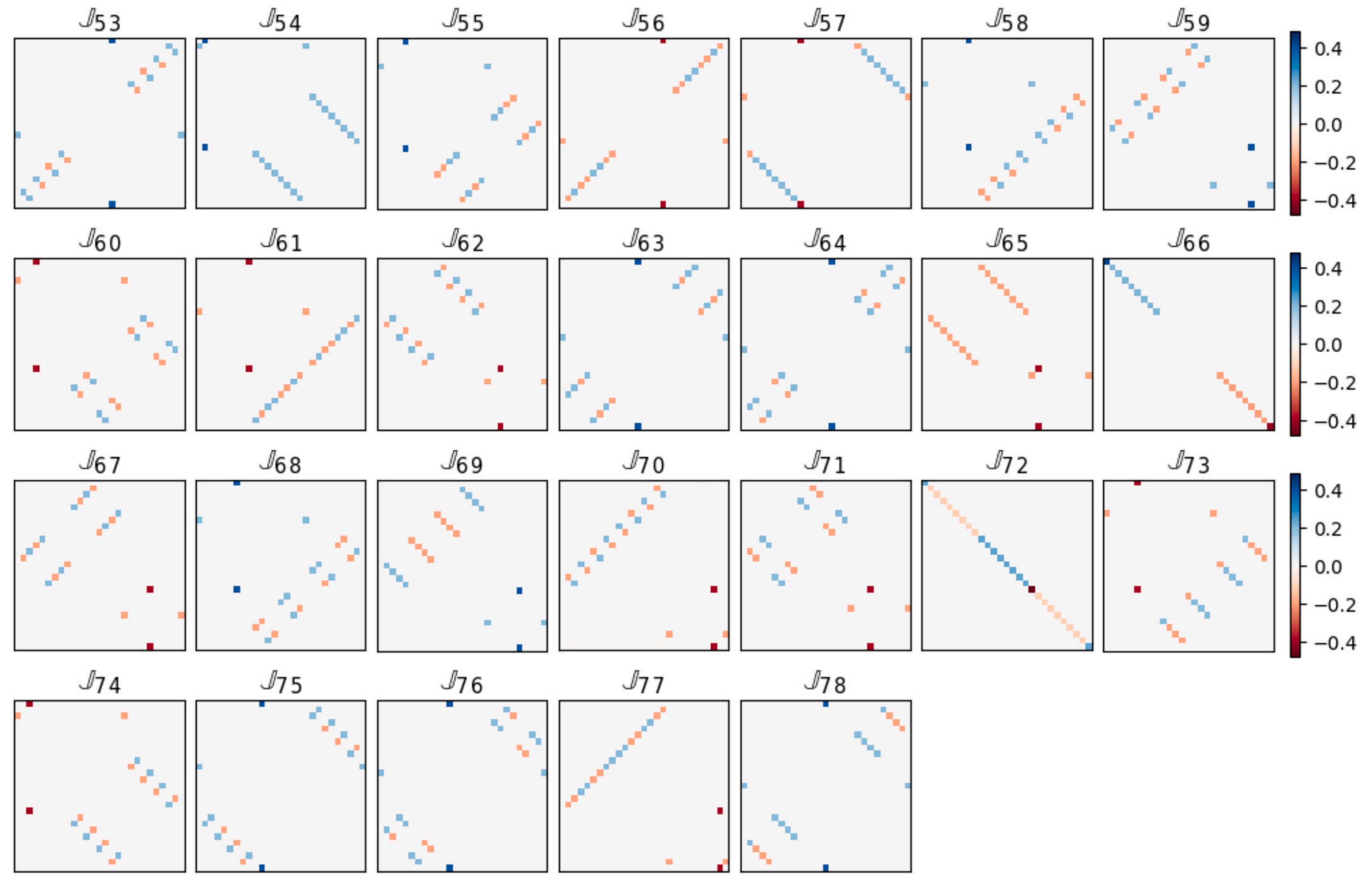


Fig. 7. The additional learned 27×27 sparse generators J_α, $\alpha = 53, \ldots, 78$, for the case of E_6. The hyperparameters used in the training of the single non-sparse generator $\mathbb{J}_{53}$ used to seed the algorithm were as in Section 5.

