

PAPER • OPEN ACCESS

Machine learning for detection of equivariant finite symmetry groups in dynamical systems

To cite this article: Pablo Calvo-Barlés *et al* 2025 *Mach. Learn.: Sci. Technol.* **6** 025058

View the [article online](#) for updates and enhancements.

You may also like

- [NCoder—a quantum field theory approach to encoding data](#)
D S Berman, M S Klinger and A G Stapleton
- [Towards harmonization of SO\(3\)-equivariance and expressiveness: a hybrid deep learning framework for electronic-structure Hamiltonian prediction](#)
Shi Yin, Xinyang Pan, Xudong Zhu et al.
- [Physical applications of GPS geodesy: a review](#)
Yehuda Bock and Diego Melgar



PAPER

OPEN ACCESS

RECEIVED
4 March 2025REVISED
27 May 2025ACCEPTED FOR PUBLICATION
3 June 2025PUBLISHED
12 June 2025

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Machine learning for detection of equivariant finite symmetry groups in dynamical systems

Pablo Calvo-Barlés^{1,2} , Sergio G Rodrigo^{1,3} and Luis Martín-Moreno^{1,2,*} ¹ Instituto de Nanociencia y Materiales de Aragón (INMA), CSIC-Universidad de Zaragoza, 50009 Zaragoza, Spain² Departamento de Física de la Materia Condensada, Universidad de Zaragoza, 50009 Zaragoza, Spain³ Departamento de Física Aplicada, Universidad de Zaragoza, 50009 Zaragoza, Spain

* Author to whom any correspondence should be addressed.

E-mail: imm@unizar.es**Keywords:** symmetry detection, machine learning, dynamical systems, equivariance

Abstract

In this work, we introduce the equivariance seeker model (ESM), a data-driven method for discovering the underlying finite equivariant symmetry group of an arbitrary function. ESM achieves this by optimizing a loss function that balances equivariance preservation with the penalization of redundant solutions, ensuring the complete and accurate identification of all symmetry transformations. We apply this framework specifically to dynamical systems, identifying their symmetry groups directly from observed trajectory data. To demonstrate its versatility, we test ESM on multiple systems in two distinct scenarios: (i) when the governing equations are known theoretically and (ii) when they are unknown, and the equivariance finding relies solely on observed data. The latter case highlights ESM's fully data-driven capability, as it requires no prior knowledge of the system's equations to operate.

1. Introduction

Symmetries are fundamental to the physical sciences, serving as powerful tools to reveal the underlying laws governing physical phenomena [1]. In mathematics, they play a crucial role in the theory of integrable systems, where they are employed to derive conservation laws [2], reduce the complexity of differential equations, and establish a definition of integrability [3–5]. Discovering the underlying symmetries of a physical system when they are not explicitly known is both highly valuable and very challenging, especially when only observational data is available. Machine learning (ML), and particularly neural networks (NNs), provides a framework for automatic symmetry discovery due to its capacity to identify patterns from data. Recent advancements have shown that ML models can uncover physical insights from data in unknown or partially known physical systems [6, 7], including governing equations of dynamical systems [8, 9] or physically relevant quantities [16–18]. ML models have also been used to discover conservation laws from different levels of available information: (i) the system's differential equations together with a predefined basis of functions that span the constants of motion [12]; (ii) only the differential equations [11]; or (iii) solely trajectory data [10, 13, 15].

Various aspects of ML-assisted symmetry discovery have been explored using different strategies. For instance, NNs have been used for finding coordinate transformations that reveal hidden symmetries [19, 20], learning transformations that preserve the statistical distribution of data sets [21, 22], recognizing symmetries from embedding layers [23, 24] and identifying pairs of symmetric events [14, 25]. Much of the research in automatic symmetry discovery has concentrated on continuous symmetry groups, particularly the learning of generators associated with Lie groups using different methods, such as fully-connected NNs [27, 29, 31], autoencoders [26], Lie convolutional NNs [30] and Neural Ordinary Differential equations [28]. However, there has been comparatively less focus on discovering finite symmetry groups. Some methods for continuous group detection have demonstrated the capability to identify individual transformations of finite groups [22], but they do not guarantee the identification of all the elements within the group. In contrast, the

discovery of a finite symmetry group has been addressed in [32], but the method is limited to invariance detection, that is, to identify transformations in mappings $\vec{y}(\vec{x})$ that act on the input \vec{x} and leave the output \vec{y} unchanged. The simplest example of an invariant function is that of an even function, $y(-x) = y(x)$. But there are symmetric functions that are not invariant, as for instance an odd function $y(-x) = -y(x)$. This is an example of the generalized family of symmetric functions known as equivariant functions, where applying a transformation to the input induces a corresponding and predictable transformation of the output. Equivariance is a symmetry property that has been widely leveraged in the design of ML models [33, 34, 36]. Exploiting this property improves model efficiency and predictive robustness, and equivariant architectures have been successfully applied across a broad range of scientific problems [35, 37].

In this article, we extend the framework for detecting invariant finite symmetry groups introduced in [32] to encompass equivariance, focusing on the symmetries inherent in the governing equations of dynamical systems. In this context, these symmetries manifest as a specific type of equivariance in which the transformation applied to the input is identical to that applied to the output. Symmetries are particularly relevant in dynamical systems. For example, symmetry considerations can be used to simplify the spectral structure of the Koopman operator—an infinite-dimensional linear operator that evolves measurement functions of state space variables—and calculate more efficiently its eigenmodes [38]. We propose a method to automatically detect the finite equivariant group of linear symmetries in arbitrary nonlinear dynamical systems from collected trajectory data.

The text is structured as follows: in section 2, we state the mathematical definition of the problem we are interested in. Next, in section 3, we introduce the general ML framework for identifying equivariant symmetry groups from data. In section 4, we illustrate our method with a one-dimensional simple example. In section 5, we demonstrate our method's effectiveness on well-known dynamical systems. Finally, we conclude in section 6 with a summary of our findings.

2. Mathematical definition of the problem

We consider nonlinear dynamical systems described by ordinary differential equations:

$$\frac{d\vec{x}}{dt} = \vec{y}(\vec{x}), \quad (1)$$

where $\vec{x}(t) \in \mathbb{R}^n$ denotes the system's state evolving over time t . The time derivative of the state is provided by the nonlinear vector field $\vec{y}(\vec{x}(t)) : \mathbb{R}^n \rightarrow \mathbb{R}^n$, that thus determines the system dynamical evolution.

A linear symmetry of the dynamical system in equation (1) is a matrix $\hat{D} \in \mathbb{R}^{n \times n}$ such that if $\vec{x}(t)$ is a solution, then $\hat{D}\vec{x}(t)$ is also a solution. For the system to exhibit this symmetry, the function \vec{y} must be equivariant under the action of \hat{D} , meaning:

$$\vec{y}(\vec{x}) = \hat{D}^{-1}\vec{y}(\hat{D}\vec{x}), \quad \forall \vec{x} \in \mathbb{R}^n. \quad (2)$$

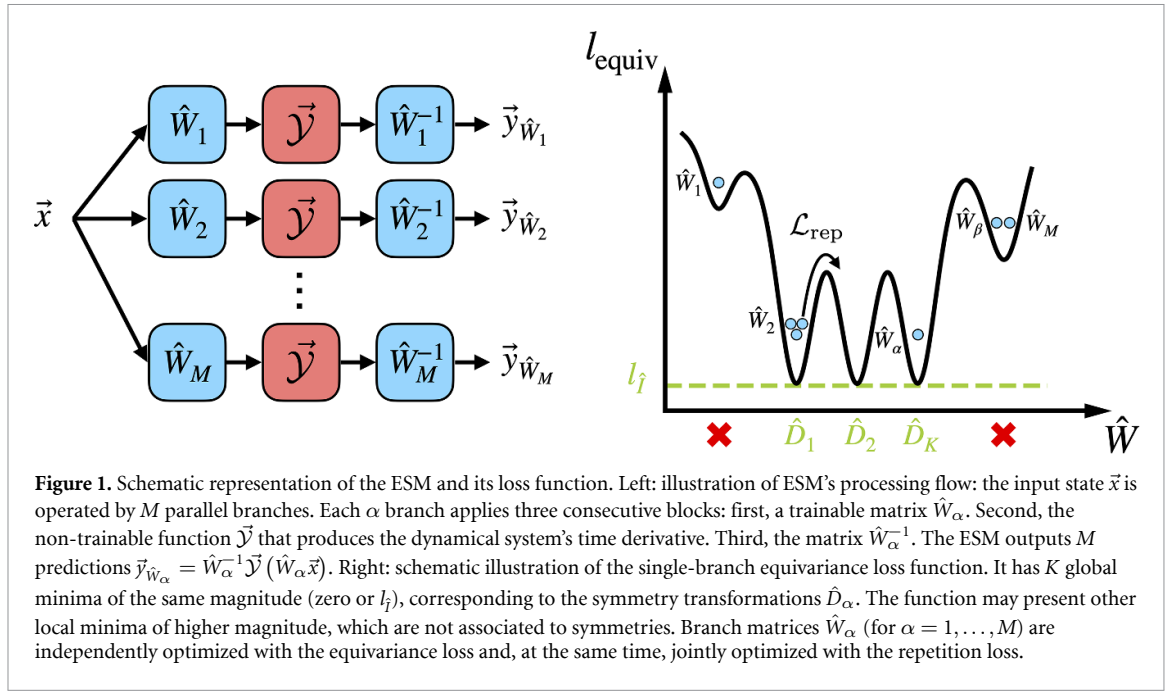
We consider dynamical systems presenting a finite symmetry group of order K . This means that there exists a unique set of matrices $\{\hat{D}_\alpha\}$, for $\alpha = 1, \dots, K$, such that each matrix satisfies the equivariance condition in equation (2).

The goal of this study is to identify all K elements of the symmetry group from a set of N_T trajectories of states $\{\vec{x}^{(r)}(t_s)\}$ and time derivatives $\{\vec{y}^{(r)}(t_s)\}$ ($r = 1, \dots, N_T$). Each trajectory correspond to a different initial condition and is observed at N_S equidistant time steps $t_s = s\Delta t$, with $s = 1, \dots, N_S$ and Δt being the time interval between observations.

The time derivatives $\{\vec{y}^{(r)}(t_s)\}$ can be obtained either analytically (if $\vec{y}(\vec{x})$ is known), measured, or numerically approximated. In this study, we consider two hypothetical scenarios: (1) when the analytical expression of $\vec{y}(\vec{x})$ is known, allowing direct computation of $\vec{y}^{(r)}(t_s) \equiv \vec{y}[\vec{x}^{(r)}(t_s)]$, and (2) a data-driven case where $\vec{y}(\vec{x})$ is unknown, so the derivatives are approximated as $\vec{y}^{(r)}(t_s) \equiv [\vec{x}^{(r)}(t_{s+1}) - \vec{x}^{(r)}(t_s)] / \Delta t$. This last scenario will be used to demonstrate that the method can operate purely in a data-driven manner, without requiring the governing equations.

3. A method for detecting equivariant symmetry groups

In this section we explain the general framework of the technique proposed in this paper to find the symmetry group of a given dynamical system. This includes the ML model architecture, the loss functions and the final refinement used to obtain the results.



3.1. Equivariance seeker model architecture

We introduce the equivariance seeker model (ESM) (depicted in figure 1(a)), which processes an input state \vec{x} through M parallel branches, indexed by $\alpha = 1, \dots, M$ (M is a hyperparameter). Each branch consists of three consecutive blocks: (1) a trainable matrix $\hat{W}_\alpha \in \mathbb{R}^{n \times n}$, (2) a non-trainable block \vec{Y} , identical for all branches, which replicates the system's derivative vector field (equations (1)), and (3) the inverse matrix of the first block, \hat{W}_α^{-1} . After processing, the ESM outputs M predictions, denoted as

$$\vec{y}_{\hat{W}_\alpha}(\vec{x}) = \hat{W}_\alpha^{-1} \vec{Y}(\hat{W}_\alpha \vec{x}), \quad (3)$$

for $\alpha = 1, \dots, M$. Note that the trainable weights of the ESM correspond to the set of branch matrices $\{\hat{W}_\alpha\}$.

