# Structured dynamics in the algorithmic agent

Giulio Ruffini[1]*

December 12, 2023

**Abstract**

In the Algorithmic Information Theory of Consciousness (KT), algorithmic agents use compressive models inferred from world data to plan actions that maximize their objective function. What are the structural and dynamical consequences of tracking natural data generated by simple world models? To address this, we first propose a formalization of the concept of a generative model using the language of symmetry—Lie group theory. Then, using a generic neural network model as an agent system, we show that data tracking constrains the agent's dynamical repertoire, forcing it to mirror the symmetry of the generative world model. This narrows down both the space of potential parameters of the agent system and its dynamical repertoire, endowing both with structure inherited from the world. Based on these insights, we examine the link between data-tracking and the manifold hypothesis, which posits that natural high-dimensional data can be compressed into a reduced number of parameters due to the presence of a low-dimensional invariant manifold within the high-dimensional phase space. This work offers a new perspective for identifying neural correlates of agenthood and structured experience in natural agents and for developing AI and computational brain models.

*giulio.ruffini@neuroelectrics.com

**Highlights**:

- We define generative models using Lie groups.
- We associate the algorithmic simplicity of a model with representations of Lie groups or Lie algebras with special properties such as nilpotency and compositionality (hierarchical function composition).
- We show that world-tracking constraints force the tracking dynamical system to mirror the symmetry in the data.
- In this context, we revisit the manifold hypothesis and the implications for AI design and brain modeling.

# Contents


**1 Introduction** **4**

**2 Generative models, Lie groups and complexity** **6**

**3 The agent as a dynamical system** **17**

**4 The world-tracking condition** **24**

**5 Conclusion** **31**

**6 Appendix: notes of Lie groups** **41**

# 1 Introduction

Deciphering the fundamental principles underlying *agency* (behavior) and shaping *experience* is a central challenge for neuroscience and AI, with broad socioethical ramifications beyond *homo sapiens* [1, 2]. In a series of papers [3–6], we have put forth a framework anchored in Algorithmic Information Theory (AIT) to study the phenomenon of *structured experience*—defined as the spatial, temporal, and conceptual organization of our first-person experience of the world and ourselves as *agents* interacting with it—a central concept connecting first and third-person perspectives. Named after Kolmogorov complexity, Kolmogorov Theory (KT) posits that a key function of the brain is information compression [3, 4]. The ability to compress information is equivalent to having access to a model of the world and provides the foundation for cognition. This ability is instrumental in shaping the perception of structured reality. Furthermore, we hypothesized that the successful comparison of compressive models with data generates the *qualia* of structured experience in *algorithmic agents* [5, 6]. Algorithmic agents are the central players in KT, defined as computational constructs using compressive world models, goals, and planning, to interact effectively with their environment.

Our definition of *agent* is inspired but not limited to natural agents—information-processing systems equipped with elements critical for evolutionary success: homeostasis (preservation of self) and *telehomeostasis* (preservation of kin). An algorithmic agent is an information processing system with an objective function (goals) that interacts bidirectionally with the external world, inferring and running compressive models, planning, and acting to maximize its objective function. Examples include animals, plants, and life in general, as well as properly designed artificial intelligence systems combining a world modeling engine, an objective function, and a planning module [5, 6].

Viewing the brain through the lenses of *computation* and *compression* offers valuable insights. The computational viewpoint of brain function directly suggests that the brain operates as a dynamical system poised near criticality. Research in theoretical neuroscience points to the unique computational and informational characteristics that emerge at the critical points of complex systems [7–11]. Criticality is increasingly recognized as a guiding principle for the brain's complex behavior, facilitating the processing of diverse information in dynamic environments [12–22]. The concept of self-organized criticality, where systems naturally evolve to critical states, further enriches this view [23–25]. Neural systems are hypothesized to reach such states through evolutionary and developmental mechanisms driven by homeostatic plasticity [26–29].

We have argued elsewhere that, at least as seen by natural agents, the world (whatever is generating data on the other side of the agent) is, to a significant extent, simple and compressible [30]. This assumption, which mirrors the "unreasonable effectiveness of mathematics" [31] and Occam's razor [32, 33], underlies our rationale for the definition of agents, and it suggests that a prior for simplicity (compressibility, in the algorithmic sense) is a key heuristic exploited by agents modeling the world. Compression, through its algorithmic definition via Kolmogorov complexity, hinges on the presence of *symmetry* or *invariance* [34], as we discuss here in detail.

Being able to compress a dataset is equivalent to having access to a model of the data [34], which in turn implies there are patterns, invariances, or regularities in the data that can be exploited. AIT provides a solid foundation for this idea: Kolmogorov complexity is defined as the length of the shortest program capable of generating a dataset (e.g., images of a hand, as in [34]). Intuitively, the length and structure of such a short program/model will be associated with the type of symmetries in the dataset. We can think of a model compressing data as the object reflecting the symmetry of the data, and it is possible to associate every example in the dataset with the model, which is then the "invariant" entity.

As hypothesized in KT, which is a theory not only of cognition (behavior, third person) but also of experience (first person), the ability to track world data using compressive models provides special computational systems called *agents* the *structure* in phenomenal experience. The qualitative aspects of structured experience are associated with the algorithmic features of these models, which reflect the structure discovered in the data the agent collects. This structure ultimately reflects the structure inherent in the *generative model* of world data. Furthermore, one may reasonably expect that the dynamics generated by agent computations using such models, as observed in behavior or neural activity, inherit and reflect model-derived structure. Consequently, studying the relationship between the structure of world data, the computational features of the models being run by the agent (model structure), the *structure* of neurophenomenological (first-person *reports*), behavioral, and physiological data features (third-person records), can be used to test the hypothesis that *an agent has (or at least reports) structured experience in a way that mirrors the structure of the compressive models they use to track (structured) world data.*

As part of this ambitious research program, we focus in this paper on *the relationship between the structure of input data and the computational features of the successful world-tracking agent.* Through this analysis, we seek to provide a new perspective on phenomena such as changes in the complexity of neural data [20, 22, 35, 36] or the *manifold hypothesis*, which states that natural data (including neuroimaging data) forms lower-dimensional manifolds in its embedding space [37–39].

Finally, we note that the fundamental principles of symmetry and Lie groups, which have provided the foundation for modern physics, including relativity theory and quantum mechanics [40, 41], are now rapidly being adopted in machine learning (see, e.g., [42–46]) after pioneering work in the context of convolutional networks [47, 48]. This paper is a parallel development from the perspective of KT with applications in both in computational neuroscience and machine learning.

The next sections are organized as follows. We formalize the concept of *structure* using Lie group theory—the mathematical language for describing continuous symmetries—to provide a definition of *generative model.* We then study the implications of invariance or equivariance (symmetry) in neural networks — first in the simpler feedforward case, then in the recurrent, dynamical network, where the connection between symmetry and dynamics is provided by the *world-tracking condition.* The discussion is high-level and does not distinguish between natural

and artificial neural networks.

## 2 Generative models, Lie groups and complexity

Artificial neural networks used for classification can be seen to represent invariance with respect to a *group of transformations* [34]. Essentially, the group provides the means for grouping instances of what is to be similarly classified. Such a perspective offers a principled way to understand the representations learned by the neural network and the invariances it might acquire during training, potentially facilitating the analysis and visualization of the learned representations in terms of the group actions or guiding the design of network architectures.

### The invariant feedforward classifier

As an example, consider the case of a deep neural network trained for image classification. The task of the network is to provide an output of "1" if the image contains a cat and a "0" otherwise. We will assume that the space of all cat images can be produced by a *generative model* that can be represented, at least locally, by the action of a *Lie group* with a small set of parameters compared to the number of pixels in the image (v. the Appendix for mathematical definitions). The Universal Approximation theorem [49] guarantees that, as a function, this classifier can be approximated well by a sufficiently large neural network, and we also expect that a deep neural network will do this more economically if the data is *compositional* in nature [50, 51]. This generally will be the case if the generative model is succinct (a short program in the language of AIT).

Similarly, we can also analyze the case of a compressive autoencoder network. The universal approximation theorem also guarantees that, given some reasonable conditions, it is possible to construct a network to implement in sequence the functions that map a generated image to the parameters in the generative model and then back to the image with any desired degree of approximation.

The equations for a vanilla feedforward network (FFN) are

$$x^{(l+1)} = f\left(W^{(l)}x^{(l)} + b^{(l)}\right), \quad l = 0, 1, \ldots, L-1 \tag{2.1}$$

where $x^{(l)}$ is the vector of neuron activations at layer $l$, $W^{(l)}$ and $b^{(l)}$ is the weight matrix and bias vector at layer $l$, respectively, $f$ is the activation function, applied element-wise, and the input data is fed as $x^{(0)}$. More generally, the network can be economically described by an equation of the form[1]

$$x = f(Wx + b), \tag{2.2}$$

where $x$ is a vector encompassing the activations of all nodes in the network, structured in a way that respects the layered organization of the network. $W$ is a block matrix describing

[1] This is not the equation for an autoencoder, equating input and output. It is a generic equation providing a relation between the state of all the nodes.

the connections between the nodes, including the weights associated with each connection. Its structure reflects the layer-wise organization of the network. $b$ is a vector containing all the biases associated with the nodes. The function $f$ is applied element-wise to its vector argument, representing an activation function that may be specific for each node. This representation abstracts the network's operation as a single step, whereas in practice, the computation unfolds through a series of transformations as the input data propagates through the successive layers of the network, as described by Equation 2.1.

We now examine the notion of a generative model of data to be used to train the FFN.

### Generative models as Lie groups

Prior work in computer vision has highlighted the role of symmetries in image data [42, 52–54], and in particular those associated with the group of rotations, translations, and dilations in $n$-dimensional Euclidean space. These can be considered as an extension of the Special Euclidean group $SE(n)$ to include scaling transformations and is often referred to as the Similarity group, denoted as $\mathrm{Sim}(n)$ [55][2]. We recall that groups act on manifolds to effect transformations (see the Appendix for various definitions)

We aim to extend these ideas to more general *generative models* — the starting point in KT. To make the discussion tangible but without loss of generality, we continue with the example of a classifier or autoencoder of cat images. We will work with several mathematical objects: the universe of all images $\mathbb{R}^X$ (with $X$ the number of pixels[3]), the subset of cat images (a very large or infinite set), a generative model of this set, and a symmetry group $G$ transforming cat images into cat images.

Our first assumption is that the space of cat images can be produced using a simple, smooth, generative model of cat images mapping points from a finite-dimensional manifold $\mathcal{C}$ (the compressed, "cat image latent space") to image space. This aligns with the view that the notion of "cat" is intrinsically a compressive model, an invariance encoded as a generative model using a few parameters. We formalize this in the following definition.

**Definition 2.1.** By a *generative model* of images, we mean a smooth function mapping points in the configuration space manifold to image space, $f : \mathcal{C} \to \mathbb{R}^X$ so that $I_c = f(c) \in \mathbb{R}^X$ for $c \in \mathcal{C}$.