Invariance: preserving the values of a vector oracle $\vec{\varphi}(\mathbf{x})$ for all sampled datapoints $\{\mathbf{x}\}$ under a given candidate transformation $\mathbb{G}$:

$$L_{\text{inv}}(\mathbb{G}, \{\mathbf{x}\}) = \frac{1}{m\varepsilon^2} \sum_{i=1}^{m} \left[\vec{\varphi}\left(\mathbf{x}_i + \varepsilon \mathbb{G} \cdot \mathbf{x}_i\right) - \vec{\varphi}(\mathbf{x}_i)\right]^2, \tag{A.1}$$

where "·" denotes ordinary tensor multiplication and the arrow vector notation implies summation over the K oracle components.

Normalization: ensuring that the transformation $\mathbb{G}$ is not trivial and normalized to 1:

$$L_{\text{norm}}(\mathbb{G}) = \left[\text{Tr}\left(\mathbb{G} \cdot \mathbb{G}^T\right) - 1\right]^2. \tag{A.2}$$

Orthogonality. This condition is used if we already have some existing orthonormalized generators $\{J\}$ and want to find an additional generator $\mathbb{G}$. It guarantees that the transformation $\mathbb{G}$ is not in the set $\{J\}$:

$$L_{\text{ortho}}(\mathbb{G}, \{\mathbb{J}\}) = \sum_{\alpha} \left[\text{Tr}\left(\mathbb{G} \cdot \mathbb{J}_\alpha^T\right)\right]^2. \tag{A.3}$$

Sparsity: an additional loss term designed to encourage sparsity was introduced in Ref. [31]:

$$L_{\text{sp}}(\mathbb{G}) = \sum_{j,k=1}^{n} \sum_{j',k'=1}^{n} \left|\mathbb{G}^{(jk)} \mathbb{G}^{(j'k')}\right| \left(1 - \delta_{jj'} \delta_{kk'}\right), \tag{A.4}$$

where $\mathbb{G}^{(jk)}$ denotes the jk-component of $\mathbb{G}$.

In this study, the total loss function was formed as a plain sum of the relevant individual terms (A.1)-(A.4).

References

[1] D.J. Gross, The role of symmetry in fundamental physics, Proc. Natl. Acad. Sci. 93 (25) (1996) 14256–14259, https://doi.org/10.1073/pnas.93.25.14256.
[2] E. Noether, Invariante variationsprobleme, Nachr. Ges. Wiss. Gött., Math.-Phys. Kl. 1918 (1918) 235–257.
[3] M.E. Peskin, Beyond the standard model, in: The 1996 European School of High-Energy Physics (Formerly CERN / JINR School of Physics), 1997, pp. 49–142, arXiv:hep-ph/9705479.
[4] C. Csáki, S. Lombardo, O. Telem, TASI Lectures on Non-supersymmetric BSM Models, WSP, 2018, pp. 501–570, arXiv:1811.04279.
[5] E. Wigner, J. Griffin, J. Griffin, Group Theory and Its Application to the Quantum Mechanics of Atomic Spectra, Pure and Applied Physics: a Series of Monographs and Textbooks, Academic Press, 1959.
[6] P. Ramond, Introduction to Exceptional Lie groups and Algebras, Tech. Rep. CALT-68-577, California Institute of Technology, Pasadena, California 91125, 1976.
[7] R. Slansky, Group theory for unified model building, Phys. Rep. 79 (1981) 1–128, https://doi.org/10.1016/0370-1573(81)90092-2.
[8] P. Ramond, Exceptional groups and physics, arXiv:hep-th/0301050, 2003.
[9] B.S. Acharya, M theory, Joyce orbifolds and super Yang-Mills, Adv. Theor. Math. Phys. 3 (1999) 227–248, https://doi.org/10.4310/ATMP.1999.v3.n2.a3, arXiv:hep-th/9812205.
[10] B.S. Acharya, S. Gukov, M theory and singularities of exceptional holonomy manifolds, Phys. Rep. 392 (2004) 121–189, https://doi.org/10.1016/j.physrep.2003.10.017, arXiv:hep-th/0409191.
[11] M. Atiyah, E. Witten, M theory dynamics on a manifold of G(2) holonomy, Adv. Theor. Math. Phys. 6 (2003) 1–106, https://doi.org/10.4310/ATMP.2002.v6.n1.a1, arXiv:hep-th/0107177.
[12] J. Halverson, D.R. Morrison, On gauge enhancement and singular limits in G_2 compactifications of M-theory, J. High Energy Phys. 04 (2016) 100, https://doi.org/10.1007/JHEP04(2016)100, arXiv:1507.05965.
[13] S. Catto, Y.S. Choun, L. Kurt, Invariance properties of the exceptional quantum mechanics (F_4) and its generalization to complex Jordan algebras (E_6), Springer Proc. Math. Stat. 36 (2013) 469–475, https://doi.org/10.1007/978-4-431-54270-4_34.
[14] A. Shahlaei, S. Rafibakhsh, F_4, E_6 and G_2 exceptional gauge groups in the vacuum domain structure model, Phys. Rev. D 97 (5) (2018) 056015, https://doi.org/10.1103/PhysRevD.97.056015, arXiv:1802.02905.