The derivative vector field block \vec{Y} is defined based on prior knowledge of the system. If the analytical differential equation (1) is known, $\vec{Y}(\vec{x}) \equiv \vec{y}(\vec{x})$. Otherwise, an oracle model is pre-trained on data pairs $\{\vec{x}^{(r)}(t_s), \vec{y}^{(r)}(t_s)\}$ to predict the derivative from an arbitrarily chosen state, including those not within the training data. In our implementation, we use as oracle a simple feed-forward NN, denoted as \vec{y}_{NN} , such that $\vec{Y}(\vec{x}) \equiv \vec{y}_{\text{NN}}(\vec{x})$. However, the framework is flexible and can incorporate other ML-based oracles [8, 39, 40].

3.2. Equivariance loss function

To identify the ground-truth symmetry transformations $\{\hat{D}_\alpha\}$, we optimize the weights $\{\hat{W}_\alpha\}$ such that each branch approximates the vector field $\vec{y}(\vec{x})$. When this goal is achieved, the converged branch matrices represent symmetry transformations, as they satisfy the equivariance condition given by equation (2).

We define the equivariance loss as

$$\mathcal{L}_{\text{equiv}}(\{\hat{W}_\alpha\}) = \frac{1}{M} \sum_{\alpha=1}^M l_{\text{equiv}}(\hat{W}_\alpha), \quad (4)$$

which depends on the set of all branch matrices and is the average of the single-branch equivariance losses, given by

$$l_{\text{equiv}}(\hat{W}) = \frac{1}{N} \sum_{i=1}^N \|\vec{y}_i - \vec{y}_{\hat{W}}(\vec{x}_i)\|^2, \quad (5)$$

where $\|\cdot\|$ denotes the L_2 norm. The dataset $\{\vec{x}^{(r)}(t_s), \vec{y}^{(r)}(t_s)\}$ is randomly shuffled and split into three parts: two subsets are used for the oracle training and validation, while the third subset is used for ESM training. In the formula above, N represents the number of samples assigned to ESM training and i is an index that combines trajectories and time steps for simplicity.

The function l_{equiv} (sketched in figure 1(b)) depends on a single matrix \hat{W} , is non-negative and reaches its minimum value when \hat{W} is a symmetry transformation. Consequently, l_{equiv} has K global minima, each

corresponding to one of the ground-truth symmetry transformations \hat{D}_α . Indeed, if the analytical derivative is available [$\vec{Y}(\vec{x}) \equiv \vec{y}(\vec{x})$], the value of l_{equiv} in these minima is zero because of equation (2). In the absence of this analytical expression [$\vec{Y}(\vec{x}) \equiv \vec{y}_{\text{NN}}(\vec{x})$], the values of l_{equiv} in global minima become of the order of the oracle's prediction error (which is assumed to be small). Still, we expect all K global minima to have approximately the same value, which can be estimated by evaluating l_{equiv} at the identity transformation \hat{I} , i.e. the trivial symmetry. This value at the identity, that we denote as $l_{\hat{I}}$, coincides with the oracle's mean squared error (MSE),

$$l_{\hat{I}} = l_{\text{equiv}}(\hat{I})|_{\vec{y}=\vec{y}_{\text{NN}}} = \frac{1}{N} \sum_{i=1}^N \|\vec{y}_i - \vec{y}_{\text{NN}}(\vec{x}_i)\|^2. \quad (6)$$

3.3. Repetition loss function

The equivariance loss in equation (4) reaches its minimum when each ESM branch identifies a symmetry transformation, meaning that each \hat{W}_α converges to one of the K global minima of l_{equiv} . However, $\mathcal{L}_{\text{equiv}}$ does not enforce the discovery of all distinct symmetry transformations within the group. For instance, nothing in $\mathcal{L}_{\text{equiv}}$ prevents all branch matrices converging to the same symmetry transformation (e.g. $\hat{W}_\alpha = \hat{I} \forall \alpha$) leaving the remaining $K - 1$ minima unexplored.

We propose a training methodology for the ESM to overcome this limitation and discover the entire finite symmetry group (all K elements) in a single run. To ensure all global minima of l_{equiv} are found, we introduce the so-called repetition loss term (defined below), that penalizes branch matrix repetition during training. First, we assume that the number of branches satisfies $M \geq K$. At the end of the process we will be able to check that the symmetry group has been found, so there is always the possibility to repeat the process with a larger M , if initially the chosen M is not large enough. Conversely, if M is large enough, pruning process removes redundant branches during training, stopping at $M = K$, as will be described later in subsection 3.4.

We introduce the repetition loss term, defined as

$$\mathcal{L}_{\text{rep}}(\{\hat{W}_\alpha\}) = A \sum_{\alpha=1}^M \sum_{\beta=1}^{\alpha-1} \exp\left(-\frac{1}{\sigma} \|\hat{W}_\alpha - \hat{W}_\beta\|_F^2\right), \quad (7)$$

that depends on two hyperparameters: amplitude A and width σ . Here, $\|\cdot\|_F$ denotes the Frobenius norm [41], which can be interpreted as a distance in the matrix space $\mathbb{R}^{n \times n}$. The repetition loss penalizes similarity between branch matrices. It consists of a sum over all distinct indices $\alpha \neq \beta$, with each term being a Gaussian function of the distance between \hat{W}_α and \hat{W}_β . The penalty is maximum when two matrices are identical ($\hat{W}_\alpha = \hat{W}_\beta$) and decreases as their distance increases. Intuitively, if two branch matrices are attracted to the same minimum of l_{equiv} , the repetition loss generates a 'repulsive action' between them, with its strength controlled by A and the action range determined by σ .

The total loss function combines equivariance and repetition losses,

$$\mathcal{L} = \mathcal{L}_{\text{equiv}} + \mathcal{L}_{\text{rep}}. \quad (8)$$

This dual approach promotes symmetry detection in such a way that branch matrices are forced to converge at distinct global minima.

Nevertheless, a naive ESM training with \mathcal{L} using fixed σ and A hyperparameters is not a good solution. The influence of \mathcal{L}_{rep} (controlled by σ and A) is highly sensitive to the l_{equiv} landscape, which depends on the dynamical system and collected data. A small σ limits the range of \mathcal{L}_{rep} , allowing multiple branches to fall into the basin of the same minimum, thus failing to separate branch matrices. Conversely, a large σ produces repulsion even between matrices located in different global minima, moving them away from the minimum of l_{equiv} , thus yielding to inaccurate symmetry prediction.

To overcome the pitfalls of a naive training with \mathcal{L} , we follow a protocol where the hyperparameters σ and A are modified throughout training according to a predefined function of the epochs, gradually diminishing the influence of \mathcal{L}_{rep} until it vanishes. By the end of training, only the equivariance loss remains ($\mathcal{L} \approx \mathcal{L}_{\text{equiv}}$), ensuring the optimization focuses solely on symmetry detection. This protocol will be illustrated in section 4 with a simple example.

3.4. Branch-removal processes: local minima and redundant solutions

As previously mentioned, we expect that the function l_{equiv} presents global minima having—exactly or approximately—the same value for all matrices that are exact symmetries of the dynamical system. Additionally, l_{equiv} may also exhibit local minima for matrices that do not represent symmetries. These local minima appear for loss values significantly larger than those of the global minima. The optimization process

can occasionally cause the branch matrices to become trapped in these local minima, creating an obstacle for identifying true symmetries.

To mitigate this, our method includes a process at the last stages of training (after \mathcal{L}_{rep} vanishes) in which we remove branches that have converged to local minima. For that, we calculate each single-branch equivariance loss, $l_{\text{equiv}}(\hat{W}_\alpha)$, and eliminate those branches α whose values are significantly higher than the function's lower bound. We thus use tolerance $\varepsilon_{\text{equiv}}$ so that, if $l_{\text{equiv}}(\hat{W}_\alpha) > \varepsilon_{\text{equiv}}$, the branch α is removed. The value of this tolerance depends on the available knowledge of the system:

- (i) In the case of $\vec{\mathcal{Y}}(\vec{x}) = \vec{y}(\vec{x})$, since the lower bound of l_{equiv} is zero, we aim to discard any branch matrix \hat{W}_α that does not satisfy $l_{\text{equiv}}(\hat{W}_\alpha) = 0$. However, converged branch matrices do not strictly reach this value in practice due to gradient descent oscillations around the minimum. For this reason, we use a tolerance $\varepsilon_{\text{equiv}} = 10^{-6}$, which is several orders of magnitude smaller than typical values of l_{equiv} in our numerical experiments.
- (ii) In the case of $\vec{\mathcal{Y}}(\vec{x}) = \vec{y}_{\text{NN}}(\vec{x})$, the value $l_{\text{equiv}}(\hat{D}_\alpha)$ of each global minimum is equal to reference value l_f given by the identity matrix, up to a fluctuation. This fluctuation arises because calculating $l_{\text{equiv}}(\hat{D}_\alpha)$ requires the interpolation of \vec{y}_{NN} at states $\hat{D}_\alpha \vec{x}_i$, which do not belong to the original ESM training set. Therefore, the quantity $\|\vec{y}_i - \hat{D}_\alpha^{-1} \vec{y}_{\text{NN}}(\hat{D}_\alpha \vec{x}_i)\|^2$ differs from the oracle's error $\|\vec{y}_i - \vec{y}_{\text{NN}}(\vec{x}_i)\|^2$ for each sample i . These individual differences result in $l_{\text{equiv}}(\hat{D}_\alpha) \neq l_f$. Nevertheless, if the oracle has sufficiently low generalization error, the values $\|\vec{y}_i - \hat{D}_\alpha^{-1} \vec{y}_{\text{NN}}(\hat{D}_\alpha \vec{x}_i)\|^2$ [and therefore $l_{\text{equiv}}(\hat{D}_\alpha)$] should lie within the oracle's error distribution. For this reason, we use a tolerance given by the maximum error among the ESM training set, i.e. $\varepsilon_{\text{equiv}} = \max_{i=1}^N \{\|\vec{y}_i - \vec{y}_{\text{NN}}(\vec{x}_i)\|^2\}$.

Once the local minima branches are pruned, all ESM branch matrices should represent a given symmetry transformation. These are maximally separated thanks to the previous \mathcal{L}_{rep} action, promoting the identification of all possible symmetries of the group. As mentioned above, the number of branches should initially exceed the order of the group K , which means that some symmetries \hat{D}_α can inevitably be found more than once. We thus perform a second final process in which redundant branches are removed. To do so, we calculate the quantities $d(\hat{W}_\alpha, \hat{W}_\beta)$ for all pairs of branches α, β , with the average element-wise distance (AED) between two matrices \hat{A} and \hat{B} defined as

$$d(\hat{A}, \hat{B}) = \frac{1}{n^2} \sum_{kl} |\hat{A}_{kl} - \hat{B}_{kl}|, \quad (9)$$

where n is the matrix dimension, and k, l index the matrix elements. Unlike the Frobenius norm, the AED is not a differentiable function in all the points of its domain, so we use it as a metric rather than a loss. It also quantifies the similarity between two matrices with the precision with which \hat{A} and \hat{B} can be asserted to be the same matrix (and thus indicates the number of significant digits in the converged matrices \hat{A} , \hat{B} , etc). Using the AEDs, we identify subsets of identical branches (i.e. subsets $\{\hat{W}_\alpha\}$ such that $d(\hat{W}_\alpha, \hat{W}_\beta) = 0$). Then, we retain a single representative from each subset, and discard the rest.