We may write this loosely in terms of some local manifold "patch" coordinates $\theta \in \mathbb{R}^M$ such

[2]In $\mathrm{Sim}(n)$, a typical element consists of a rotation matrix $R$, a translation vector $\vec{t}$, and a scaling factor $\lambda$. A point $\vec{x}$ in $\mathbb{R}^n$ is transformed under $\mathrm{Sim}(n)$ as $\vec{x}' = \lambda R\vec{x} + \vec{t}$, which can be expressed in matrix form using homogeneous coordinates [56]. Defining $\tilde{x} = (x, y, z, 1)$, this transformation can be represented as $\tilde{x}' = A\tilde{x}$ with

$$A = \begin{pmatrix} \lambda R & \vec{t} \\ 0 & 1 \end{pmatrix}$$

[3]Real images are digital, and the values that a pixel can take are finite. We ignore these details for now.

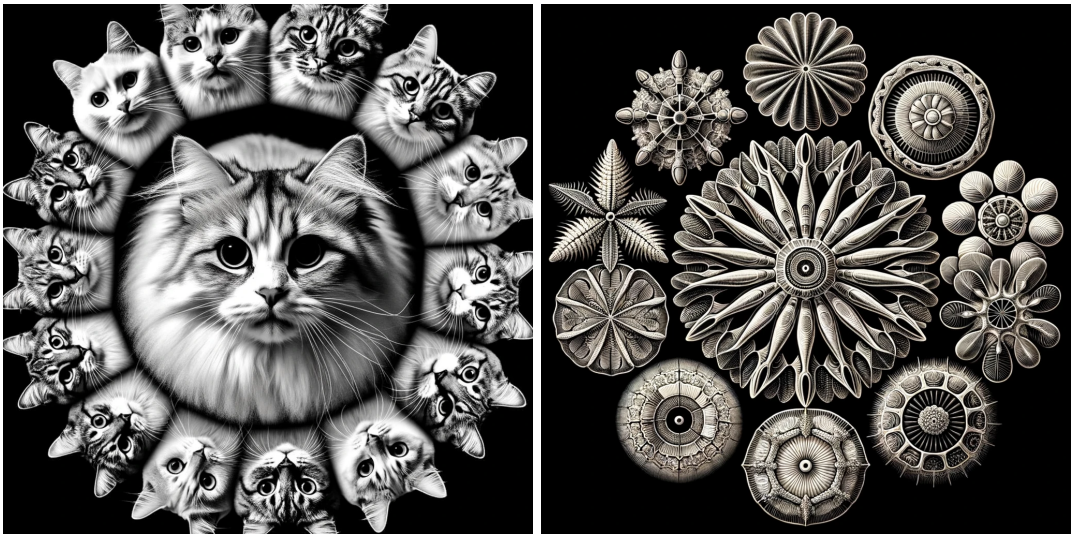

**Figure 1: Variations from canonical cat (left) or diatom (right).** Artist's representation illustrating cat and diatom images (in Ernst Haeckel's style) derived from a central archetype (center) via the transformative action of a Lie group in latent space. In such a generative model, any image can be used as an archetype due to the transitivity of the acting group.

that $I_\theta = f(\theta) \in \mathbb{R}^X$ (v. the Appendix for a definition of manifold). Here, $M$ is the number of parameters of the model, and $X$ is the number of pixels of the image generated, $I_\theta$. Parameters are coordinates of cat space manifold $\mathcal{C}$ (locally $\mathbb{R}^M$) — cat image *configuration space*.

A subset of the space of all possible images corresponds to the category of "cats". The symmetry group $G$ acts on the set of images of cats or, equivalently, their corresponding parameters and transforms them into other images of cats, formally defined as a maps $G \times \mathcal{C} \to \mathcal{C}$, and $G \times \mathbb{R}^X \to \mathbb{R}^X$, corresponding to $\gamma \cdot \theta = \theta'$ (configuration space) and $\gamma \cdot I \to I'$, with $\gamma \in G$ (image space).

The largest such group is the group of *automorphisms* (permutations) of cat images [57]. A subgroup of this is the diffeomorphism group Diff($\mathcal{C}$), the infinite-dimensional Lie Group [58] of *smooth* deformations of cat images into cat images.

To further formalize the notion of a Lie-group generative model, we start from a reference image of a "cat" and define continuous transformations to other images of cats. A desirable property of a generative model defined this way is that it should not depend on the choice of a reference image — the concept of "cat" should not hinge on a specific canonical cat. To ensure this, we require a *transitivity* requirement specifying that the group provides a transformation linking from any image to any other. Since we can appeal to the infinite-dimensional group Diff($\mathcal{C}$)=$G$, we know we can always associate a Lie group with the generative model. However, appealing to the simplicity of the generative model, and in particular, to the requirement of it having a finite number of parameters, we make the stronger assumption that the generative world model of cat

images can, at least locally, be associated with a *finite* Lie group $G$.

Finite Lie groups capture the essence of continuous symmetries in a compact, finite-dimensional form, making them highly amenable for theoretical study and practical computation. In contrast to the general diffeomorphism group $\mathrm{Diff}(\mathcal{C})$, which is often infinite-dimensional and unwieldy, finite Lie groups provide a structured approach to understanding global behavior from local information near the group identity element. This makes them particularly useful as candidates for simple generative models of world data, as they encapsulate key symmetries and invariances in a tractable manner through a few group generators.

This motivates the following definition, closely related to the notion of a **group of transformations** acting on a manifold (see the Appendix):

**Definition 2.2.** An $r$-parameter generative model $I = f(c)$, $c \in \mathcal{C}$, is a **(Lie) generative model** if it can be written in the form

$$I = f(c) = f(\gamma \cdot c_0) = \gamma \cdot I_0, \quad \gamma \in G \tag{2.3}$$

where $c_0$ is an arbitrary reference point, $I_0 \in \mathbb{R}^X$ is an arbitrary reference image, and $G$ is an $r$-dimensional Lie group.

For example, in the case of cat images in a particular manifold patch, we may write $f(\theta) = h(\gamma_\theta \cdot P_0) = \gamma_\theta \cdot I_0$, where $h$ is a projection function (describing the operation of a camera, which, for simplicity, we may assume to be fixed, with uniform background, etc.), and $P_0$ is an arbitrary reference point-cloud capturing the 3D form of a particular reference cat. In the second equality, $I_0$ is the corresponding reference cat image, and $\gamma_\theta$ is a representation of the group $G$ on that object.

In this example, the action (representation) of the group element $\gamma$ varies depending on which type of object (point cloud, image, or parameter) it is applied to, i.e., we have a different *representation* of the group corresponding to the same transformation. For example, in the case of point cloud data and rotations, we could describe the action of the group by matrices, i.e., we have a linear representation in a vector space (each point in the point cloud is encoded by three numbers, and the point cloud itself is simply an assembly of coordinates for each point). Linearity of representations may not generally hold—the representations found in natural settings are often non-linear, outside the domain of classical *representation theory* [59].

Representations of a finite-dimensional Lie group can be infinite-dimensional, i.e., in function space (Hilbert space). For example, if $\gamma$ acts on a point $x$ in space, then the object $f(x)$ (a function from configuration space to $\mathbb{R}$) changes to $f(\gamma \cdot x)$. If we think of $f(x)$ as an array indexed by $x$, i.e., $f_x$, we see that the group's action is to reshuffle the indices. This is a linear operation (it can be implemented by a matrix), and the action on such functions carries the group structure in an infinite dimensional space (as a linear operator, the analog of a matrix in infinite dimensions).

The assumption of finite dimensionality is not restrictive when the generative model depends on only one continuous parameter. It holds for generative processes associated with rotations, translations, and dilations since these correspond to finite Lie groups. It will also hold in the case of transformations of *shape space*, e.g., if the cat image generator source is a robot with any number of rigid joints, which can be described using the special Euclidean group $\mathrm{SE}(n)$ [60].

## Are all generative models Lie generative models?

Suppose we have a generative model for some dataset. This means that we can generate variants of the same object in the dataset (the dataset or the model of it can be seen as that which remains constant, e.g., "catness"). Thus, we talk about *symmetry*, *continuous transformations* (perhaps discrete ones, too) that leave the essence of the object (*catness*, or the dataset) invariant. We also expect that it is possible to rewrite the generative model using any coordinate system: there is no particular choice of coordinates in configuration space that is better than others, there is no particular origin, and no special cat.

The conditions under which the images generated using a simple computer program such as a generative neural network or an explicitly coded set of instructions in, e.g., a 3D modeling program like Blender [61] (both of which can be seen as compressive models) carry the group structure of a finite Lie group, are not obvious. But, in general, we may expect that any generative model displaying transitivity, smoothness and using a finite number of parameters may be described by a finite Lie group. This can always be done *locally* but may not be possible *globally*. As in robotics, it may be that the action of the Lie group is not fully realized (e.g., in some configurations of the system, some transformations may be constrained), but the notion is still useful (it can be generalized [59]). In any case, a Lie group structure allows us to navigate the configuration space manifold in a succinct manner. Recent work in this direction using natural data suggests it may be possible to exploit this structure [62–64].

If natural data can be captured in a latent space, e.g., neuroimaging data using a variational autoencoder [65], we can choose an arbitrary point in the latent space as a reference and study the relationship between displacements in different directions around that point. If this can be defined by a Lie group (at least locally), then this relationship should be independent of the choice of the reference point. E.g., the structure of the group rotations in the latent space of cat images should act independently of the choice of the reference cat.

We can refine the requirements for associating such a group with a generative model as follows. Let $M$ be a smooth manifold. We say that $M$ admits a *point-transitive finite-dimensional Lie subgroup* of $\mathrm{Diff}(M)$ if there exists a finite-dimensional Lie subgroup $G \subseteq \mathrm{Diff}(M)$ such that for any two points $x, y \in M$, there exists $g \in G$ such that $g(x) = y$. In the case of **1D manifolds**, every connected, smooth 1D manifold is diffeomorphic either to $\mathbb{R}$ or to $S^1$, the circle. For both these types, there exists a finite-dimensional Lie subgroup of the diffeomorphism group that acts point-transitively. Specifically, translations generate such a group for $\mathbb{R}$, and rotations for $S^1$. Thus, any generative model using these 1D manifolds is a Lie generative model.

However, not all configuration space manifolds have the property of finite Lie transitivity. Mostow's theorem provides insights into point-transitive actions on **surfaces (2D manifolds)** [66]. According to the theorem, a 2D manifold $M$ without boundary admits a point-transitive Lie group action if and only if $M$ is a plane, sphere, cylinder, torus, projective plane, Möbius strip, or Klein bottle. As we move to higher-dimensional manifolds, a comprehensive classification similar to Mostow's theorem becomes increasingly complex. Nevertheless, certain partial classifications and criteria exist that specify the conditions under which a manifold allows a transitive Lie group action.

Despite the difficulties in finding the conditions for the existence of global finite Lie groups acting transitively on a given manifold, transitivity can be expected in local Lie group action (see Appendix). The proof for *global* transitivity requires specific properties or structures on $C$. Establishing global transitivity is equivalent to showing that for any two points $c_1, c_2 \in C$, there exists an element in the Lie group that can transform $c_1$ into $c_2$. Mostow's theorem suggests there are deep links between the geometry of $\mathcal{C}$ and the algebraic structure of groups acting transitively on it.