[15] S. Rafibakhsh, A. Shahlaei, Confinement in F_4 exceptional gauge group using domain structures, EPJ Web Conf. 137 (2017) 13013, https://doi.org/10.1051/epjconf/201713713013.

[16] F. Gursey, P. Ramond, P. Sikivie, A universal gauge theory model based on E6, Phys. Lett. B 60 (1976) 177–180, https://doi.org/10.1016/0370-2693(76)90417-2.

[17] D. Croon, T.E. Gonzalo, L. Graf, N. Košnik, G. White, GUT physics in the era of the LHC, Front. Phys. 7 (2019) 76, https://doi.org/10.3389/fphy.2019.00076, arXiv:1903.04977.

[18] F. Gursey, P. Sikivie, E(7) as a universal gauge group, Phys. Rev. Lett. 36 (1976) 775, https://doi.org/10.1103/PhysRevLett.36.775.

[19] I. Bars, M. Gunaydin, Grand unification with the exceptional group E8, Phys. Rev. Lett. 45 (1980) 859, https://doi.org/10.1103/PhysRevLett.45.859.

[20] R. Iten, T. Metger, H. Wilming, L. del Rio, R. Renner, Discovering physical concepts with neural networks, Phys. Rev. Lett. 124 (2020) 010508, https://doi.org/10.1103/PhysRevLett.124.010508, https://link.aps.org/doi/10.1103/PhysRevLett.124.010508.

[21] S. Krippendorf, M. Syvaeri, Detecting symmetries with, Neural Netw. 3 (2020), arXiv:2003.13679.

[22] Z. Liu, M. Tegmark, Machine learning conservation laws from trajectories, Phys. Rev. Lett. 126 (18) (2021) 180604, https://doi.org/10.1103/PhysRevLett.126.180604, arXiv:2011.04698.

[23] G. Barenboim, J. Hirn, V. Sanz, Symmetry meets AI, SciPost Phys. 11 (2021) 014, https://doi.org/10.21468/SciPostPhys.11.1.014, arXiv:2103.06115.

[24] B.M. Dillon, G. Kasieczka, H. Olischlager, T. Plehn, P. Sorrenson, L. Vogel, Symmetries, safety, and self-supervision, SciPost Phys. 12 (6) (2022) 188, https://doi.org/10.21468/SciPostPhys.12.6.188, arXiv:2108.04253.

[25] Z. Liu, M. Tegmark, Machine learning hidden symmetries, Phys. Rev. Lett. 128 (18) (2022) 180201, https://doi.org/10.1103/PhysRevLett.128.180201, arXiv:2109.09721.

[26] K. Desai, B. Nachman, J. Thaler, Symmetry discovery with deep learning, Phys. Rev. D 105 (9) (2022) 096031, https://doi.org/10.1103/PhysRevD.105.096031, arXiv:2112.05722.

[27] S. Craven, D. Croon, D. Cutting, R. Houtz, Machine learning a manifold, Phys. Rev. D 105 (9) (2022) 096030, https://doi.org/10.1103/PhysRevD.105.096030, arXiv:2112.07673.

[28] A. Moskalev, A. Sepliarskaia, I. Sosnovik, A. Smeulders, Liegg: studying learned Lie group generators, arXiv:2210.04345, https://doi.org/10.48550/ARXIV.2210.04345, https://arxiv.org/abs/2210.04345, 2022.

[29] R.T. Forestano, K.T. Matchev, K. Matcheva, A. Roman, E.B. Unlu, S. Verner, Deep learning symmetries and their Lie groups, algebras, and subalgebras from first principles, Mach. Learn.: Sci. Technol. 4 (2) (2023) 025027, https://doi.org/10.1088/2632-2153/acd989, arXiv:2301.05638.