3.5. Verifying the results of the ESM

We can further confirm the symmetry group discovery by leveraging two fundamental group properties [42]: (1) a symmetry group is closed under multiplication, and (2) the inverse of every group element must also belong to the group. Since these properties were not explicitly enforced during ESM optimization, they can be used to validate whether the identified set of matrices forms a group representation. For this purpose, we employ a group metric similar to the one proposed in [32].

The group metric is defined as

$$d_{\text{group}} = \frac{1}{2} (d_{\text{closed}} + d_{\text{inverse}}), \quad (10)$$

where the 'closed metric' is given by

$$d_{\text{closed}} = \frac{1}{M^2} \sum_{\alpha=1}^M \sum_{\beta=1}^M \min_{\gamma} d(\hat{W}_\alpha \hat{W}_\beta, \hat{W}_\gamma), \quad (11)$$

and the 'inverse metric' is defined as

$$d_{\text{inverse}} = \frac{1}{M} \sum_{\alpha=1}^M \min_{\gamma} d(\hat{W}_\alpha^{-1}, \hat{W}_\gamma). \quad (12)$$

The group metric is a non-negative function of the branch matrices $\{\hat{W}_\alpha\}$ that equals zero if and only if the set of matrices forms a group.

4. Illustration of the ESM training method

In this section, we illustrate the proposed ESM training procedure using a simple example. We choose a dynamical system governed by

$$\frac{dx}{dt} = -x^3, \quad (13)$$

i.e. $y(x) = -x^3$. The solutions are analytically known ($x(t) = \pm 1/\sqrt{2t+C}$, being C an arbitrary constant), and decay asymptotically to $x = 0$ as $t \rightarrow \infty$. This system is symmetric under sign inversion: if $x(t)$ is a solution, so is $-x(t)$. Thus, the symmetry group has $K = 2$ elements: $D_1 = 1$ (identity) and $D_2 = -1$, that is $x' = D_\alpha x$ leaves equation (13) invariant. Note that, in this simple case, the transformations are scalars.

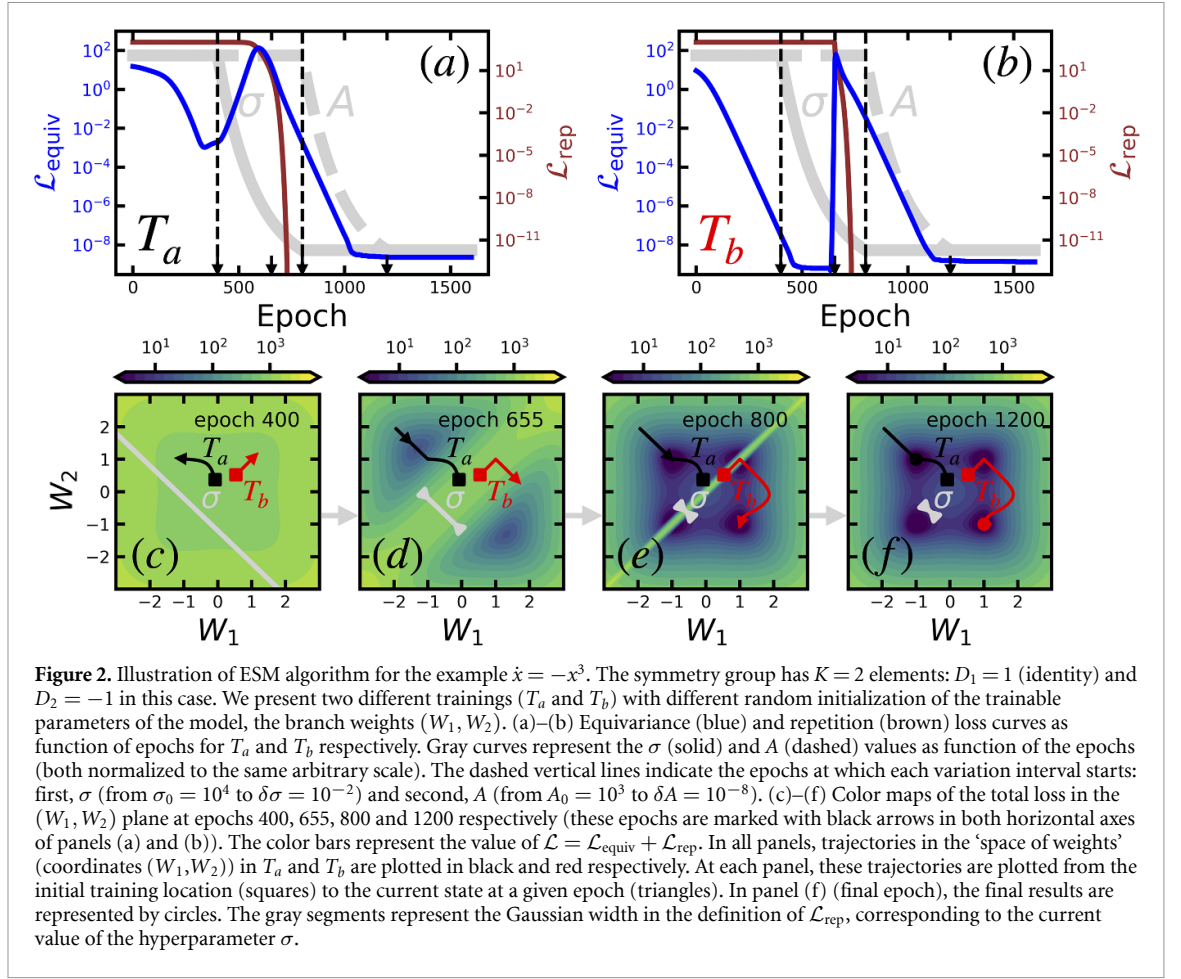
To generate the dataset, we simulate $N_T = 10$ trajectories $\{x^{(r)}(t_s)\}$ with time discretization of $\Delta t = 10^{-3}$ at $N_S = 100$ steps, using the Runge–Kutta method. The initial condition of each trajectory is randomly sampled from a uniform distribution within the interval $x(t=0) \in [-2.5, 2.5]$ (the initial condition determines the constant C for each trajectory). We illustrate the method in two cases: (1) when y_i is known analytically and (2) when only the trajectories are known and y_i is estimated from them. We randomly shuffle the 1000 training pairs and use 750 samples for the oracle y_{NN} (see appendix A for further details) and $N = 250$ for training the ESM.

The process begins with fixed initial values $\sigma = \sigma_0$ and $A = A_0$, and the gradual elimination of \mathcal{L}_{rep} is executed in two stages. In the first stage, σ is progressively reduced to $\sigma = \delta\sigma \approx 0$, which vanishes the influence of \mathcal{L}_{rep} for branch pairs \hat{W}_α and \hat{W}_β located in different global minima (we do not set $\delta\sigma = 0$ because that would introduce a discontinuity in \mathcal{L}_{rep}). In the second stage, A is decreased to $A = \delta A \approx 0$, eliminating any remaining contribution of \mathcal{L}_{rep} , even for redundant branches within the neighborhood of the same minimum, leaving only \mathcal{L}_{equiv} to drive the optimization (setting $\delta A \neq 0$ is convenient for reasons that become evident later in this section). As we will show below, these hyperparameter reduction processes allow knowing whether the number of branches has exceeded the order K or not. If $M > K$, the procedure ends with the branch-removal processes explained in subsection 3.4, culminating with an ESM having $M = K$ branches. In addition to this training, we validate the K -order symmetry group finding by calculating the group metric exposed in subsection 3.5.

Figure 2 shows the training procedure of a 2-branch ESM ($M = K$) with the analytical derivative function. The loss function \mathcal{L} depends on the branch weights (W_1, W_2), and \mathcal{L}_{equiv} has four global minima corresponding to the combinations of those in l_{equiv} : $(1, -1)$ and $(-1, 1)$, which capture all matrices within the equivariant symmetry group, and $(1, 1)$ and $(-1, -1)$, which leave one matrix undetected. We perform two trainings (T_a and T_b , panels (a) and (b) in figure 2) with different weight initialization in order to demonstrate the method's effectiveness under various conditions. In both, hyperparameters σ and A are decreased exponentially (see appendix A): the width runs from $\sigma_0 = 10^4$ to $\delta\sigma = 10^{-2}$ (between epochs 400 and 800), and the amplitude runs from $A_0 = 10^3$ to $\delta A = 10^{-8}$ (between epochs 800 and 1200).

Figure 2(a) shows the evolution of the equivariance and repetition losses during training for setup T_a , while figure 2(b) does the same for setup T_b . Besides, figures 2(c)–(f) visualize the total loss \mathcal{L} landscapes (color maps of the total loss in the (W_1, W_2) plane) at four representative epochs. The branch weight evolutions for both ESM trainings are traced from the initial configuration to the current one at each key epoch.

Initially, and up until epoch 400, when σ begins to diminish, the large σ value makes \mathcal{L}_{rep} nearly constant, leaving the optimization solely influenced by \mathcal{L}_{equiv} . As a result, both trainings converge to isolated \mathcal{L}_{equiv} minima (see figure 2(c)). Due to the logarithmic scale of the color bar, these minima cannot be distinguished in the figure. However, note that the total loss landscape is identical to that at epoch 1200 in figure 2(f)–(w) where \mathcal{L}_{rep} vanishes—except for an additive constant A . We observe that T_a converges to $(-1, 1)$, which is a desired situation where the two symmetry matrices are found, while T_b converges to $(1, 1)$, which is an undesired situation where one symmetry matrix is repeated and the other is not found. During the process of σ reduction from epochs 400 to 800, as shown in figures 2(a) and (b), the repetition loss begins to reshape the loss landscape, as it can be observed at epoch 655 in figure 2(d), driving the branches to reduce \mathcal{L}_{rep} while increasing \mathcal{L}_{equiv} in order to minimize the total loss. In T_a , the predicted branch weights undergo only slight perturbations as the \mathcal{L} minimum $(-1, 1)$ shifts, reflecting the weak repulsion between separated branch matrices due to high σ . In contrast, \mathcal{L}_{rep} transforms $(1, 1)$ into a maximum, causing the branch weights in T_b to move away from it. Since the branch matrices are repeated, they experience a strong repulsion, which pushes the second matrix toward the alternative l_{equiv} minimum, -1 .