In summary, if the data is associated with a generative model, we can always find a Lie generative model that generates it locally. Some generative models admit global group actions and are, therefore, Lie generative groups. This depends on the geometry of the generative model configuration space $\mathcal{C}$. It is nevertheless always possible to cast generative models as local Lie generative models.

## Lie groups and algorithmic complexity

We may conjecture that a highly compressive generative model *implies* the presence of a simple Lie group (characterized by a small, finite number of generators), a strong statement that will require bridging the gap between abstract algebra and algorithmic information theory[4]. The intuition behind this idea is simple: a compressive program capable of generating a large dataset will involve repeating an operation or a sequence of operations many times to produce the dataset. The recursive property inherent in Lie groups, which allows complex elements to be constructed from repeated applications of simple algebraic operations, is strikingly reminiscent of foundational concepts in algorithmic information theory and recursive function theory. This resemblance is particularly evident in the way information can be encoded recursively, akin to the compression of data, and how complexity arises in Turing machines through the application of simple, iterative procedures.

Let's consider first the other direction, i.e., that the presence of a Lie group structure in a generative model significantly influences its algorithmic complexity. If we can navigate a configuration space with a Lie group, any function on that space—including a generative model—can

[4]Note that the discussion revolves around the notion of algorithmic or Kolmogorov complexity, not *computational complexity* (roughly the resources needed to carry out the computation and another driving principle in KT since agents confront bounded resources).

be manipulated using the Lie group's generators. This can be highly efficient in some scenarios: by determining how to generate a single instance within our model, we can then apply the Lie group's transformations in small, discrete steps to produce a diverse set of outputs. This process is remarkably recursive and compressive, suggesting that an entire class of data, such as images of cats, could be generated from a relatively short program. As we now explain, the program would only need to encode the initial instance and the transformation rules derived from the Lie group's generators, thus enabling the generation of a vast array of images through a sequence of incremental, systematic transformations.

To see why, recall that a small transformation generated by a Lie group element $\gamma$ can be described as a perturbation around the identity element as $\gamma = 1_G + \epsilon T$, where $\gamma$ is a group element representing a small transformation, $1_G$ is the identity element representing no transformation, $\epsilon$ is a small parameter representing the "size" of the transformation, and $T$ is an element the Lie algebra, inducing infinitesimal transformations in the considered space (in our context, the "cat space"). This formulation offers a linear approximation of $\gamma$ in the vicinity of the identity element, utilizing the Lie algebra structure derived from the Lie group. It is a fundamental concept in working with continuous groups, facilitating the analysis of properties associated with symmetry and invariance. Similar to the exponential function's definition via a limit,

$$e^x = \lim_{n\to\infty} \left(1 + \frac{x}{n}\right)^n,$$

in Lie groups, elements are expressible as exponentials of Lie algebra elements, which has profound implications (refer to the Appendix). Specifically, any element within the connected component of the identity can be represented as a product of infinitesimally perturbed identity elements, denoted by $\gamma = 1 + \sum_k \epsilon_k T^k$. Formally, this is written as:

$$g = \lim_{n\to\infty} \left(1 + \frac{1}{n} \sum_k \theta_k T^k\right)^n = \exp[\sum_k \theta_k T^k]. \tag{2.4}$$

This *recursive* expression encapsulates the fundamental property of Lie groups that any group element (in the connected component) can be decomposed into an (infinite) series of infinitesimal transformations. More formally, any element of the connected component of any Lie group $G$ can be expressed as a finite product of exponentials, and if $G$ is a connected, compact matrix Lie group, the exponential map for $G$ is subjective (covers the entire group) [56].

As mentioned before, the above applies not only to the group but also to any representation of the group or, more generally, to the action of a group on a manifold. The action preserves the group structure and the number of its generators. The simplicity of the underlying Lie group is maintained for functions on a manifold under the action of the Lie group. In that case, the Lie algebra's infinitesimal generators $T_i$ (z) correspond to vector fields on the target manifold and the group action on a point $\theta$ is given again by the exponential map. This map is conceptualized as a "flow" along the vector field generated by $T_i(z)$. The transformation of the generative function $f$ under the group action is expressed by the series $\prod_n (1 + \epsilon z T(z) \nabla)^n$, representing the iterative application of the differential operator $zT(z)\nabla$ to $f$. The computation appears complex due to

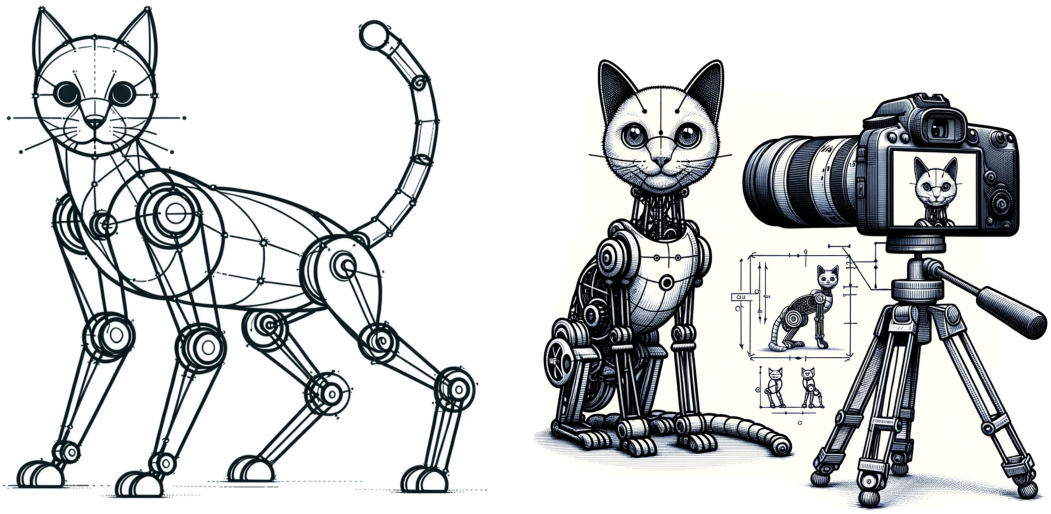

**Figure 2: Cat robot and generated cat image.** Artist's representation illustrating the generative model of a cat derived from a robot construction using the product of exponentials formula (Eq. 2.7) in robot kinematics. The robot consists of a set of joints controlled by a Lie group. Left: cat robot using joints to control pose and expression. Right, projection using a camera into an image.

the calculation of higher-order derivatives of $f$. However, these operators are linear, and the Lie algebra's structure constants and commutation relations may simplify the evaluation with recurrence patterns and algebraic properties such as *nilpotency* potentially reducing the number of distinct terms to be computed. This will depend on the character of the Lie group.

On the other hand, if a model can efficiently and globally encapsulate the complex variability of a dataset with a limited set of parameters and a short program length, it's plausible to conjecture that these parameters correspond to a Lie algebra with special properties. In the Appendix, we analyze it from the point of view of Turing machines as implicitly using groups (or some relaxed versions of them such as monoids) for computation.

This would be especially relevant in the context of compositional generative functions, where one can write

$$I(\theta) = f_n(f_{n-1}(...f_1(\theta))) \tag{2.5}$$

The compositional nature of the function — a form of recursion — is associated with a shorter set of computations to evaluate the function (see [50]). Thus, the computation of the generators of the transformations of the object $I$ will be simpler, too. An explicit form of a compositional Lie-generative model is (a special case of Eq. refeq:generative), with $\gamma = (\gamma_1, ..., \gamma_M)$,

$$I(\gamma) = f_M(\gamma_M \cdot f_{M-1}(...\gamma_2 \cdot f_1(\gamma_1 \cdot x_0))) \tag{2.6}$$

This hierarchical functional structure can be related to Poggio's conjecture, which states that the *compositional sparsity* of the underlying target function (i.e., the world), which corresponds

to the task to be learned, is the key principle underlying the success of machine learning [67], and in particular of deep learning architectures such as CNNs or transformers. Compositional sparsity, the idea that a function can be computed through the composition of functions, each with a constituent function of low dimensionality [68], is equivalent to efficient computability, that is, computability in polynomial time [67]. Thus, algorithmic and computational complexity notions may converge on the role of Lie action in generative (and compressive) models.

### Again, cats

To give a concrete example of a Lie compositional generative model, imagine generating cat images by taking pictures of a jointed cat robot (see Figure 2). The position or state of the robot reflects generative compositionality and is given by the Product of Exponentials formula used in robot kinematics [69],

$$T = \prod_{n \in \text{parents}} e^{[\mathbf{S}_n]\theta_n} M \tag{2.7}$$

Here $T \in SE(3)$ is the final end effector position of interest (a particular cat claw, for example) and orientation in the special Euclidean group, $\theta_n$ is the vector of joint variables for the $n$-th joint, $[\mathbf{S}_n]$ is the skew-symmetric matrix representation of the screw axis corresponding to the $n$-th joint, $e^{[\mathbf{S}_n]\theta_n}$ is the matrix exponential representing the action of the $n$-th joint's transformation, and $M$ is the initial home configuration of the end effector. In order to represent the full state of the robot, a similar expression can be written for each joint (not just one effector). Each joint carries a representation of a Lie group. The calculations, as well as the calculations of generators, are compositional. Using such a Lie-group compositional structure, we can generate an easily compressible dataset of robot pose images (much as we discussed in the case of hand images [34]).

The wider claim is that this principle applies to other world generative models (such as more realistic cat images). Fundamental physical laws display striking symmetry principles such as covariance (general relativity) or gauge invariance (quantum field theory). It is not obvious how these symmetries may percolate down after coarse-graining of phase space by the agent's sensory system, but it does not seem unreasonable that they would.

### Implications of invariance under group action

A required property of our cat classifying network is invariance under the action of the group $G$ regardless of its finite or infinite Lie nature. In other words, if the network classifies a point $\theta \in \mathcal{C}$ (via the image $I_\theta$ generated using the parameters corresponding to $\theta$) as a "cat," it should also classify the point $\gamma \cdot \theta$ as a "cat" for any $\gamma \in G$. If the group is a finite Lie group, it will suffice to study this for the *finite* number of generators of the group — linear operators in the Lie algebra of the group. As we saw, an important property of Lie groups is that any element of the group can be written in terms of the action of the group near the identity (with some nuances, as this applies only to the part of the group connected with the identity). The composition of group actions from elements near the identity covers the connected part of the group. This recursive

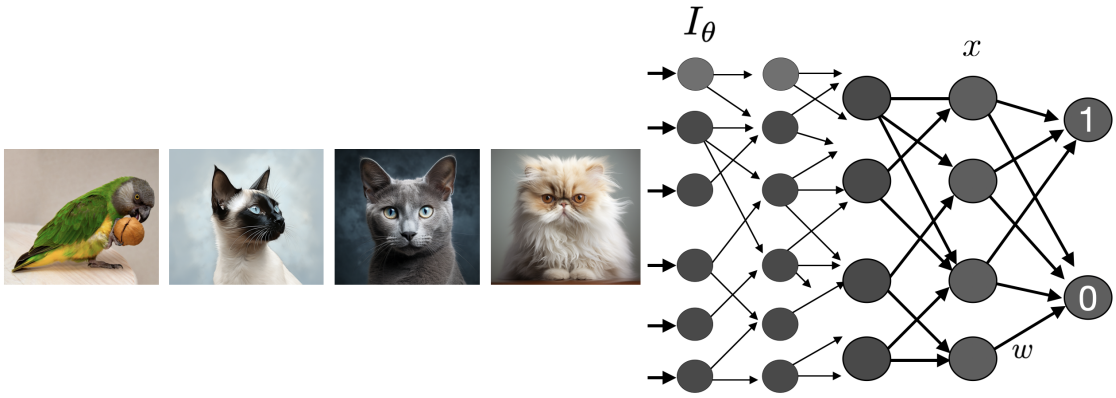


**Figure 3: Image classification task.** Classification of cat images can be seen as learning the invariances of a generative model. Such a network can also be implemented using an autoencoder of cat images with a skip connection to detect anomalies.

characteristic of Lie groups ensures that to establish invariance or equivariance of an object (such as a function on a manifold under group action), it is sufficient to verify this property for the generators $T_k$. The behavior under the action of infinitesimal transformations near the identity informs the behavior under the full group action. However, this does not necessarily hold if the Lie group is disconnected; in such cases, invariance must be confirmed for each component separately.