[30] A. Roman, R.T. Forestano, K.T. Matchev, K. Matcheva, E.B. Unlu, Oracle-preserving latent flows, Symmetry 15 (7) (2023), https://doi.org/10.3390/sym15071352, https://www.mdpi.com/2073-8994/15/7/1352.

[31] R.T. Forestano, K.T. Matchev, K. Matcheva, A. Roman, E.B. Unlu, S. Verner, Discovering sparse representations of Lie groups with machine learning, Phys. Lett. B 2 (2023), arXiv:2302.05383.

[32] H.-Y. Chen, Y.-H. He, S. Lal, S. Majumder, Machine learning Lie structures & applications to physics, Phys. Lett. B 817 (2021) 136297, https://doi.org/10.1016/j.physletb.2021.136297, arXiv:2011.00871.

[33] K. Holland, P. Minkowski, M. Pepe, U.J. Wiese, Exceptional confinement in G(2) gauge theory, Nucl. Phys. B 668 (2003) 207–236, https://doi.org/10.1016/S0550-3213(03)00571-6, arXiv:hep-lat/0302023.

[34] S. Deldar, H. Lookzadeh, S.M. Hosseini Nejad, Confinement in G(2) gauge theories using thick center vortex model and domain structures, Phys. Rev. D 85 (2012) 054501, https://doi.org/10.1103/PhysRevD.85.054501, arXiv:1112.4963.

[35] S.M. Hosseini Nejad, S. Deldar, Role of the SU(2) and SU(3) subgroups in observing confinement in the G(2) gauge group, Phys. Rev. D 89 (1) (2014) 014510, https://doi.org/10.1103/PhysRevD.89.014510, arXiv:1401.3968.

[36] T. Deppisch, E6Tensors: a Mathematica package for E6 tensors, Comput. Phys. Commun. 213 (2017) 130–135, https://doi.org/10.1016/j.cpc.2016.09.010, arXiv:1605.05920.

[37] I. Todorov, S. Drenska, Octonions, exceptional Jordan algebra and the role of the group F_4 in particle physics, Adv. Appl. Clifford Algebras 28 (4) (2018) 82, https://doi.org/10.1007/s00006-018-0899-y, arXiv:1805.06739.

[38] I. Todorov, M. Dubois-Violette, Deducing the symmetry of the standard model from the automorphism and structure groups of the exceptional Jordan algebra, Int. J. Mod. Phys. A 33 (20) (2018) 1850118, https://doi.org/10.1142/S0217751X1850118X, arXiv:1806.09450.

[39] D. Corradetti, Complexification of the exceptional Jordan algebra and its application to particle physics, J. Geom. Symmetry Phys. 61 (2021) 1–16, https://doi.org/10.7546/jgsp-61-2021-1-16.

[40] J.C. Baez, The octonions, Bull. Am. Math. Soc. 39 (2002) 145–205, https://doi.org/10.1090/S0273-0979-01-00934-X, Erratum: Bull. Am. Math. Soc. 42 (2005) 213, arXiv:math/0105155.

[41] D.P. Kingma, J. Ba Adam, A method for stochastic optimization, arXiv:1412.6980, 2017.

[42] F. Bernardoni, S.L. Cacciatori, B.L. Cerchiai, A. Scotti, Mapping the geometry of the F(4) group, Adv. Theor. Math. Phys. 12 (4) (2008) 889–994, https://doi.org/10.4310/ATMP.2008.v12.n4.a6, arXiv:0705.3978.

[43] F. Bernardoni, S.L. Cacciatori, B.L. Cerchiai, A. Scotti, Mapping the geometry of the E(6) group, J. Math. Phys. 49 (2008) 012107, https://doi.org/10.1063/1.2830522, arXiv:0710.0356.

[44] V. Talamini, Flat bases of invariant polynomials and P-matrices of E7 and E8, J. Math. Phys. 51 (2010) 023520, https://doi.org/10.1063/1.3272569, arXiv:1003.1095.

[45] R.T. Forestano, K.T. Matchev, K. Matcheva, A. Roman, E.B. Unlu, S. Verner, Identifying the group-theoretic structure of machine-learned symmetries, arXiv:2309.07860, 2023.