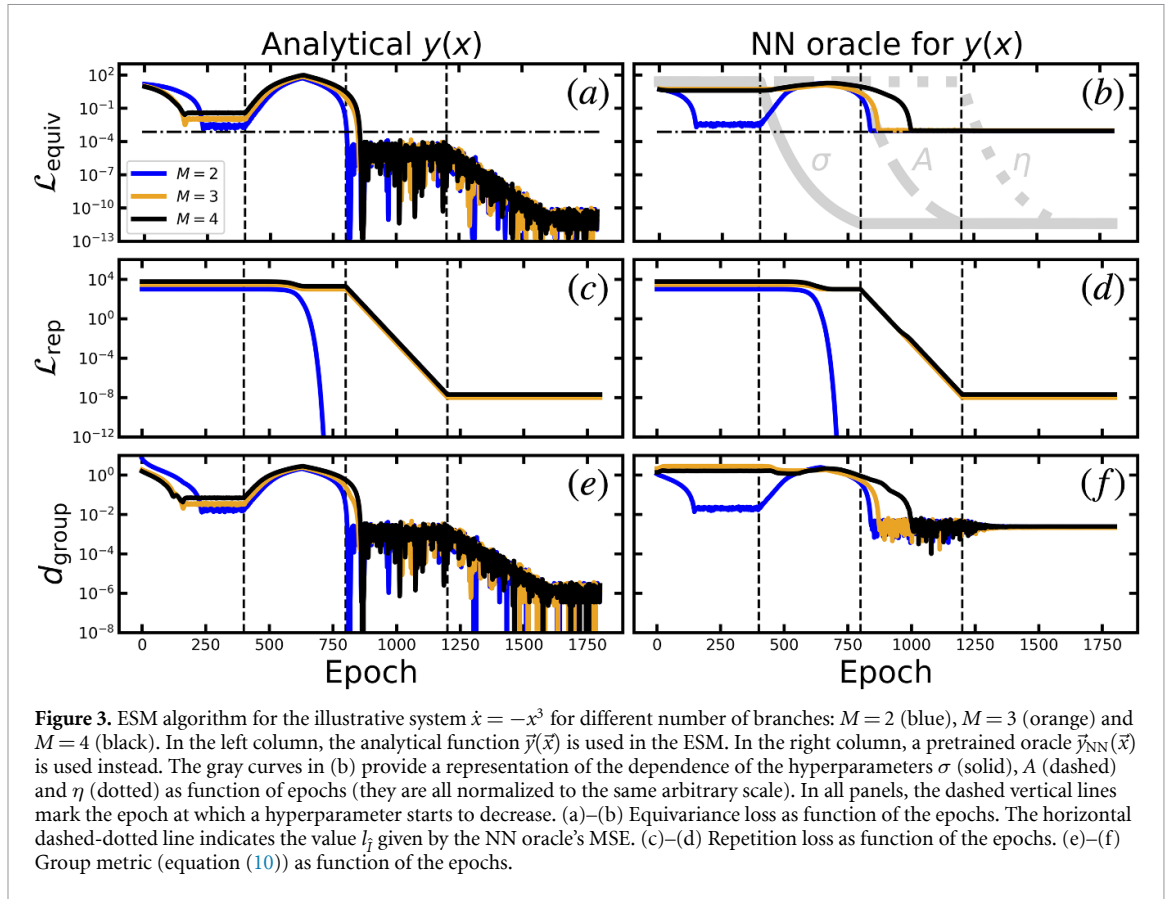
By the end of the σ reduction (epoch 800, see figure 2(e)), the branch weights in both trainings are fully separated into distinct symmetry minima. At this stage, non-zero values of \mathcal{L}_{rep} appear only close to the diagonal $W_1 = W_2$. Note that local minima of \mathcal{L} appear around the diagonal at $(1, 1)$ and $(-1, -1)$. However, the dynamic σ -process has prevented the training to end up in these minima. Finally, as A decreases from epochs 800 to 1200, \mathcal{L}_{rep} vanishes across the entire landscape, seen both in the evolution of the loss (figures 2(a) and (b)) and the landscape at epoch 1200 (panel (f), same figure). Since $M = K$, this variation has no effect on the optimization of the branch matrices because they are already located in minima of distinct symmetries.

Importantly, this experiment with $M = K$ demonstrates that, regardless of the weight initialization, the ESM training procedure always identifies the entire symmetry group, without detecting the same symmetry twice.

The variation of A becomes crucial when $M > K$. We demonstrate this by considering ESMs with different number of branches: $M = 2, 3$ and 4 (see figure 3). For each M , we perform two trainings, one using the analytical derivative function and another using a pretrained oracle [$\mathcal{Y}(x) = y_{\text{NN}}(x)$]. The variation stages for σ and A are identical to those in figure 2. In addition to the steps previously followed, the learning rate η is exponentially reduced from $\eta_0 = 10^{-3}$ to $\delta\eta = 10^{-6}$ between epochs 1200 and 1600 in order to mitigate gradient descent oscillations around the achieved minimum and, thus, improve convergence precision.

Figure 3(a) shows the equivariance loss curves for each number of branches using $y(x)$, while figure 3(b) renders the same quantities when using $y_{\text{NN}}(x)$ as oracle. Similarly, figures 3(c) and (d) show the corresponding repetition loss curves.

When $M = 2$, the behavior is consistent with figure 2. At the end of the σ variation process, $\mathcal{L}_{\text{equiv}}$ is minimized and \mathcal{L}_{rep} vanishes, indicating that the ESM has identified M distinct symmetries. However, when $M = 3$ or $M = 4$ ($M > K$), \mathcal{L}_{rep} does not vanish. In this case, since the number of branches has exceeded the number of unique global minima, some matrices inevitably fall into the same basin of attraction. These overlapping matrices interact via \mathcal{L}_{rep} , even at low σ , introducing perturbations that prevent the total minimization of $\mathcal{L}_{\text{equiv}}$ and, thus, an accurate prediction of the ground-truth symmetries. This difference in



behavior between $M = K$ and $M > K$ cases provides a criterion for identifying the order of the symmetry group K when it is unknown.

The importance of reducing parameter A becomes evident at this stage. For $M = 3$, the solution converges to $\{W_\alpha\} = \{1, -1, -1\}$, where one ground-truth symmetry is repeated, resulting in a final $\mathcal{L}_{\text{rep}} = \delta A$. For $M = 4$, the solution becomes $\{W_\alpha\} = \{1, -1, 1, -1\}$, with both ground-truth symmetries repeated, yielding $\mathcal{L}_{\text{rep}} = 2\delta A$ (see figures 3(c) and (d)). Although matrix repetition occurs, it is no longer an issue as all global minima, and thus all symmetry elements D_α , are identified.

Notably, for all M , the equivariance losses reach their expected bound (0 and l_f), confirming that the identified transformations are valid symmetries. A clear distinction can be observed in figure 3(a) and (b) between training with the analytical derivative function and the NN oracle. For ESMs trained with $\mathcal{Y}(x) = y(x)$, $\mathcal{L}_{\text{equiv}}$ achieves extremely low values (around 10^{-12} , which is much lower than the tolerance $\varepsilon_{\text{equiv}}$) during the η -decreasing stage, demonstrating that the equivariance loss can be arbitrarily minimized. In contrast, for ESMs trained with $\mathcal{Y}(x) = y_{NN}(x)$, $\mathcal{L}_{\text{equiv}}$ consistently converges to the l_f value given by the NN oracle's prediction error and cannot be further minimized, even as η is reduced to enhance convergence precision.

For each value of M , we track the group metric (equation (10)) as a function of epochs, as shown in figures 3(e) and (f). In all cases, d_{group} decreases alongside $\mathcal{L}_{\text{equiv}}$: the better the matrices satisfy the symmetry conditions, the better the set satisfies the group properties. This is particularly evident for $M = K = 2$, where $W_1 = 1$ and $W_2 = -1$ naturally fulfill the group criteria. When $M > K$, the group metric converges to a similarly low value, confirming that all symmetries are captured and redundant matrices maintain group closure and inverse properties.

Up to this point, we have demonstrated that the ESM reliably identifies all K symmetry group elements, regardless of the weight initialization or the number of branches M (provided that $M \geq K$). In simpler systems (e.g. equation (13)), no additional processing is needed. The optimization is not affected by local minima in l_{equiv} , and redundant matrices in cases where $M > K$ can be easily identified by inspection. However, for more complex systems with higher dimensionality, solutions may occur at local minima, and also, recognizing duplicated matrices becomes less straightforward. For that cases, it would be necessary to apply the branch-removal processes explained in subsection 3.4.

5. Application of the ESM to complex dynamical systems

In this section, we showcase several examples of the ESM method applied to dynamical systems with higher complexity than the simple example considered in section 4, including some exhibiting chaotic behavior. As illustrations, we use Thomas symmetric attractor, Lorenz system, and a system of coupled Duffing oscillators.

The data generation is similarly conducted for all systems. The states $\{\vec{x}^{(r)}(t_s)\}$ are obtained by generating $N_T = 1000$ trajectories using the Runge–Kutta method, each with $N_S = 100$ time steps ($\Delta t = 10^{-3}$). Initial conditions are randomly sampled from a uniform distribution within the region of \mathbb{R}^n that satisfies $-2.5 < x_k(t=0) < 2.5$. Derivatives $\{\vec{y}^{(r)}(t_s)\}$ are obtained both analytically and numerically as in the previous section. After random shuffling, we use 95 000 data pairs to train and validate the oracles \vec{y}_{NN} (see the A for details), and $N = 5000$ pairs to train the ESM. In order to assess the robustness to limited data, C presents a study of ESM accuracy as we randomly remove increasing fractions of the mentioned data set. We also introduce a group-based fine-tuning optimization that substantially improves ESM performance when the available data are severely reduced.

We used the same hyperparameter settings in all these systems as well (see the appendix A for a detailed summary of used hyperparameters). In each case, we apply three hyperparameter-tuning stages: the width σ is reduced from $\sigma_0 = 10^{-1}$ to $\delta\sigma = 10^{-4}$ (epochs 600–1200), the amplitude A from $A_0 = 10^6$ to $\delta A = 10^{-8}$ (epochs 1200–1800), and the learning rate η from $\eta_0 = 10^{-3}$ to $\delta\eta = 10^{-6}$ (epochs 1800–2400). Branch-removal processes are conducted at epoch 2600 for local minima and at epoch 2800 for redundant branches.

5.1. Thomas' symmetric attractor

Let us start with the Thomas' symmetric attractor. It is a three-dimensional system which was proposed by René Thomas [43] and is described by the following equations:

$$\begin{aligned}\frac{dx_1}{dt} &= \sin x_2 - bx_1 \\ \frac{dx_2}{dt} &= \sin x_3 - bx_2 \\ \frac{dx_3}{dt} &= \sin x_1 - bx_3.\end{aligned}\tag{14}$$

Depending on the parameter b , this dynamical system exhibits various regimes, including chaotic behavior [44] (we used $b = 0.208\,186$ to be in the chaotic regime). These equations are symmetric under cyclical permutations of the state variables (x_1, x_2, x_3) and under sign inversion (i.e. if $\vec{x}(t)$ is a solution, so is $-\vec{x}(t)$). Together, these symmetries form a group of order $K = 6$.

The results of the ESM trainings are shown in figure 4. We consider two ESMs: one with $M = 4$ branches (to analyze the ESM performance when selecting an insufficiently large M hyperparameter) and another with $M = 30$ branches (to significantly exceed K). Figures 4(a)–(f) show the evolution of the $\mathcal{L}_{\text{equiv}}$, \mathcal{L}_{rep} , and d_{group} for both values of M . Branch removal processes for the ESM with $M = 30$ are indicated by the red and purple shaded regions. As in the previous section, we compare trainings that use either the analytical function $\vec{y}(\vec{x})$ from equation (14) (figure 4(a),(c),(e)) or the NN oracle \vec{y}_{NN} (figures 4(b),(d),(f)).

Let us first focus on the ESM with $M = 4$ branches. Immediately before the variation of σ , we observe that the equivariance loss is significantly minimized while the repetition loss vanishes. This indicates that the ESM has identified four distinct symmetry transformations (so final-training branch removals are not necessary in this case). However, the group metric does not decrease in parallel with $\mathcal{L}_{\text{equiv}}$, suggesting that the set $\{\hat{W}_\alpha\}$ does not form a group, and thus some symmetry transformations of the dynamical system have not been found.

In general, d_{group} is an appropriate measure to reveal whether the number of branches M is sufficient to determine all symmetry transformations. Indeed, if M is lower than the true symmetry group order K , it is likely that the found matrices do not form a group. The possible exception is when the set of matrices form a subgroup, which would cause the group metric to decrease significantly even when $M < K$. To identify whether a subgroup has been found, the ESM should be considered with one extra branch. If the converged \mathcal{L}_{rep} still vanishes, the previous set was a subgroup and some symmetry matrices were missing.