A change in the input image from a class-preserving infinitesimal transformation will propagate and affect the network state $x$. The transformation can be described by $\epsilon$ where $x \to \gamma \cdot x = x + \delta x$, with $\delta x = \epsilon T x$, where $\delta x$ is a small change in the state vector $x$, $T$ is a linear transformation generator associated with the group action, and $\epsilon$ is a small parameter. This induces the transformation

$$x' = f(W(x + \delta x) + b) = f(W(x + \epsilon T x) + b), \tag{2.8}$$

and we seek conditions under which the output of the network is invariant to first order in $\epsilon$.

Using the chain rule, we can express the transformation matrix at a given layer $l$ in terms of the transformation matrix at the input layer and the Jacobian matrices of the transformations at all the preceding layers,

$$T_l = J_{l-1} W_{l-1} \cdot J_{l-2} W_{l-2} \cdot \ldots \cdot J_1 W_1 \cdot T_0 \tag{2.9}$$

where $J_i$ denotes the Jacobian matrix of the transformations at layer $i$, encompassing the activation functions, and $T_0$ is the transformation matrix at the input layer, the linear approximation to $I_\theta = \gamma_\theta \cdot I_0 \approx f((1 + \epsilon T_0)\theta_0)$.

A necessary condition for the invariance of the neural network output with respect to the group action is that there exists at least one layer $l$ for which the condition $J_l \cdot W_l \cdot T_l x_l = 0$ holds for all

cat image inputs (see [63] for similar reasoning on the conditions imposed by symmetry). Note that the Jacobian depends on the input, too. This ensures that the transformed state vector $x_l + T_l x_l$ remains in the null space of the weight matrix $W_l$ at layer $l$, guaranteeing the invariance of the state of this layer with respect to the transformations generated by the group action. In other words, the space of cat images must be in the kernel of the operator $T_l$ for some layer $l$. This property will be inherited at higher layers as well, and the network output (the final layer) will be invariant to these changes.

This is a necessary condition for the network to perform well on cat images, but it may not guarantee that non-cat images are classified correctly — it may happen that some non-cat images are classified as cat images. However, a slightly more complex FFN can be used: an autoencoder of cat images exploiting the generative model can be used (as discussed below), and skip connections used for anomaly detection by comparing the input image with the autoencoded one. The more generators we find, the more constraints we can add to the training loss function.

For the case of a feedforward autoencoder, the analogous *equivariance* requirement is that the transformation of the output and the input be the same, namely,

$$T_N = J_{L-1} W_{L-1} \cdot J_{L-2} W_{L-2} \cdot \ldots \cdot J_1 W_1 \cdot T_0 = T_0 \tag{2.10}$$

when acting on cat images, which sets an $M$-dimensional family of constraints on $W$ requiring that the matrix $T_N = J_{L-1} W_{L-1} \cdot J_{L-2} W_{L-2} \cdot \ldots \cdot J_1 W_1$ have the space of cat images as an eigenspace corresponding to the unit eigenvalue.

The above are necessary requirements for invariance or equivariance under infinitesimal transformations. However, for a finite or infinite (with more nuances) connected Lie group, invariance under infinitesimal transformations guarantees global invariance/equivariance because any element in the group (at least the part connected with the identity) can be reached by repeated infinitesimal transformations [59]. This property is aligned with the idea of recursion in a compressive generative model.

At this point, we may reflect back on the motivation for using Lie group-related transformations. The smooth structure of Lie groups is the root of the link between infinitesimal and global invariance. *Smoothness* ensures the existence and uniqueness of solutions to the first-order ordinary differential equations associated with the flows generated by vector fields on the group by virtue of the Frobenius theorem [59]. Thus, smoothness facilitates the progression from local to global symmetries through the exponentiation of infinitesimal transformations, a crucial aspect for the study of continuous symmetry groups.

The link between this mathematical structure and the notion of recursion in a compressive generative model is quite natural. A compressive generative model essentially aims to capture and recreate complex data structures through a simplified, recursive algorithm. The correspondence arises from the fact that infinitesimal transformations, when recursively applied, generate the global structure of the Lie group, giving rise to a recursive and compressive generative model.

Systems specifically designed for invariant behavior, such as convolutional networks with pooling (using average pooling for example) are partially translationally invariant. These architectures explicitly implement the required symmetry [37, 42]. A generalization of this idea is to design the network to "filter" the data across group orbits (such as translations or rotations), then analyze the results (e.g., computing the maximum or the average) [43], or using data augmentation techniques, or finding fundamental architectural principles for group invariance [70]. The success of deep architectures may reflect this principle together with the hierarchical nature of group actions in the generative model.

In summary, requiring invariance of classification of data produced by a (simple) generative world model results in a set of equations regulating the behavior of the weight parameter matrix under transformations associated with the generative model. This can be useful in designing neural networks — where the weight equations can be enforced in the loss function during training — if the generators of the group are known. It can also be useful in the other direction: to infer symmetries underlying world data from networks trained on it.

# 3 The agent as a dynamical system

How do agents track world data? Although FFNs can be viewed as dynamical systems, they are limited to a computational universe bounded by a fixed number of sequential steps. A more comprehensive perspective, complementing the algorithmic and biological paradigms, emerges through the utilization of ordinary differential equations (ODEs), which possess the capability to instantiate universal Turing machines [71–73]. While the algorithmic perspective defines the agent as a program or Turing machine exchanging information with the environment, the dynamical view describes it less abstractly (but equivalently) and in a manner that makes more clear its connections to physics and biology — e.g., as a human brain or a neural network running on hardware. As a dynamical system, the agent is characterized by state variables and rules governing state transitions as a function of an external parameter denominated as *physical time*.

We briefly define some terminology used in what follows. The *phase space* in dynamical systems theory is a multidimensional manifold (see Appendix for a definition) where each point specifies the state of the system (the necessary and sufficient information to project the state in time). Finally, by the *geometry of the phase space*, we refer loosely to the structure and properties of the subspace of dynamical trajectories in space, including the presence of any special points or areas, such as *attractors* or *repellors*.

### A simple illustrative case

We begin with a simple toy model to study the role of symmetry and world-tracking conditions. Consider the constrained system of differential equations (or *differential algebraic equation*

[74])

$$\begin{aligned} \dot{x} &= 0 \\ \dot{y} &= 1 \\ x &= C \end{aligned} \tag{3.1}$$

The first two equations are regular ODEs, and the last is a constraint ($C$ is an arbitrary parameter). These equations are solvable for any value of $C$, because the general solution to the ODEs is

$$\begin{aligned} x &= A \\ y &= t + B \end{aligned} \tag{3.2}$$

Thus, it is possible to select the *integration constants* $A$ and $B$ that satisfy the constraint. This happy state of affairs is the result of the constraint not being independent of the conserved quantities of the system.

Note that the same conclusion would apply if the third equation in 3.1 were replaced by $f(x) = C$, for any function $f$ such that $C$ is in its range.

Another way to view the compatibility of the equations is that the ODE subsystem displays translational symmetry in $x$, with the symmetry $x \to x + \epsilon$. This means that if $x$ is a solution to the differential equations, then so is $x + c$, for any constant $c$. Hence, we can shift the trajectory $x$ by any value we wish and still obtain a valid solution to the ODE. The other necessary element for the existence of a solution is that $\dot{x} = 0$, which says that $x$ is constant.

In contrast, there is another symmetry in the system, $y \to y + \epsilon$, but $\dot{y} \neq 0$. Had we asked for $y = C$, no solution would exist. A consistent constraint would be $y = t + C$, for any $C$.

Conserved quantities in and ODE, such as $x = C$ above, can be used to create new coordinates with zero derivatives, which manifest the symmetry, just like $x$ in our example. The symmetry group shifts solutions across the conservation surface, i.e., it shifts the constant associated with the conserved law.

In Equation 3.1, we have two conserved quantities (corresponding to two constants of the motion, or integration constants), and two symmetries associated with two generators[5] associated with them,

$$\Phi_1 = x, \qquad x \to x + \epsilon,\ y \to y, \quad T_1 = \partial/\partial x \tag{3.3}$$

$$\Phi_2 = y - xt, \qquad x \to x,\ y \to y + \epsilon, \quad T_2 = \partial/\partial y \tag{3.4}$$

[5]The operator notation, e.g., $T_1 = \partial/\partial x$ corresponds to the generator of a local Lie group. The reason for this is that

$$(1 + \epsilon T_1)f(x) = (1 + \epsilon \frac{\partial}{\partial x})f(x) = (1 + \epsilon \frac{\partial}{\partial x})f(x) = f(x) + \epsilon \frac{\partial f(x)}{\partial x} \approx f(x + \epsilon)$$

which can be extrapolated (and verified using the Taylor expansion of the function) to $e^{\epsilon \partial/\partial x} f(x) = f(x + \epsilon)$.

The slightly more complex system

$$\begin{aligned} \dot{x} &= 0 \\ \dot{y} &= x \\ x &= C \end{aligned} \tag{3.5}$$

has the symmetry $x \to x+\epsilon$, $y = xt+b \to y+\epsilon t$, generated by $T_1$. This generator produces shifts in the two independent conserved quantities, $C_1 = x$ and $C_2 = y - xt$. The other symmetry is $x \to x$, $y \to y + \epsilon$, generated by $T_2$.

### General model

A general model of the state of neurons or other elements of a single dynamical system[6] receiving external inputs can be written in the form of a multidimensional ordinary differential equation (ODE),

$$\dot{x} = f\big(x; w, \eta(t)\big) \tag{3.11}$$

with $x \in \mathbb{R}^X$ a highly dimensional state variable representing the activity of neurons or neuron populations (e.g., membrane potential or firing rate), and where $w$ stands for connectivity parameters (and maybe others). For example, in a model of the human brain, each coordinate may represent the average firing rate of populations of neurons (a neural mass model) or a more detailed ternary code of silences, spikes, and bursts in each cell [75]. This finite-dimensional equation can be generalized to a partial differential equation (a neural field equation), but these important details are not essential in what follows.