Now, consider the $M = 30$ case. As expected, \mathcal{L}_{rep} does not vanish after decreasing the Gaussian width σ , indicating that K has been exceeded and some matrices are repeated. Unlike previous experiments, the minimization of $\mathcal{L}_{\text{equiv}}$ does not converge to its lower bound after the Gaussian amplitude variation because some matrices \hat{W}_α converge to local minima that do not correspond to exact symmetries. This can be seen in figure 4(g) for the $\vec{y}(\vec{x}) = \vec{y}_{NN}(\vec{x})$ case. The panel shows the single-branch equivariance losses for each α before the first removal, ordered in ascending l_{equiv} values. We show with a horizontal dashed-dotted line the

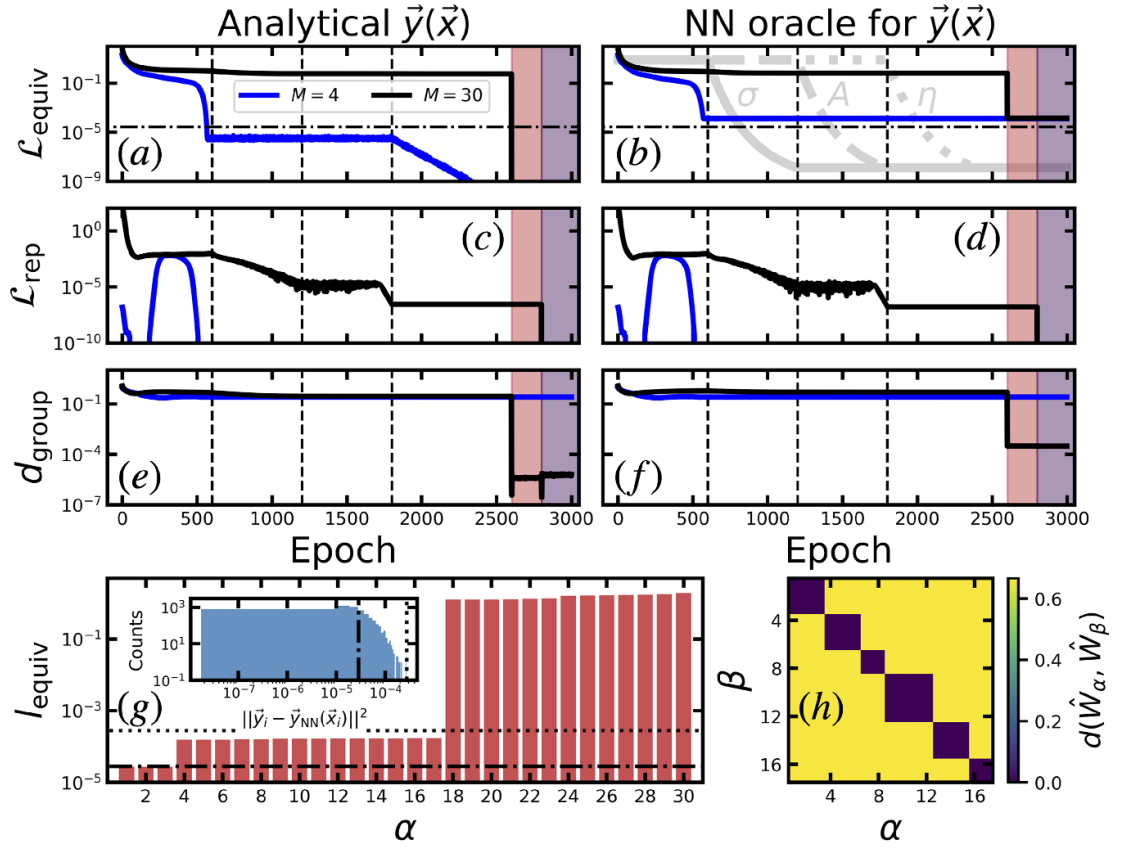


Figure 4. ESM algorithm for the Thomas' cyclically symmetric attractor. (a)–(f) Optimization of the matrices $\{\hat{W}_\alpha\}$, using both the analytical function $\vec{y}(\vec{x})$ and oracle $\vec{y}_{NN}(\vec{x})$, for $M = 4$ (blue) and $M = 30$ (black). Three quantities are shown as function of the epochs: equivariance loss (a)–(b), repetition loss (c)–(d) and group metric (e)–(f). The gray curves in (b) indicate the variation of σ , A and η , and the vertical dashed lines in all panels represent the starting of each hyperparameter decrease. The dashed-dotted lines in (a)–(b) indicate the bound l_f given by the oracle's MSE. In all panels, the red and purple areas indicate the regimes at which some branches have been removed by each of the two final-training processes. (g)–(h) Branch-removal processes for $M = 30$ using $\vec{y}_{NN}(\vec{x})$. (g) Removal of matrices in local minima. The plot shows each single-matrix equivariance loss for all the matrices before the removal. The dotted-dashed line represents l_f , and the dotted line represents the tolerance ε_{equiv} given by the maximum oracle squared error. The inset displays the histogram of squared errors for all ESM training samples, with the dashed-dotted line representing l_f the dotted line representing ε_{equiv} . (h) Removal of redundant matrices. The plot shows the AED between all possible matrix pairs before the removal.

approximate bound of the function, $l_f \simeq 3 \cdot 10^{-5}$. With the dotted line, we show the tolerance $\varepsilon_{equiv} \simeq 3 \cdot 10^{-4}$ that we use to remove local minima branches, whose value is given by the maximum oracle's error. As an inset, we represent the histogram of the oracle errors $\|\vec{y}_i - \vec{y}_{NN}(\vec{x}_i)\|^2$. Similarly, the vertical dashed-dotted and dotted lines represent l_f and ε_{equiv} respectively. We observe that 13 branch matrices converge to local minima with $l_{equiv}(\hat{W}_\alpha) \sim 1$ (several orders of magnitude higher than l_f). The remaining 17 branches converge to global minima according to our criterion. It can be seen that three out of these 17 global minima matrices have l_{equiv} values extremely close to l_f because they converge to the identity transformation. The others, associated to non-trivial symmetry transformations \hat{D}_α , have slightly higher l_{equiv} values, but still within the oracle's error distribution.

The 13 local minima matrices are eliminated during the first removal process. In figure 4(a) and (b), we observe that \mathcal{L}_{equiv} drops abruptly after this removal. In the case of $\vec{Y}(\vec{x}) = \vec{y}(\vec{x})$, \mathcal{L}_{equiv} completely vanishes. On the other hand, for $\vec{Y}(\vec{x}) = \vec{y}_{NN}(\vec{x})$, the equivariance loss decreases to a value similar to that achieved with the 4-branch ESM, which is consistent with the l_{equiv} values in figure 4(g).

Finally, figure 4(h) displays the AEDs $d(\hat{W}_\alpha, \hat{W}_\beta)$ before the second removal in the \vec{y}_{NN} case. In this figure, branch indices α, β are arranged in such a way that the first corresponds to the lowest l_{equiv} value, and subsequent indices are sorted by proximity (the lowest AED to the previous branch). We can observe that the 17 remaining branch matrices contained 6 unique symmetry transformations. Removing redundant matrices makes \mathcal{L}_{rep} drop to zero (see figures 4(c) and (d)), confirming that no matrices are repeated. Importantly, d_{group} remains unchanged (see figures 4(e) and (f)), verifying that the removed matrices were redundant and the remaining set still forms a valid symmetry group. At the end of the process, the ESM accurately predicts the finite equivariant symmetry group with $M = K = 6$ branch matrices, whose values are shown in appendix B.

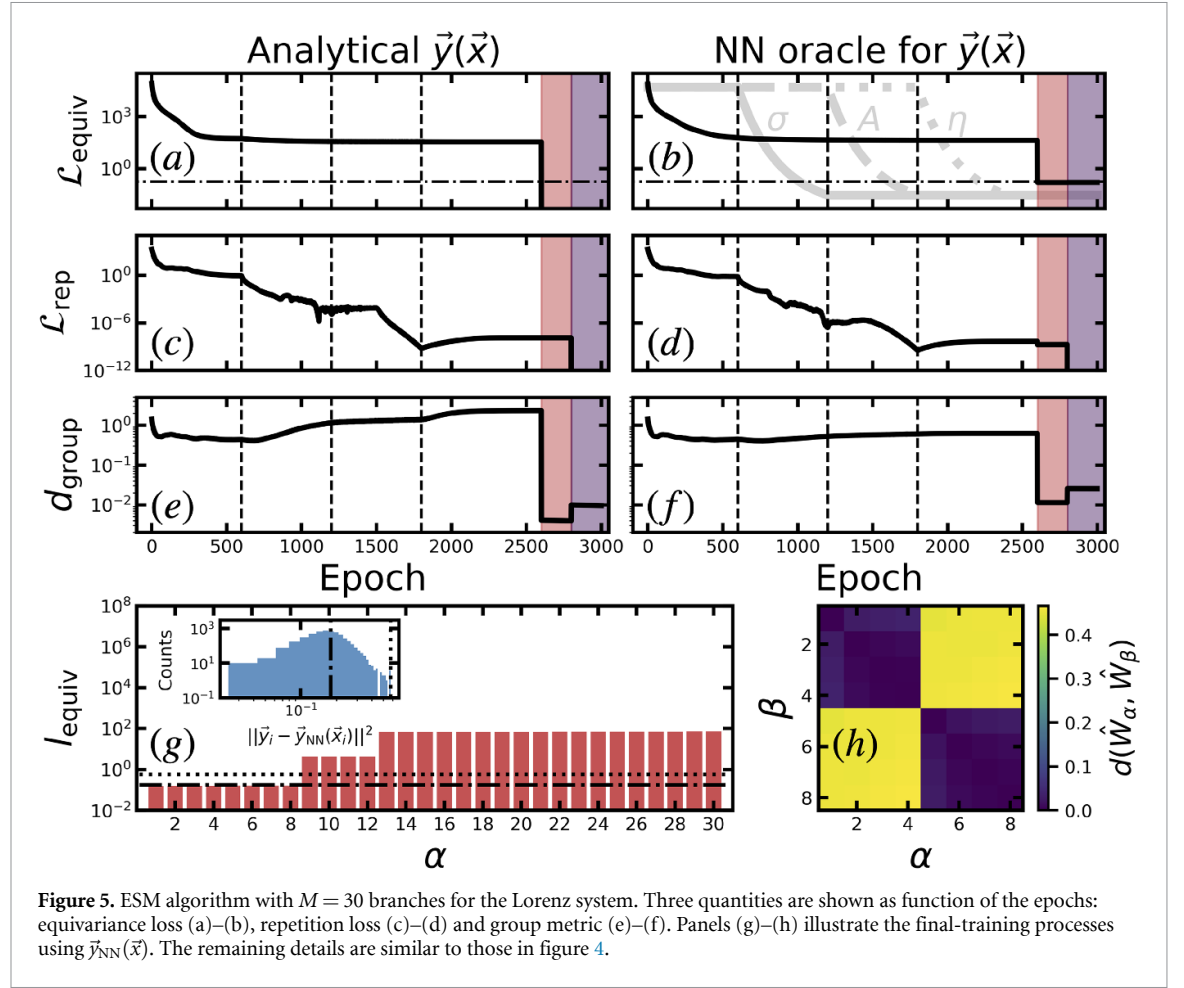


Figure 5. ESM algorithm with $M = 30$ branches for the Lorenz system. Three quantities are shown as function of the epochs: equivariance loss (a)–(b), repetition loss (c)–(d) and group metric (e)–(f). Panels (g)–(h) illustrate the final-training processes using $\vec{y}_{\text{NN}}(\vec{x})$. The remaining details are similar to those in figure 4.

5.2. Lorenz system

We consider now the Lorenz system, which is a three-dimensional system described by the following equations:

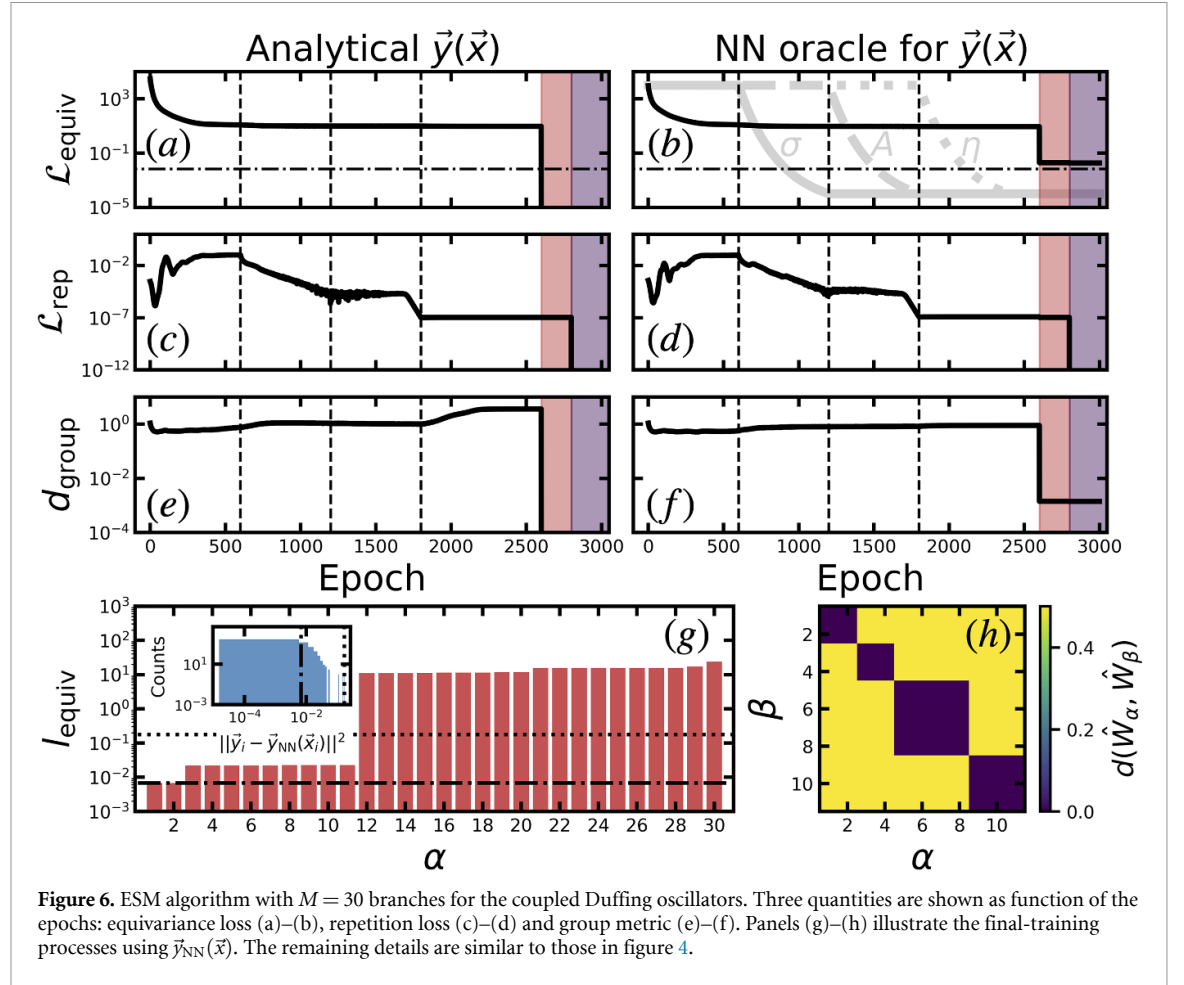
$$\begin{aligned} \frac{dx_1}{dt} &= a(x_2 - x_1) \\ \frac{dx_2}{dt} &= x_1(b - x_3) - x_2 \\ \frac{dx_3}{dt} &= x_1x_2 - cx_3 \end{aligned} \quad (15)$$

being a , b and c are fixed parameters (we used $a = 10$, $b = 28$ and $c = 8/3$). This is a paradigmatic example of chaotic dynamics and was originally developed as a simplified model of atmospheric convection [45].

The equations are symmetric under the sign inversion of the first two state variables, i.e. if $\vec{x}(t) = [x_1(t), x_2(t), x_3(t)]^T$ is a solution, so is $\vec{x}'(t) = [-x_1(t), -x_2(t), x_3(t)]^T$. Thus, the symmetry group has order $K = 2$.

The results are presented in figure 5 with the same structure as in figure 4. This time, we explored only the $M > K$ case: we trained an ESM with $M = 30$ branches, both using the analytical function $\vec{y}(\vec{x})$ and the NN oracle \vec{y}_{NN} . Figures 5(a)–(f) show the evolution of $\mathcal{L}_{\text{equiv}}$, \mathcal{L}_{rep} and d_{group} , and figures 5(g)–(h) specifies the branch-removal processes for the case $\vec{y}(\vec{x}) = \vec{y}_{\text{NN}}(\vec{x})$. We can observe that the oracle's error distribution sets a bound of the function $\mathcal{L}_{\text{equiv}}$ given by $l_i \simeq 0.2$, with a tolerance of $\varepsilon_{\text{equiv}} \simeq 0.6$.

We observe that the ESM training presents a similar behavior to the previous case: after decreasing σ , \mathcal{L}_{rep} does not vanish because matrix repetition occurs inevitably, as $M > K$. Besides, $\mathcal{L}_{\text{equiv}}$ does not vanish either because 22 matrices \hat{W}_α converge to local minima. We observe in figure 5(g) (NN oracle case) that the values of $\mathcal{L}_{\text{equiv}}(\hat{W}_\alpha)$ for those \hat{W}_α in local minima are of the order of 10^1 and 10^2 , much higher than l_i . These matrices are pruned in the first branch-removal process, producing $\mathcal{L}_{\text{equiv}}$ to drop to its lower bound ($\mathcal{L}_{\text{equiv}} \simeq 0$ and $\mathcal{L}_{\text{equiv}} \simeq l_i$). We see in figure 5(h) that the 8 global minima branches contain only 2 unique



symmetries. The second branch-removal process prunes 6 matrices, leaving the ESM with $M = K = 2$ (see appendix B for obtained values).

5.3. Coupled Duffing oscillators

The last example is a four-dimensional dynamical system that consists of two coupled Duffing oscillators, whose equations of motion are

$$\begin{aligned}
 \frac{dx_1}{dt} &= x_2 \\
 \frac{dx_2}{dt} &= -\delta x_2 - \alpha x_1 - \beta x_1^3 + \gamma(x_1 - x_3) \\
 \frac{dx_3}{dt} &= x_4 \\
 \frac{dx_4}{dt} &= -\delta x_4 - \alpha x_3 - \beta x_3^3 + \gamma(x_3 - x_1)
 \end{aligned} \tag{16}$$

where x_1 and x_2 denote the position and velocity of the first oscillator respectively, whereas x_3 and x_4 those of the second. The fixed parameters are: δ (damping), α , β (linear and non-linear restoring forces), and γ (coupling constant). For the numerical experiment, we used $\delta = 0.3$, $\alpha = -1$, $\beta = 1$ and $\gamma = 5$. The system exhibits symmetry under sign inversion [if $\vec{x}(t)$ is a solution, so is $-\vec{x}(t)$] and permutation of oscillators (if $\vec{x}(t) = [x_1(t), x_2(t), x_3(t), x_4(t)]^T$ is a solution, so is $\vec{x}'(t) = [x_3(t), x_4(t), x_1(t), x_2(t)]^T$). Together, these form a symmetry group of order $K = 4$.

The results are shown in figure 6 with the same structure as in previous cases. We trained an ESM with $M = 30$ branches, also in the $\vec{\mathcal{Y}}(\vec{x}) = \vec{y}(\vec{x})$ and $\vec{\mathcal{Y}}(\vec{x}) = \vec{y}_{\text{NN}}(\vec{x})$ cases. As we see, the ESM behavior is consistent with those presented in previous subsections. This time, in the NN oracle case [see Fig 6(g)], the l_{equiv} bound is given by $l_i \simeq 7 \cdot 10^{-3}$, with a tolerance of $\varepsilon_{\text{equiv}} \simeq 0.2$. We see that 19 branches converge to local minima (with values $l_{\text{equiv}}(\hat{W}_\alpha) \sim 10^1$), and 11 branches converge to global minima with l_{equiv} values that lie within the oracle's error distribution. To conclude, we observe in figure 6(h) that, at the end of the protocol, 4 distinct symmetries were detected, coinciding with the order $K = 4$ (see appendix B for the obtained values).

6. Conclusions

In this work, we introduced the ESM, a ML framework designed to discover the finite equivariant group of linear symmetries in arbitrary nonlinear dynamical systems. The method focuses on identifying equivariant transformations through a training process regularized by the repetition loss, which ensures the complete identification of all symmetries within the group. Unlike iterative approaches, the ESM unveils all these elements in a single training cycle, with the group metric validating the results.

Notably, the ESM effectively works in both theory-informed and purely data-driven settings. While it can leverage analytical equations when available, it requires only trajectory data to operate, with no prior knowledge of the system's governing equations. This is achieved by pre-training an oracle approximator to estimate the system's dynamics. In our experiment, we employed a simple feed-forward NN as the oracle, but the method is adaptable and can integrate alternative ML methods.

We expect the ESM can be extended in the future to identify Lie equivariant symmetry groups (continuous symmetries) and other families of symmetries in dynamical systems (such as non-linear transformations involving also the independent variable t), as well as to partial differential equations (PDEs).

Data availability statement

The data supporting the findings of this study are synthetically generated. The method for generating these data is explained in detail in the manuscript, allowing any reader to reproduce them on their own. The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgment

We acknowledge projects PID2023-148359NB-C21 and CEX2023-001286-S (financed by MICIU/AEI/10.13039/501100011033) and the Government of Aragon through Project Q-MAD.

Code availability

The code for this paper can be found in the following Github repository: https://github.com/pablocalvo7/Equivariance_Seeker_Model.git.

Appendix A. Training details

This appendix details the training procedures for the experiments in this paper, including ESM training and NN oracle pre-training. All calculations were performed using TensorFlow [46] and Keras [47] libraries.

For all ESM experiments, hyperparameter variation followed the formula:

$$h(\text{epoch}) = \begin{cases} h_0 & \text{for epoch} < \text{epoch}_i \\ h_0 e^{-\omega(\text{epoch} - \text{epoch}_i)} & \text{for epoch}_i \leq \text{epoch} \leq \text{epoch}_f \\ \delta h & \text{for epoch} > \text{epoch}_f \end{cases} \quad (\text{A.1})$$

with

$$\omega = \frac{1}{\text{epoch}_f - \text{epoch}_i} \log \frac{h_0}{\delta h}, \quad (\text{A.2})$$

where h represents the tuned hyperparameter (either σ , A or η), h_0 and δh denote the initial and final hyperparameter values, and $[\text{epoch}_i, \text{epoch}_f]$ specifies the variation interval. These values, as well as further hyperparameters and training information, are summarized in table A1 for each experiment in this paper. Note that the experiments illustrated in figures 4–6 share identical training setting.

We trained four different feedforward NN oracles \vec{y}_{NN} , one for each dynamical system considered in this paper. As mentioned in the main text, these NNs approximate the derivative field, so $\vec{y}_{\text{NN}}(\vec{x}) \simeq \vec{y}(\vec{x})$. For that, we train them using a subset of randomly chosen pairs from the data set $\{\vec{x}^{(r)}(t_s), \vec{y}^{(r)}(t_s)\}$, with $\vec{y}^{(r)}(t_s)$ computed numerically as explained in section 2. All oracles were trained using the MSE loss and the Adam optimizer.