The input term $\eta(t)$ reflects inputs from the world represented as an external forcing independent of the state of the agent that makes the equations non-autonomous (i.e., explicitly time-dependent). We can imagine it representing random kicks to the trajectory or a more steady, purposeful forcing from agent sensors. External inputs (which can be seen to re-program the agent in the metaphor of the Universal Turing machine) can force changes in the state trajectory

[6]The general equations for a Turing pair implemented as a network dynamical system are given by the autonomous system

$$\dot{x} = f_x\big(x; w_x, \eta_y\big) \tag{3.6}$$

$$\eta_x = o_x(x) \tag{3.7}$$

$$\eta_y \approx t_x(x) \tag{3.8}$$

$$\dot{y} = f_y\big(x; w_y, \eta_x\big) \tag{3.9}$$

$$\eta_y = o_y(y) \tag{3.10}$$

These equations express the idea that the outputs of a subsystem are the inputs of the other. They include the world-tracking condition for the agent. Here, $x$ is a multidimensional variable capturing the state of the agent, and $y$ captures the state of the world. The $w$s are connectivity weights in the network (or other parameters), and the $\eta$s represent the outputs of a subsystem, which become inputs for the other. The $o$ functions are variable selectors of the subset of variables that become outputs, and finally, $t_x$ is a variable selector in the agent that is to match the inputs from the world (the world-tracking subsystem). The third equation, $\eta_y = t_x(x)$ captures the world-tracking condition.

or even drive the landscape through a dramatic topological transition [22, 76]. For example, we can imagine changes in topology induced by time-varying parameters associated with bifurcations in the system.

We call Equation 3.11 the "fast time" dynamics equation [22]. It governs dynamics at short time scales (seconds), where connectivity parameters $w$ are assumed to be constant. In reality, $w$ may itself be dynamic at slower time scales (reflecting *plasticity* phenomena). This equation fixes a *dynamical landscape*, the geometry of trajectories in phase space defined by the set of ODEs. More precisely, the landscape is the manifold defined by the motion of trajectories in the phase space manifold (with local coordinates $x \in \mathbb{R}^n$). The dynamical landscape is determined by the form of the function $f()$ and by the parameters $w$, which may be mutable. In non-autonomous systems, the landscape will itself be dynamic.

Because of the time-dependent input term, it seems to be difficult to say very much about the behavior of the fast-time equation. However, there are two important properties of the dynamics that we will leverage. We will assume:

1. Agent inputs are generated by simple rules (the world is inherently simple), i.e., by a hierarchical generative model as discussed above.
2. The agent is able to track its inputs, i.e., uses an internal generative model to approximately match them.

We analyze the mathematical consequences of these statements below.

## Conserved quantities and symmetries

In the context of dynamical systems, *symmetry* is a transformation that maps a solution of the ODEs into another or, equivalently, maps the set of solutions into itself. More precisely, by the symmetry of an ODE, we mean that there is a group $G$ such that for all elements $\gamma \in G$, if $x$ is a solution to the equations, then $\gamma \cdot x$, a transformation of $x$ specified by the group element, is also a solution to the ODE. We allow for the possibility of readjusting some parameters in the ODE in the process.

Identifying symmetries in a system enables us to simplify its analysis because it allows the solutions to be transformed into one another through the symmetry operation, implying that the solutions can be grouped into equivalence classes under the symmetry group.

Symmetries can be categorized as discrete or continuous, and in the latter case they are described by local Lie groups. Paralleling the implications of Noether's theorem [40, 41, 77], the existence of symmetries can help in analyzing and solving ODEs. For example, the existence of a continuous symmetry means that we can locally define a coordinate system in which some of the coordinates do not appear in the equations other than in the derivatives (they are cyclic) [77]. Furthermore, there is a deep connection between symmetries and conservation quantities, as we now describe.

A *conservation law* is a map from state trajectories to a manifold. For example, energy conservation is a map from solutions to Newton's equations to real numbers. Momentum conservation is a map from solutions to $\mathbb{R}^3$.

The relationship between symmetry and conservation laws can be understood as follows. First, if the system is invariant under the action of a group, we can transform a solution into another through the action of a group element (this may work out for the entire space of solutions, or we may need to select representatives of each solution subspace to reach the entire solution space).

We can then index every solution using the group elements and a reference solution (or perhaps a small set of reference solutions). Thus, we can, in principle, construct a map from each solution to the representative of the set of group elements associated with the reference solution. All the points in the solution (the state trajectory) are thus mapped to a group element plus reference solution. This information is the conserved quantity. Thus, we can think of this process as the act of creating a structured labeling system for the solution space using the group.

As a well-known example in physics, in the classical Kepler problem (two-body problem in celestial mechanics), an autonomous ODE in six-dimensional phase space, a (bound) solution may be classified using conserved quantities corresponding to symmetries: the shape of an ellipse (eccentricity), its size (the energy or total angular momentum, one parameter), and the orientation of the plane containing it (three parameters). The latter results from the rotational symmetry of the problem and allows for the classification of the solutions with the orientation of the plane (the angular momentum). We can think of the specification of a solution in a constructive way: specify first the energy (one parameter) and eccentricity (one parameter) of an ellipse with its major axis along the $x$-axis and a minor axis in the $y$-axis. Then, rotate this plane in 3D (three parameters from the SO(3) or, equivalently, the SU(2) group). The existence of these five conserved parameters in the Kepler problem or in other similar central potential problems stems from symmetries of the equations (the O(4) group) [78].

## Constraints on solutions

Let us consider, for simplicity, a well-behaved autonomous system of $X$ equations and unknowns,

$$\dot{x} = f(x; w) \tag{3.12}$$

(with a smooth function $f$). The solution involves $X$ integration constants. Paraphrasing the classical argument in Landau and Lifschitz [79], since the equations of motion for such a closed system do not involve time explicitly, the choice of the origin of time is entirely arbitrary, and one of the arbitrary constants in the solution of the equations can always be taken as an additive constant $t_0$ in the time, and the solution written as $x = x(t + t_0, C)$, with $C \in \mathbb{R}^{X-1}$.

We can use one of these $X$ equations to express $t + t_0$ as a function of $x$ and the rest of the constants. We can think of the corresponding component of $x$, say $x_0$, as the "clock" (this is

actually what real clocks are)[7], and we write $t + t_0 = u(x_0, C)$. Substituting this, the solution can then be written as $x_n = x_n(x_0, C)$ for the other $X - 1$ components.

Thus, we have $X - 1$ constraints

$$\Phi_n(x) = x_n - x_n(x_0, C) = 0, \;\; n = 1, ..., X - 1 \tag{3.13}$$

that the solutions must satisfy.

Since solutions cannot cross, one may expect a one-to-one correspondence between $C$'s and solutions. However, this may fail when the constants of the motion are implicitly defined through non-injective functions. In cases where a constant of motion $Q(x)$ is expressed in terms involving non-injective functions, such as $Q(x) = \sin(C)$, the map from solutions to constants of motion is no longer one-to-one, leading to multivalued mappings.

Up to this multivalued ambiguity, we can now use the $X - 1$ equations in 3.13 to express the remaining $C$s as a function of $x$. We thus obtain $X - 1$ independent constants of the motion,

$$C_n = Q_n(x) \;\; n = 1, ..., X - 1 \tag{3.14}$$

so $\dot{Q} = 0$. We may rewrite this more rigorously by choosing a fixed branch of the solution.

Therefore, any given solution in $X$ dimensional space is constrained by $X - 1$ conserved quantities[8], which results, as we already knew from the start, in one-dimensional trajectories.

There cannot be any further independent conserved quantities unless the solution is a fixed point, but any combination of constants of the motion is a constant of the motion.

While the autonomous system of equations defined by $\dot{x} = f(x; w)$, where $f$ is a smooth function, allows us to find $X - 1$ conserved quantities $Q_n(x)$, it is crucial to consider the differentiability of these quantities across the entire phase space. Evaluating the time derivatives of these constants of the motion, we have

$$\dot{C}_n = \sum_m \frac{\partial Q_n}{\partial x_m} \dot{x}_m = \sum_m \frac{\partial Q_n}{\partial x_m} f_m(x) = 0,$$

implying the conservation over time. However, this expression entails potential singular points where components of $f(x)$ vanish or where the partial derivatives $\frac{\partial Q_n}{\partial x_m}$ become singular, rendering $Q_n(x)$ not well-defined. These singular points introduce barriers to implementing a global change

[7] We can even add such a coordinate to represent time by adding a corresponding phase space coordinate, e.g., by adding some physical mechanism such as a particle moving in space, so that $\dot{x}_0 = 1$ and $x_0 = t + t_0$.

[8] *Integrable* systems in the context of differential equations are distinguished by the presence of a full set of *analytic* constants of motion, equal in number to the system's degrees of freedom [80] These constants are not only independent but also in *involution* (i.e., their Poisson brackets vanish), a condition ensuring their functional independence. This concept extends beyond the mere existence of integration constants in solutions to ordinary differential equations, as discussed by Landau. In integrable systems, these analytic constants facilitate a detailed characterization of the system's dynamics, often allowing the motion to be confined to an invariant torus in the phase space, and enabling the system to be solved by quadratures. This precise structure of constants of motion imbues integrable systems with a degree of solvability and predictability uncommon in more general dynamical systems.

of coordinates in phase space using these conserved quantities, as they may not be differentiable at these points.

When it is possible to find globally such canonical variables, the system is called integrable. An *integrable system* in the context of dynamical systems and mathematical physics is one for which it is possible to find as many independent constants of motion as degrees of freedom, which allows for the system to be solved exactly through analytic methods. These constants of motion are typically associated with symmetries of the system. Integrability is a strong condition and is typically hard to achieve or prove for most systems, especially when going beyond two degrees of freedom.

**Transition to Canonical Coordinates**

To advance our understanding of the relationship between symmetry and conserved quantities, we ignore the difficulty of differentiability and assume we can transition to a local "canonical" coordinate system, where each coordinate but one ($x_0(t)$) remains constant, corresponding to the conserved quantities identified previously. Representing these coordinates as $Q_i$, with $i$ ranging from 1 to $X-1$, we have the transformed system represented as

$$\dot{Q}_i = 0, \quad i = 1, \ldots, X-1, \tag{3.15}$$

with only one active coordinate evolving with time. The solutions are $x_0 = x_0(t)$ for some function $x_0$ and $Q_i = C_i$. Clearly, any transformation that maps a solution to $Q_i \to Q_i' = Q_i + h_i(Q_i)$ generates a new solution, i.e., is a symmetry of the ODE. The new solutions is simply $(x_0(t+t_0, C'), Q')$.

In the canonical coordinates, each conserved quantity engenders a one-parameter symmetry characterized by transformations of the form

$$Q_i \to Q_i + \epsilon_i, \quad i = 1, \ldots, X-1, \tag{3.16}$$

yielding a Lie group with $X-1$ parameters.