We now summarize each oracle training hyperparameters and settings:

- 1D illustrative system (equation (13)): 500 samples for training and 250 for validation. The architecture is composed by the input layer ($n = 1$ neuron), 2 hidden layers (100 neurons) with ReLU activation function

Table A1. Summary of ESM training hyperparameters from each experiment presented in this article. This involves four different systems: the illustrative one-dimensional system (equation (13)), Thomas system (equation (14)), Lorenz system (equation (15)) and the coupled Duffing oscillators (equation (16)).

| Hyperparameter | Figure 2 | Figure 3 | Figure 4, 5 and 6 |
|-----------------------------|--------------------------|----------------------|----------------------|
| $(\sigma_0, \delta\sigma)$ | $(10^4, 10^{-2})$ | $(10^4, 10^{-2})$ | $(10^{-1}, 10^{-4})$ |
| $(A_0, \delta A)$ | $(10^3, 10^{-8})$ | $(10^3, 10^{-8})$ | $(10^6, 10^{-8})$ |
| $(\eta_0, \delta\eta)$ | 10^{-4} (no variation) | $(10^{-3}, 10^{-6})$ | $(10^{-3}, 10^{-6})$ |
| Variation interval σ | 400–800 | 400–800 | 600–1200 |
| Variation interval A | 800–1200 | 800–1200 | 1200–1800 |
| Variation interval η | — | 1200–1600 | 1800–2400 |
| Epoch 1st removal | — | — | 2600 |
| Epoch 2nd removal | — | — | 2800 |
| Total epochs | 1200 | 1800 | 3000 |
| N (ESM data) | 250 | 250 | 5000 |
| Optimizer | SGD | RMSprop | RMSprop |
| Mini-batch size | 64 | 64 | 1024 |

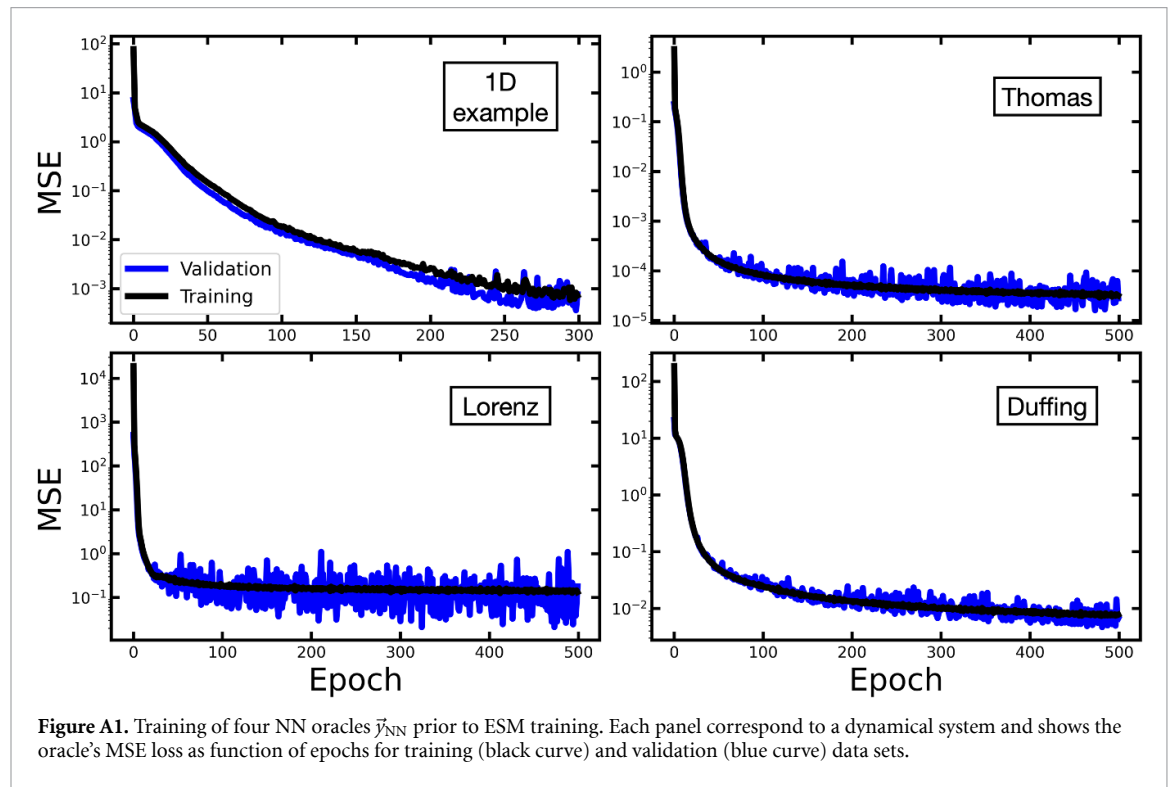


Figure A1. Training of four NN oracles \vec{y}_{NN} prior to ESM training. Each panel correspond to a dynamical system and shows the oracle's MSE loss as function of epochs for training (black curve) and validation (blue curve) data sets.

and the output layer ($n = 1$ neuron) with linear activation function. The NN was trained during 300 epochs using a learning rate of 10^{-4} and a mini-batch size of 32.

- Thomas, Lorenz and Duffing systems (equations (14)–(16)): 90 000 samples were used for training and 5000 for validation. The architecture is composed by the input layer comprising either 3 (Thomas and Lorenz) or 4 (Duffing) neurons, 3 hidden layers (500 neurons each) with ReLU activation function and the output layer composed of 3 or 4 neurons, with linear activation function. The NN was trained during 500 epochs using a learning rate of 10^{-5} and a mini-batch size of 256.

Oracle training curves are shown in figure A1. Once trained, the oracle weights are fixed and integrated into the \vec{y} block of the ESMs.

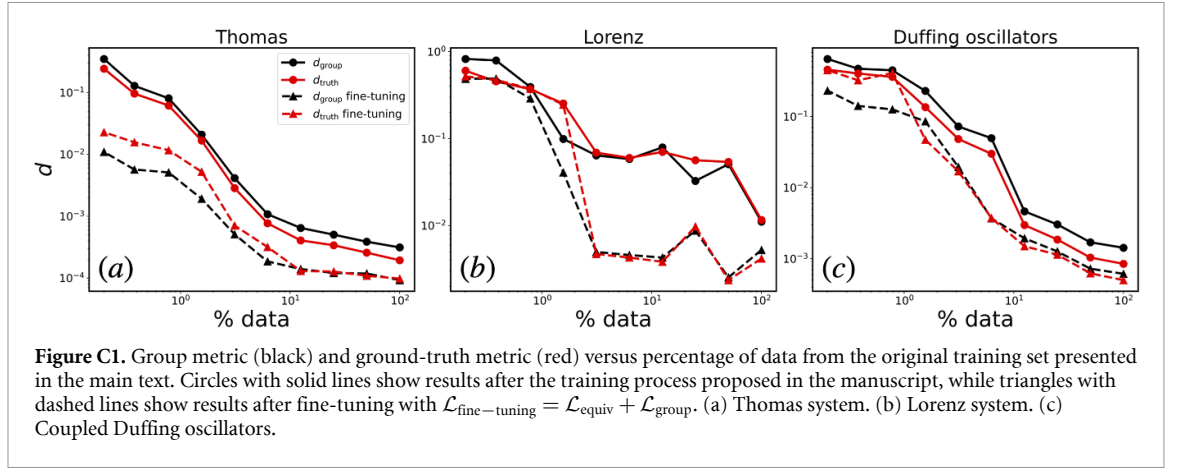
Appendix B. Branch matrix results

In this appendix, we present the matrix values of the symmetry groups discovered using our method in section 5, that is, for the Thomas attractor, Lorenz system and the Duffing oscillators. Specifically, we provide in table B1 the numerical results obtained from the ESMs with $M = 30$ branches in the case of $\vec{Y}(x) = \vec{y}_{\text{NN}}(\vec{x})$, truncated to five decimal places.

In general, the appropriate decimal precision for rounding the obtained matrix values should be chosen such that the group metric d_{group} is minimized. Notably, in all cases presented in table B1, rounding the matrix values to the nearest integer yields the exact symmetry matrices $\{\hat{D}_\alpha\}$, ensuring that d_{group} becomes exactly zero.

Table B1. Obtained matrix values $\{\hat{W}_\alpha\}$ using ESMs with $M = 30$ branches and NN oracles for the systems presented in section 5: Thomas system (equation (14)), Lorenz system (equation (15)) and the coupled Duffing oscillators (equation (16)).

| Thomas | Lorenz | Duffing |
|---|--|--|
| $\begin{pmatrix} 1.00034 & -0.00003 & -0.00002 \\ 0.00006 & 1.00001 & -0.00008 \\ -0.00014 & -0.00003 & 0.99983 \end{pmatrix}$ | $\begin{pmatrix} 0.95049 & 0.06961 & -0.00038 \\ 0.02487 & 0.97252 & -0.00023 \\ 0.00095 & -0.00043 & 0.99007 \end{pmatrix}$ | $\begin{pmatrix} -0.00036 & 0.00051 & -1.00051 & -0.00172 \\ 0.00068 & 0.00228 & 0.00024 & -0.99846 \\ -0.99871 & -0.00416 & 0.00109 & 0.00327 \\ -0.00013 & -0.99827 & 0.00049 & 0.00125 \end{pmatrix}$ |
| $\begin{pmatrix} -0.00018 & -1.00007 & -0.00016 \\ -0.00053 & 0.00014 & -1.00011 \\ -1.00073 & 0.00007 & 0.00017 \end{pmatrix}$ | $\begin{pmatrix} -0.96805 & -0.05061 & 0.00096 \\ -0.01825 & -0.98413 & 0.00012 \\ 0.00070 & -0.00052 & 0.99607 \end{pmatrix}$ | $\begin{pmatrix} 0.99990 & 0.00061 & -0.00032 & -0.00216 \\ 0.00040 & 1.00076 & 0.00009 & 0.00005 \\ 0.00014 & -0.00221 & 1.00057 & -0.00100 \\ 0.00015 & 0.00073 & 0.00019 & 0.99999 \end{pmatrix}$ |
| $\begin{pmatrix} -0.00019 & 0.00020 & -1.00025 \\ -1.00009 & -0.00008 & -0.00006 \\ -0.00033 & -1.00014 & -0.00005 \end{pmatrix}$ | | $\begin{pmatrix} -0.00013 & -0.00049 & 1.00048 & -0.00110 \\ 0.00037 & -0.00034 & 0.00004 & 1.00050 \\ 1.00062 & -0.00053 & -0.00142 & -0.00137 \\ 0.00020 & 1.00082 & 0.00009 & -0.00081 \end{pmatrix}$ |
| $\begin{pmatrix} -1.00048 & 0.00011 & 0.00037 \\ 0.00004 & -1.00032 & -0.00023 \\ -0.00037 & 0.00016 & -1.00021 \end{pmatrix}$ | | $\begin{pmatrix} -0.99945 & 0.00026 & 0.00017 & -0.00120 \\ 0.00045 & -0.99925 & 0.00057 & 0.00217 \\ 0.00021 & -0.00318 & -0.99978 & 0.00154 \\ -0.00022 & 0.00124 & 0.00039 & -0.99861 \end{pmatrix}$ |
| $\begin{pmatrix} 0.00009 & -0.00017 & 1.00008 \\ 1.00069 & -0.00003 & -0.00001 \\ -0.00001 & 1.00044 & 0.00005 \end{pmatrix}$ | | |
| $\begin{pmatrix} -0.00011 & 1.00037 & 0.00056 \\ 0.00004 & -0.00020 & 1.00058 \\ 1.00014 & -0.00001 & -0.00013 \end{pmatrix}$ | | |



Appendix C. Impact of data set size on ESM accuracy and group-based fine-tuning

In this appendix, we evaluate ESM robustness against reduced training data for the three systems of section 5 and introduce a group-loss fine-tuning step to boost matrix precision. All experiments use the purely data-driven setup, so ESM accuracy is in principle bounded by the NN oracle’s error.