The generators of these symmetries, fundamental to our discussion, can simply be denoted by partial derivative operators with respect to the respective coordinates,

$$V_i = \frac{\partial}{\partial Q_i}. \tag{3.17}$$

These generators capture the symmetry properties of the system, satisfying commutation relations indicative of the independence of the conserved quantities given by

$$[V_i, V_j] = 0, \tag{3.18}$$

The action of the group on the solution $(x_0, Q_i)$ can be described through the transformations ($x_0$ is left unaffected),

$$Q \to Q' = (1 + \epsilon V_n) \cdot Q_i = Q_i + \epsilon \delta_{ni} \tag{3.19}$$

with $\delta_{ni}$ being the Kronecker delta. That is, the effect of the transformation is a shift of the constants of the motion. Note that $(x_0, Q')$ is also a solution to the ODE since $\dot{Q}' = 0$. This demonstrates that $V_n$ is the generator of a one-parameter group of symmetries representing translations in the canonical coordinates.

In summary, we established that in a non-trivial $X$-dimensional first-order autonomous ODE, one can exploit time invariance to identify precisely $X - 1$ independent constraints, producing one-dimensional solutions (trajectories). Transitioning to canonical coordinates, when possible, where all but one coordinate ($x_0(t)$) remain fixed, at least locally, we elucidated that symmetry can be engendered through transformations mapping $Q_i$ to $Q_i + h_i(Q_i)$, yielding new solutions and therefore being symmetries of the ODE system. The generators $V_i = \partial/\partial Q_i$ foster a group of symmetries encapsulating translations in the canonical coordinates, underpinning a Lie group with $X - 1$ parameters.

Finally, our argument shows that there are constraints that lead to "local" conserved quantities and symmetries (and 1D trajectories), but there may not be such globally well-defined quantities. There may be singularities in the conservation functions [81].

As we have seen, $N$-dimensional ODEs have $N - 1$ conserved quantities (essentially functions of the constants of the motion). It is possible to solve ODEs with constraints if the constraints are functions of the constants of the motion (and hence constant). The conserved quantities can be used to create, at least locally, new coordinates, which are then constant. The symmetry of the equation moves these constants.
So, constrained ODEs can be solved if the constraints correspond to symmetries of the ODE, where a **symmetry** is a transformation of the set of solutions of the ODE (the family of solutions parametrized by the integration constants) into itself.

# 4 The world-tracking condition

We may think of the external world as providing stochastic but structured input to the dynamic system. One of the objectives of the agent is to track world data through the implementation of the world model in its modeling engine, which is to be compared with the data. This is essentially the notion of a compressive autoencoder if the world model is implemented using a neural network.

This, and the assumption that world data is the product of a simple generative model implies that the dynamics of the agent must be constrained and can be studied from the point of view of symmetry. To analyze this, we consider another concrete example: an agent watching a moving human hand generated by some simple physics-based computational model (such as Blender [82]), as we described in [6]. The image of the hand is projected onto a screen and captured by the agent's eye, producing a retinal image.

We represent the retina image by $I = I_\theta \in \mathbb{R}^Y$, with $Y$ the number of "pixels" in the retina (e.g., $Y = 10^6$, the approximate number of fibers in the optic nerve [83]) and $\theta \in \mathbb{R}^M$ a small set of parameters reflecting the state of about $M = 10^2$ ligaments and tendons (as may be generated by a realistic animation [82], for example). Although the space of all possible images has a large dimensionality $Y$, the space of all hand-generated images has a much smaller dimensionality $M << Y$. More precisely, the points in image space corresponding to hand images lie in an $M$-dimensional manifold embedded into a $Y$-dimensional box.

The agent is equipped with a brain consisting of a large network of neurons, and the brain state specified by $x \in \mathbb{R}^X$ ($X$ is very large, there are about $2 \times 10^{10}$ neurons in the human cerebral cortex). The dimensions of the brain, pixel, and model space are drastically different: there are many more neurons than data points (pixels) and very few model parameters (hand shape and orientation parameters), $M << Y << X$.

The generative nature of images can be understood as the action of a continuous group $\mathcal{G}$ on the parameter set of the hand model and, therefore, of the images, $\gamma \cdot I_\theta = I_{\gamma \cdot \theta}$. This equation states that the action of the group on image space can be understood as its action on hand parameter space. The set of all possible hand images (the hand manifold) is invariant under the action of $\mathcal{G}$. This set is also equivalent to the semantic notion of "hand" or the algorithmic pattern of "hand."

We start from Equation 3.11, which becomes $\dot{x} = f\big(x; w, I_{\theta(t)}\big)$, where, for simplicity, we have expressed the model inputs as $\eta(t) = I_{\theta(t)}$. The image may be preprocessed in some way prior to its input to the network (the agent may be tracking only some features of the image, not trying to match every pixel in the image), but this is not important here.

Unless $\theta$ is fixed, our equation is non-autonomous. Even so, the range of possible inputs is now greatly constrained by their simple generative nature. This facilitates the task of the agent, which is modeling and (approximately) matching these inputs. We represent this by the *world-tracking constraint*: there is a projection from neural states to inputs — the neural state variable $x$ carries a representation of the inputs via some projection operator (a variable selector in practice) $g(x)$. Putting together this with the neural dynamics equation, we have the *world-tracking neurodynamics equations (WTNE)*

$$\begin{aligned} \dot{x} &= f\big(x; w, I_{\theta(t)}\big) \\ g(x) &\approx I_{\theta(t)} \end{aligned} \tag{4.1}$$

with the inputs $I_{\theta(t)}$ produced by a simple generative model and some dynamical generation process of its parameters.

This a *differential algebraic equation* containing both differential equations and algebraic constraints. It will not have a solution for an arbitrary set of fixed weights $w$: once the input is fixed, the system dynamics and, therefore $g(x)$ are determined, which will not, in general, match the inputs. In fact, we may relax the problem by asking for approximate tracking, as indicated

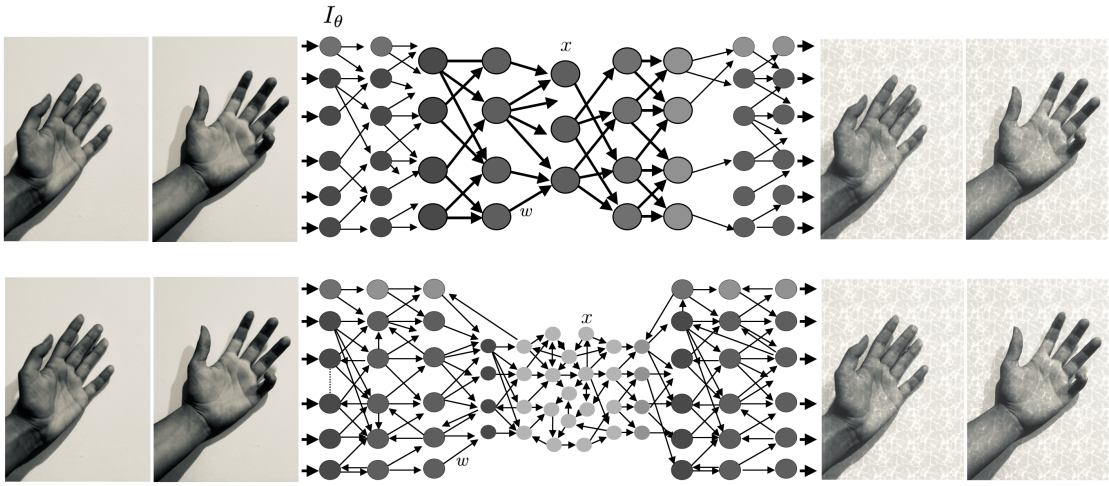


**Figure 4: World-tracking neurodynamics.** Tracking the world, in this case a set of frames of a moving hand, is essentially the task of a compressive autoencoder, which can be described by Equation 4.1. The top panel displays a feedforward (FFN) autoencoder, while the bottom provides a more general recurrent (RRN) autoencoder (connections going backward are added to highlight the potential use of predictive coding). Both realizations display an information bottleneck (latent space), where input data is mapped to generative model parameter space.

by the "≈" symbol in the second equation. This may be implemented, e.g., by a constraint of the form $||I_{\theta(t)} - g(x)||^2 < \epsilon$.

Another way to view Equations 4.1 is as an *autoencoder* of the input function, i.e., with the input $I_\theta(t)$ approximately equal to the output $g(x)$, as outlined in Figure 4. The network implementation is recurrent and reminiscent of architectures in Reservoir Computing [84–86], but it can also represent a feedforward system such as a variational autoencoder [87]—or something in between, as in Figure 4. In this machine-learning context, we rewrite the WTNE autoencoder equation as

$$\begin{aligned} \dot{x} &= f\big(x; w, I_{\theta(t)}\big) \\ w &= \arg\min_w ||g(x(t;w)) - I_{\theta(t)}||^2 \end{aligned} \tag{4.2}$$

where the dataset used for training is the generated image space discussed above.

In what follows, we assume the problem has been solved, with the $w$ parameters adjusted to provide such a solution (through some training process). We will also work with Equation 4.1 rather than Equation 4.2 for the sake of simplicity, although we should keep in mind that in a real scenario, we will be content with approximate solutions.

### The fixed input problem

The projection operator $g$ in Equations 4.1 restricts the dynamics of the state $x$ through $\dim(I)=Y$ time-dependent constraint equations. The conditions set by the equations must be met with fixed weights $w$ no matter what the world is doing as long as it is under the control of the generative process. In particular, if the inputs are constant (the hand is not moving or moving very slowly), we have

$$\dot{I}_\theta = \frac{d}{dt} g(x) = 0 \tag{4.3}$$

This implies that the dynamics are characterized by a large set of conserved quantities $g_n(x) = C_n$, $n = 1, ..., Y$. Since $g$ is a variable selector, this just means that some of the $x$ elements will be constant (the network readout layer). The trajectories in phase space will, therefore, lie in a reduced manifold defined by the coordinates $g(x) = C$. Having a constant readout given a constant input is trivially achieved with a feedforward "non-dynamical" architecture (as in the cerebellum [88]) but is more interesting in a realistic recurrent network such as the human cortex. Indeed, one may wonder if it can be at all possible. The mechanism for this is provided by the symmetry of ODE.

As we saw above in Section 3, the existence of these constraints implies the ODE has symmetry associated with them. This implies the parameters $w$ will satisfy some relations.

Another way of seeing this is that the only possible way of having a solution to Equation 4.1 is if the world-tracking constraint is compatible with the conserved quantities in the ODE — not an independent new constraint, which would force the dynamics to be a single point. Expressed in canonical coordinates, the world-tracking constraint must be a function of the cyclic (conserved) canonical variables. The world tracking condition requires that the system encode some conservation laws, which translate into local canonical variables where the variables corresponding to the conserved quantities are constant, i.e., $\dot{Q}_n = 0$. This subspace constitutes a center manifold with zero eigenvalues. This means that the generator of world inputs must be part of the symmetry group of the ODE system. Finally, as we saw, approximate symmetries can be achieved in steady state.