For each dynamical system, we create nested data subsets by successively and randomly removing half of the data starting from the original 100 000 pairs $\{\vec{x}_i, \vec{y}_i\}$: 100%, 50%, 25%... down to 195 pairs (0.195%). At each dataset size, we split the data into oracle-training (90%), oracle-validation (5%) and ESM-training (5%), as in the main experiments. We then compute (i) the group metric d_{group} , which estimates accuracy with no ground-truth knowledge, and (ii) the ‘ground-truth metric’, defined as

$$d_{\text{truth}} = \frac{1}{M} \sum_{\alpha=1}^M \min_{\gamma} d(\hat{W}_{\alpha}, \hat{D}_{\gamma}), \quad (\text{C.1})$$

which is the average AED between each found branch matrix and its nearest true symmetry, giving an exact measure of accuracy (decimal precision).

Figure C1 shows the results for each system. d_{group} is plotted with black circles and solid lines as function of the percentage of retained data, while d_{truth} is shown in red. In all cases, both metrics remain within the same order of magnitude down to $\sim 10\%$, demonstrating that the ESM is robust against data reduction. At lower percentages ($\leq 1\%$), the metrics rise, indicating a significant loss of precision in symmetry detection.

To improve the ESM precision, specially in the low-data regime, we fine-tune the branch matrices by retraining with

$$\mathcal{L}_{\text{fine-tuning}} = \mathcal{L}_{\text{equiv}} + \mathcal{L}_{\text{group}}, \quad (\text{C.2})$$

where the “group loss” term is defined as

$$\mathcal{L}_{\text{group}} = \frac{1}{M^2} \sum_{\alpha=1}^M \sum_{\beta=1}^M \min_{\gamma} \|\hat{W}_{\alpha} \hat{W}_{\beta} - \hat{W}_{\gamma}\|_{\text{F}}^2 + \frac{1}{M} \sum_{\alpha=1}^M \min_{\gamma} \|\hat{W}_{\alpha}^{-1} - \hat{W}_{\gamma}\|_{\text{F}}^2. \quad (\text{C.3})$$

This loss term enforces the branch matrices to satisfy the two fundamental group properties: like d_{group} , it consists of a ‘closed’ term plus an ‘inverse’ term. We apply this fine tuning only after the main training—when branches are already separated and \mathcal{L}_{rep} is no longer needed. The incorporation of $\mathcal{L}_{\text{group}}$ during the initial training stage was avoided because it introduces additional local minima that significantly hinder ESM optimization.

Figure C1 also plots d_{group} and d_{truth} after fine-tuning (triangles and dashed lines). In all systems, fine-tuning yields substantial accuracy gains—e.g. in the Thomas system $d_{\text{truth}} \sim 10^{-2}$ even with just 0.195% of the data. In the Lorenz and Duffing cases, the metrics improve by an order of magnitude for data fractions between 1% and 10% ($d_{\text{truth}} \sim 10^{-3}$ is achieved at 3.12% in the Lorenz system and at 6.24% in the Duffing oscillators).

Notably, even in low-data regimes, fine-tuning the branch matrices using the combined equivariance and group loss allows the ESM to recover the true symmetries with high accuracy. The non-zero distance between the predicted symmetry matrices and the exact ones is attributable to the finite error of the NN oracle, which

increases as the dataset size decreases. The exact symmetry matrices can be obtained either by appropriately truncating the predicted elements or through a second fine-tuning stage using only the group loss ($\mathcal{L}_{\text{group}}$). In this latter approach, the predicted matrices converge arbitrarily close to the exact symmetries, limited numerically only by the learning rate selected for the optimization process.

ORCID iDs

Pablo Calvo-Barlés  <https://orcid.org/0000-0002-8371-9840>

Sergio G Rodrigo  <https://orcid.org/0000-0001-6575-168X>

Luis Martín-Moreno  <https://orcid.org/0000-0001-9273-8165>

References

- [1] Gross D J 1996 The role of symmetry in fundamental physics *Proc. Natl Acad. Sci.* **93** 14256–9
- [2] Noether E 1918 Invariante variationsprobleme Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, *Mathemat. Phys. Kl.* **1918** 235–57 (available at: <https://eudml.org/doc/59024>)
- [3] Olver P J 1993 *Applications of Lie Groups to Differential Equations* vol 107 (Springer)
- [4] Hitchin N J, Segal G B and Ward R S 2013 *Integrable Systems: Twistors, Loop Groups and Riemann Surfaces* vol 4 (Oxford University Press)
- [5] Lou S Y and Jia M 2024 From one to infinity: symmetries of integrable systems *J. High Energy Phys.* **2024** 1–10
- [6] Wang H et al 2023 Scientific discovery in the age of artificial intelligence *Nature* **620** 47–60
- [7] Yu R and Wang R 2024 Learning dynamical systems from data: an introduction to physics-guided deep learning *Proc. Natl Acad. Sci.* **121** e2311808121
- [8] Brunton S L, Proctor J L and Kutz J N 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems *Proc. Natl Acad. Sci.* **113** 3932–7
- [9] Rudy S H, Brunton S L, Proctor J L and Kutz J N 2017 Data-driven discovery of partial differential equations *Sci. Adv.* **3** e1602614
- [10] Liu Z and Tegmark M 2021 Machine learning conservation laws from trajectories *Phys. Rev. Lett.* **126** 180604
- [11] Liu Z, Madhavan V and Tegmark M 2022 Machine learning conservation laws from differential equations *Phys. Rev. E* **106** 045307
- [12] Liu Z, Sturm P O, Bharadwaj S, Silva S J and Tegmark M 2024 Interpretable conservation laws as sparse invariants *Phys. Rev. E* **109** L023301
- [13] Ha S and Jeong H 2021 Discovering invariants via machine learning *Phys. Rev. Res.* **3** L042035
- [14] Wetzel S J, Melko R G, Scott J, Panju M and Ganesh V 2020 Discovering symmetry invariants and conserved quantities by interpreting siamese neural networks *Phys. Rev. Res.* **2** 033499
- [15] Lu P Y, Dangovski R and Soljačić M 2023 Discovering conservation laws using optimal transport and manifold learning *Nat. Commun.* **14** 4744
- [16] Iten R, Metger T, Wilming H, del Rio L and Renner R 2020 Discovering physical concepts with neural networks *Phys. Rev. Lett.* **124** 010508
- [17] Greydanus S, Dzamba M and Yosinski J 2019 Hamiltonian neural networks *Advances in Neural Information Processing Systems* p 32
- [18] Cranmer M, Greydanus S, Hoyer S, Battaglia P, Spergel D and Ho S 2020 Lagrangian neural networks (arXiv: 2003.04630)
- [19] Liu Z and Tegmark M 2022 Machine learning hidden symmetries *Phys. Rev. Lett.* **128** 180201
- [20] Bondesan R and Lamacraft A 2019 Learning symmetries of classical integrable systems (arXiv: 1906.04645)
- [21] Desai K, Nachman B and Thaler J 2022 Symmetry discovery with deep learning *Phys. Rev. D* **105** 096031
- [22] Yang J, Walters R, Dehmamy N, Yu R, Sabato S and 2023 Generative adversarial symmetry discovery *Proc. 40th Int. Conf. on Machine Learning, Volume 202 of Proc. Machine Learning Research* ed A Krause, E Brunskill, K Cho, B Engelhardt and J Scarlett (PMLR) pp 39488–508
- [23] Barenboim G, Hirn J and Sanz V 2021 Symmetry meets AI *SciPost Phys.* **11** 014
- [24] Krippendorff S and Syaeri M 2020 Detecting symmetries with neural networks *Mach. Learn.: Sci. Technol.* **2** 015010
- [25] Decelle A, Martin-Mayor V and Seoane B 2019 Learning a local symmetry with neural networks *Phys. Rev. E* **100** 050102
- [26] Gabel A, Klein V, Valperga R, Lamb J S W, Webster K, Quax R and Gavves E 2023 Learning lie group symmetry transformations with neural networks *Proc. 2nd Annual Workshop on Topology, Algebra and Geometry in Machine Learning (TAG-ML), Volume 221 of Proc. of Machine Learning Research* (PMLR) pp 50–59
- [27] Gabel A, Quax R and Gavves E 2024 Data-driven lie point symmetry detection for continuous dynamical systems *Mach. Learn.: Sci. Technol.* **5** 015037
- [28] Ko G, Kim H, and Lee J 2024 Learning infinitesimal generators of continuous symmetries from data (arXiv: 2410.21853)
- [29] Forestano R T, Matchev K T, Matcheva K, Roman A, Unlu E B and Verner S 2023 Deep learning symmetries and their lie groups, algebras and subalgebras from first principles *Mach. Learn.: Sci. Technol.* **4** 025027
- [30] Dehmamy N, Walters R, Liu Y, Wang D and Yu R 2021 Automatic symmetry discovery with lie algebra convolutional network *Advances in Neural Information Processing Systems* vol 34 pp 2503–15
- [31] Hou W, Li M and You Y-Z 2024 Machine learning symmetry discovery for classical mechanics (arXiv: 2412.14632)
- [32] Calvo-Barlés P, Rodrigo S G, Sánchez-Burillo E and Martín-Moreno L 2024 Finding discrete symmetry groups via machine learning *Phys. Rev. E* **110** 045304
- [33] Bronstein M M, Bruna J, LeCun Y, Szlam A and Vandergheynst P 2017 Geometric deep learning: Going beyond euclidean data *IEEE Signal Process. Mag.* **34** 18–42
- [34] Cohen T and Welling M 2016 Group equivariant convolutional networks *Proc. 33rd Int. Conf. on Machine Learning, Volume 48 of Proc. Machine Learning Research (New York, New York, USA)* ed M F Balcan and K Q Weinberger (PMLR) pp 2990–9
- [35] García Satorras V, Hoogeboom E and Welling M 2021 E (n) equivariant graph neural networks *Int. Conf. on Machine Learning* (PMLR) pp 9323–32
- [36] Cohen T, Weiler M, Kicanaoglu B and Welling M 2019 Gauge equivariant convolutional networks and the icosahedral cnn *Int. Conf. on Machine Learning* (PMLR) pp 1321–30
- [37] Thomas N, Smidt T, Kearnes S, Yang L, Li L, Kohlhoff K and Riley P 2018 Tensor field networks: rotation-and translation-equivariant neural networks for 3D point clouds (arXiv: 1802.08219)

- [38] Salova A, Emenheiser J, Rupe A, Crutchfield J P and D'Souza R M 2019 Koopman operator and its approximations for systems with symmetries *Chaos* **29** 093128
- [39] Schmid P J 2010 Dynamic mode decomposition of numerical and experimental data *J. Fluid Mech.* **656** 5–28
- [40] Chen R T Q, Rubanova Y, Bettencourt J and Duvenaud D K 2018 Neural ordinary differential equations *Advances in Neural Information Processing Systems* p 31
- [41] Brunton S L and Kutz J N 2019 *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems and Control* (Cambridge University Press)
- [42] Arfken G B, Weber H J and Harris F E 2011 *Mathematical Methods for Physicists: a Comprehensive Guide* (Academic)
- [43] Thomas R 1999 Deterministic chaos seen in terms of feedback circuits: analysis, synthesis, labyrinth chaos *Int. J. Bifurcation Chaos* **9** 1889–905
- [44] Sprott J C and Chlouverakis K E 2007 Labyrinth chaos *Int. J. Bifurcation Chaos* **17** 2097–108
- [45] Lorenz E N 1963 Deterministic nonperiodic flow *J. Atm. Sci.* **20** 130–41
- [46] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems
- [47] Chollet F *et al* 2015 Keras (available at: <https://keras.io>)