### The simple example revisited

Consider the constrained ODE

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 \\ x_1^2 + x_2^2 &= C^2 \end{aligned} \tag{4.4}$$

This problem has a solution because the general solution to the first two equations is

$$\begin{aligned} x_1 &= A \sin(t + t_0) \\ x_2 &= A \cos(t + t_0) \end{aligned} \tag{4.5}$$

for any constant $A$, which automatically guarantees that $x_1^2 + x_2^2 = A^2$. Hence, finding a solution to the three equations is possible, but only because the constraint is compatible with the conserved quantities of the ODEs. Adding an incompatible constraint, say $x_1 = 1$, would force fixing the time variable $t$ because the intersection of the trajectory with the constraint would be a single point. For more clarity, we may express the system of equations in symmetry manifesting canonical coordinates (polar coordinates, in this case, corresponding to $x_1$ and $x_2$),

$$\begin{aligned} \dot{\phi} &= 1 \\ \dot{r} &= 0 \\ r^2 &= C^2 \end{aligned} \tag{4.6}$$

A solution is possible because of the second equation, which manifests the invariance of the ODE component to rescaling of the variables (changes in radius, with generator $\partial/\partial r$), translations of $r$, $r \to r + \epsilon$. Again, the conserved quantity corresponds to a center manifold (zero real part eigenvalue of the system).

**Approximate symmetry after transients**

If the second equation is instead changed to $\dot{r} = -(r - C)$, we can have the constraint to hold in steady state after transients die out,

$$\begin{aligned} \dot{\phi} &= 1 \\ \dot{r} &= -(r - C) \\ r^2 &\to C^2 \end{aligned} \tag{4.7}$$

The equations encode the desired constraint value, and they are no longer translational invariant (in $r$). Rather, a simultaneous translation of $r$ and $C$ leaves the ODE system invariant. We can think of $C$ as an input and the equation $r \to C$ as a post-transient constraint. The solution is $r = C + ae^{-t} \to C$. If $C$ changes slowly compared to the transient, the equations will still hold. The input, instead of the initial conditions, implements the group control on the solution space.

Another example of how to achieve long-term symmetry (after transients die out) is

$$\begin{aligned} \dot{\phi} &= 1 \\ \dot{r} &= y \\ \dot{y} &= -y \\ r^2 &\to C^2 \end{aligned} \tag{4.8}$$

which is equivalent to

$$\begin{aligned} \dot{\phi} &= 1 \\ \dot{r} &= ae^{-t} \\ r^2 &\to C^2 \end{aligned} \tag{4.9}$$

with solution $r = C - ae^{-t}$. The trick is to make the equations for $r$ evolve in time to the desired form, $\dot{r} \to 0$, which then delivers the desired properties of symmetry and conservation law. In this case, unlike the previous one, the constraint satisfied is a function of the initial conditions.

### General case

To analyze the general case in more detail, we now split the variables $x$ into two blocks, $x = (z, u)$. With some abuse of notation, since $u = I_\theta$ is constant ($\dot{u} = 0$), we can write the generic fixed image world-tracking equations without reference to $u$ as

$$\begin{aligned} \dot{z} &= f_z\big(z; w, I_\theta\big) \\ I_\theta &\approx \sigma(w_u z) \end{aligned} \tag{4.10}$$

where for simplicity, we assume a feedforward read-out layer with a pointwise nonlinearity, with $g(x) = w_u z = u$ (recall that $g$ is a variable selector (linear) operator selecting a subset of $\sim 10^6$ $x$ variables). We use the approximate symbol in the second equation to allow for transients and approximate matching.

We assume the system is symmetric, i.e., we assume that if with the input $I_\theta$, $z$ is a solution, then so is $\gamma \cdot z$ with the input $\gamma \cdot I_\theta$, that is,

$$\begin{aligned} \frac{d}{dt}(\gamma \cdot z) &= f_z\big(\gamma \cdot z; w, \gamma \cdot I_\theta\big) \\ \gamma \cdot I_0 &\approx \sigma(w_u \gamma \cdot z) \end{aligned} \tag{4.11}$$

This is simply a consequence of the tracking constraints and the Lie-generative nature of the world model. The action of the Lie group on $z$ may be nonlinear, but the condition applies to an arbitrary infinitesimal transformation $\gamma = 1 + \epsilon^k T_k(z)$, i.e., in the linear regime (with $T$ a linear operator, see [46] for a more formal mathematical discussion.). These operators commute with time differentiation ($T_k(z)$ act on configuration space), hence

$$\begin{aligned} \gamma \cdot f_z\big(z; w, I_\theta\big) &= f_z\big(\gamma \cdot z; w, \gamma \cdot I_\theta\big) \\ \gamma \cdot I_0 &\approx \sigma(w_u \gamma \cdot z) \end{aligned} \tag{4.12}$$

for an infinitesimal transformation. Expanding $\gamma = 1 + \epsilon^k T_k(z)$, we can write a pair of such equations for each generator $T^k$ (linear operator) since $\epsilon$ is arbitrary and the generators linearly independent. We also have one constraint equation for the case $\epsilon = 0$.

As a consequence of the world tracking constraints associated with inputs from a generative model, we see that the system of equations must have $Y + M(Y + X)$ conservation laws or constraints in the system, which need to be accommodated by the weights (the structure of the equations). A group with a large number of generators will impose tighter constraints (conservation laws) on the possible $w$ space.

**Dynamical inputs**

As the input image moves slowly through changes in the $M$-dimensional parameters $\theta$ in the generative model, the reduced manifold of trajectories will also shift. Through this process, the manifold will increase its dimensionality, but it can only acquire a maximum of $M$ extra dimensions. Because the world model is assumed to be simple, i.e., with $M$ small, even in the case where the parameters $\theta$ are dynamic, the invariant manifold of the tracking system will remain dimensionally constrained.

The fact that images are generated by a simple model using a few parameters compared to the dimensionality of image space is crucial. In the example above, we analyzed the case where the image does not vary in time, i.e., a static input. If, on the contrary, the image is fully dynamic (each pixel varying in time independently of the others), there would be no reduction of the dimension of the trajectory space. There would be as many constraints as pixels, but they would be time-dependent, and possibly, no satisfactory weights $w$ would be found in such a case.

It is the intrinsic low-dimensionality of data generation that allows almost all the constraint equations to independently contribute conserved quantities: the image $I_\theta$ lies in a small dimensional manifold, and the operator $g$ must project $x$ into it.

The world-tracking equations are expected to hold in an approximate way and only after a transient time. As in the simple example above, it may be the case that the world-tracking constraints are realized as attractors in the dynamics. In the case of a dynamics input generated by dynamical parameters in the generative model, we can expect to see an approximate match of the output to the input only after a transient period of time.

In summary, and as we saw in the simple examples above, we assumed

1. The agent inputs are generated by simple rules (the world is inherently simple), i.e., by a hierarchical Lie generative model as discussed above.
2. The agent is able to track its inputs, i.e., encodes an internal generative model to approximately match them, perhaps after a transient time.

As a consequence,

1. As a dynamical system, the successful agent displays conserved quantities corresponding to the world-tracking constraints generated by a Lie group: starting from a reference constraint, the others are generated by a Lie group.
2. The constraints force symmetries in the dynamical system that meets them: the agent carries the symmetries of the world generative model.
3. This is encoded in the structural elements of the dynamical system (the $w$s in our formulation).

# 5 Conclusion

In this paper, we have studied the implications of tracking world data produced by a structured, generative model. As we have seen, the tracking system must be able to represent the symmetries in the data, constraining computational structural elements and its own dynamics. World-tracking conditions are satisfied by special, symmetric networks. This is because the equations have to have (approximately) conserved quantities compatible with the required outputs.

If the inputs are not varying, dynamics will lie in a reduced manifold of minimal dimension (constrained by the world tracking constraints). If they vary by obeying a generative world model, the dimension of the reduced manifold will increase, but little compared to the embedding neural dimension. If inputs violate the world model, the dynamics will no longer follow the world-tracking constraints, and tracking will fail. The dynamics may no longer be constrained.

## Revisiting the manifold hypothesis

At this point, we may revisit the manifold hypothesis. The *manifold hypothesis* posits that high-dimensional data, such as world data collected from sensory systems (e.g., images) or neuroimaging data, can be compressed into a reduced number of parameters due to the presence of a low-dimensional manifold within the high-dimensional phase space [37, 39, 89, 90]. Such manifolds can take various forms, such as attractors and unstable manifolds.

Trajectories in phase space are one-dimensional objects embedded in a high-dimensional space. The space of all solutions (with all possible initial conditions) can effectively fill the entire phase space. In what sense may we then talk about a reduced manifold? On the one hand, each of the trajectories may lie in a simple to describe low-dimensional manifold such as a plane (providing a *smooth representation* [37]). If the system has many global symmetries, this will be the case in the right coordinate system (which will exist), with the limit of the reduced manifold being of one dimension in the appropriate canonical coordinates. These may be discovered using deep learning techniques.

An analogy for manifold reduction can be made with the phenomenon of disk accretion in celestial mechanics, which is associated with the planar nature of planetary systems or galaxies. The underlying reason is symmetry: the laws of physics are rotationally invariant. This leads to a conservation law, the conservation of angular momentum. If we imagine a simulation of such a system with random initial conditions (non-planar), the system will have some non-zero angular momentum in some direction. If the system is dissipative (i.e., with negative eigenvalues in our prior discussions), then momentum will dissipate in the non-conserved directions, but it cannot dissipate in the conserved direction.

On the other hand, if the system is non-autonomous but has symmetries and the inputs are generated with compatible symmetries, the dynamics will again be captured in a reduced subspace.

In summary, a reduced dimension data manifold reflects the simplicity in the generative models

underlying the data (both tracking system and inputs) and of the system that has learned the generative model, with simplicity defined using symmetry properties.

## Connections with empirical observations

The simplicity associated with generative model processes may manifest in different ways. An autoencoder of recorded neural data from a brain exposed to time-varying world inputs will be able to regenerate, after training, the dynamics using only as many dynamical parameters as in the input world-generative model.

Empirical observations of dynamics indicate that data trajectories typically lie in a manifold of dimensionality much lower than $\mathbb{R}^X$ (the invariant manifold), which we also loosely identify with the *latent space* when using autoencoders to compress neural data.

Complexity analysis is also used frequently to characterize neuroimaging data. Intuitively, characteristics of the invariant manifold, such as its intrinsic dimensionality, will impact how much it can be compressed using tools such as the Lempel-Ziv algorithm [91].

Because of the impact of world-tracking constraints, the nature of reduced, invariant manifolds can be expected to depend on brain state and world conditions since both are crucially implicated in the task of world-tracking. This has implications for neuroimaging data collected from different wakefulness states (awake, REM, NREM) [65], under anesthesia, minimally conscious (MCS) or unresponsive wakeful state (UWS), Locked-In Syndrome (LIS), epilepsy or Alzheimer's disease (AD) [92], which should display different latent space features, including dimensionality (compressibility, complexity), or exhibit task dependence (e.g., eyes open or closed). This can be attributed, at least in part, to the world-tracking features associated with brain state or task.

In particular, constrained dynamics principles may explain certain empirical observations, such as the increase of EEG alpha power in the eyes closed condition (and reduced complexity [93]), anesthesia, or LIS, all of which are characterized by clamping down world inputs. On the other extreme, it may clarify how psychedelics, which interfere with sensory processing, disrupt and lift the system away from the expected world model, with network dynamics departing from the normative invariant manifold (which manifests as an increase in complexity [20]).

Tailored data analysis techniques stemming from this may include deep learning techniques using autoencoders to discover latent spaces and topological data analysis to characterize their structure.

## Applications for building agent models

Symmetry is an example of an algorithmic feature that systems that compute the same function must share, regardless of their specific implementation. In particular, this means that natural agents such as human brains implement symmetries inherent in world data in their architecture and, in turn, generate data with conserved quantities. This translates into reduced dimensionality

dynamics in invariant manifolds and may explain complexity features observed in neuroimaging data across brain states and neurological or psychiatric disorders. These and related techniques, such as topological data analysis, may help uncover further correlates of structured experience with applications in consciousness research in natural and artificial systems.

Importantly for applications in artificial intelligence [42–44, 47, 48, 62, 63, 94, 95] or computational brain modeling, the constraints on dynamics translate into structural constraints, i.e., conditions on the parameters of the model in order to make it compatible with world-tracking requirements. This can be useful to guide the design of artificial neural networks such as autoencoders, but also of biologically inspired *whole-brain models* (such as digital twins [96]). Both of these are neural network models characterized by a large number of parameters. Demanding that the models display symmetry and conserved quantities with low dimensionality can help reduce the potential degeneracy of fixing parameters from data (or choosing models [97]). Moreover, if the parameters are known, the system's dynamics can be studied through its symmetry properties and by analyzing the dimension, structure, and stability of its attractors (eigenvalue/bifurcation analysis), which represent the system's long-term behavior [59, 77].

For example, neural dynamics should reflect the spatio-temporal symmetries in the data they process. This includes time translation and space transformations such as rotations and translations. This will be useful once whole brain models are "connected" with the world (including models of the sensory system), receiving external inputs and generating outputs (e.g., to classify an input).

An important part of this program is to characterize more general symmetries in the world and translate them into generative world models and Lie groups. An approach to attack this may be the use of deep learning models that can produce the appropriate latent spaces or derive the Lie group generators from networks that have been trained on world data [44, 46, 54].

Finally, as a prior for simplicity in brain or network models and to regularize them, one can demand that the models be as symmetric as possible while respecting the data used to train them.

The historical trajectory of group theory — the mathematical theory of symmetry — is a testament to its fundamental role in shaping the landscape of modern mathematical and physical sciences. From its embryonic development at the hands of Évariste Galois and Paolo Ruffini [98]—who, to unveil the algebraic symmetries governing polynomial equations, created the seeds of group theory—, to the insightful works of Sophus Lie [59, 77], who extended these concepts into the realm of continuous groups to address the intricacies of differential equations, group theory is rich in its fruits. The subsequent epoch, marked by Emmy Noether's transformative theorem [40, 41], cemented the profound connection between symmetries and conservation laws, further elucidating the inherent simplicity underlying complex natural phenomena and driving the development of the Standard Model in physics. These historical milestones underscore the enduring quest for understanding through the lens of symmetry. As we have discussed here in the context of the algorithmic agent (KT), the convergent principles of algorithmic simplicity

and symmetry should continue to yield useful insights into the nature of artificial and natural intelligence.

## References


[1] Thomas Metzinger. "Artificial Suffering: An Argument for a Global Moratorium on Synthetic Phenomenology". In: *Journal of Artificial Intelligence and Consciousness* 08.01 (Mar. 2021), pp. 43–66. ISSN: 2705-0785. DOI: 10.1142/S270507852150003X. (Visited on 12/02/2023).

[2] *Ethics of Artificial Intelligence — Internet Encyclopedia of Philosophy*. hhttps://iep.utm.edu/ethics-of-artificial-intelligence/. Accessed on 2023-12-02. 2023. (Visited on 12/02/2023).

[3] Giulio Ruffini. "Information, complexity, brains and reality ("Kolmogorov Manifesto")". In: *arXiv* (2007). URL: https://doi.org/10.48550/arXiv.0704.1147.

[4] Giulio Ruffini. "Reality as Simplicity". In: *arXiv: 0903.1193* (2009). URL: https://arxiv.org/abs/0903.1193.

[5] Giulio Ruffini. "An algorithmic information theory of consciousness". en. In: *Neurosci. Conscious.* 2017.1 (Oct. 2017), nix019.

[6] Giulio Ruffini and Edmundo Lopez-Sola. "AIT foundations of structured experience". en. In: *J. AI. Consci.* 09.02 (Sept. 2022), pp. 153–191.

[7] Woodrow L Shew and Dietmar Plenz. "The functional benefits of criticality in the cortex". en. In: *Neuroscientist* 19.1 (Feb. 2013), pp. 88–100.

[8] Pedro A M Mediano and Murray Shanahan. "Balanced information storage and transfer in modular spiking neural networks". In: *arxiv* (Aug. 2017). arXiv: 1708.04392 [q-bio.NC].

[9] E Erten et al. "Criticality and information dynamics in epidemiological models". In: *Entropy (Basel)* 19.5 (Apr. 2017), p. 194.

[10] Mike Li et al. "Transitions in information processing dynamics at the whole-brain network level are driven by alterations in neural gain". In: *PLOS Computational Biology* 15.10 (2019), pp. 1–27. DOI: 10.1371/journal.pcbi.1006957. URL: https://doi.org/10.1371/journal.pcbi.1006957.

[11] Pedro A M Mediano et al. "Integrated information as a common signature of dynamical and information-processing complexity". In: *Chaos* 32.1 (Jan. 2022), p. 13115.

[12] G. Deco and V. K. Jirsa. "Ongoing Cortical Activity at Rest: Criticality, Multistability, and Ghost Attractors". In: *Journal of Neuroscience* 32.10 (Mar. 2012), pp. 3366–3375. DOI: 10.1523/jneurosci.2523-11.2012. URL: https://doi.org/10.1523/jneurosci.2523-11.2012.

[13] Gustavo Deco et al. "Key role of coupling, delay, and noise in resting brain fluctuations". In: *Proceedings of the National Academy of Sciences* 106.25 (2009), pp. 10302–10307. ISSN: 0027-8424. DOI: 10.1073/pnas.0901831106. URL: http://www.pnas.org/content/106/25/10302.

[14] Filippo Cieri et al. "Brain Entropy During Aging Through a Free Energy Principle Approach". In: *Frontiers in Human Neuroscience* 15 (2021), p. 139. ISSN: 1662-5161. DOI: 10.3389/fnhum.2021.647513. URL: https://www.frontiersin.org/article/10.3389/fnhum.2021.647513.

[15] Enzo Tagliazucchi et al. "Criticality in large-scale brain FMRI dynamics unveiled by a novel point process analysis". In: *Front. Physiol.* 3 (Feb. 2012), p. 15.

[16] Dante R Chialvo. "Emergent complex neural dynamics". en. In: *Nat. Phys.* 6.10 (Oct. 2010), pp. 744–750.

[17] Ariel Haimovici et al. "Brain organization into resting state networks emerges at criticality on a model of the human connectome". en. In: *Phys. Rev. Lett.* 110.17 (Apr. 2013), p. 178101.

[18] Daniele Marinazzo et al. "Information transfer and criticality in the Ising model on the human connectome". en. In: *PLoS One* 9.4 (Apr. 2014), e93616.

[19] Fran Hancock et al. "May the 4c's be with you: An overview of complexity-inspired frameworks for analyzing resting-state neuroimaging data". Mar. 2022.

[20] Giulio Ruffini et al. "LSD-induced increase of Ising temperature and algorithmic complexity of brain dynamics". In: *PLoS Comput. Biol.* 19.2 (Feb. 2023), e1010811.

[21] Matthijs Meijers, Sosuke Ito, and Pieter Rein ten Wolde. "Behavior of information flow near criticality". In: *Phys. Rev. E* 103 (1 Jan. 2021), p. L010102. DOI: 10.1103/PhysRevE.103.L010102. URL: https://link.aps.org/doi/10.1103/PhysRevE.103.L010102.

[22] Giulio Ruffini et al. *Neural Geometrodynamics, Complexity, and Plasticity: A Psychedelics Perspective*. Dec. 2023. DOI: 10.1101/2023.08.14.553258. (Visited on 12/06/2023).

[23] Henrik Jeldtoft Jensen. *Self-Organized Criticality: Emergent Complex Behavior in Physical and Biological Systems*. Cambridge Lecture Notes in Physics. Cambridge University Press, 1998. DOI: 10.1017/CBO9780511622717.

[24] Gunnar Pruessner. *Self-Organised Criticality*. Cambridge, England: Cambridge University Press, Aug. 2012.

[25] Kim Christensen and Nicholas R Moloney. *Complexity and criticality*. London, England: Imperial College Press, 2005.

[26] Gina G Turrigiano and Sacha B Nelson. "Homeostatic plasticity in the developing nervous system". In: *Nature reviews neuroscience* 5.2 (2004), pp. 97–107.

[27] Dietmar Plenz et al. "Self-Organized Criticality in the Brain". In: *Frontiers in Physics* 9 (2021). ISSN: 2296-424X. DOI: 10.3389/fphy.2021.639389. URL: https://www.frontiersin.org/article/10.3389/fphy.2021.639389.

[28] Janina Hesse and Thilo Gross. "Self-organized criticality as a fundamental property of neural systems". In: *Frontiers in Systems Neuroscience* 8 (2014). ISSN: 1662-5137. DOI: 10.3389/fnsys.2014.00166. URL: https://www.frontiersin.org/article/10.3389/fnsys.2014.00166.

[29] RL Carhart-Harris. "The entropic brain — revisited". In: *Neuropharmacology* 142.167–178 (2018).

[30] G. Ruffini. "An algorithmic information theory of consciousness". In: *Neurosci Conscious* (2017). DOI: 10.1093/nc/nix019.PMID:30042851.

[31] Eugene P. Wigner. "The Unreasonable Effectiveness of Mathematics in the Natural Sciences. Richard Courant Lecture in Mathematical Sciences Delivered at New York University, May 11, 1959". In: *Communications on Pure and Applied Mathematics* 13.1 (1960), pp. 1–14. ISSN: 1097-0312. DOI: 10.1002/cpa.3160130102. (Visited on 12/06/2023).

[32] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & sons, 1991.

[33] Ming Li and Paul Vitanyi. *An introduction to Kolmogorov Complexity and its applications*. Springer, 1997.

[34] Giulio Ruffini. "Models, networks and algorithmic complexity". In: (2016).

[35] Robin L Carhart-Harris et al. "The entropic brain: a theory of conscious states informed by neuroimaging research with psychedelic drugs". en. In: *Front. Hum. Neurosci.* 8 (Feb. 2014), p. 20.

[36] R L Carhart-Harris and K J Friston. "REBUS and the anarchic brain: Toward a unified model of the brain action of psychedelics". en. In: *Pharmacol. Rev.* 71.3 (July 2019), pp. 316–344.

[37] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: (2012).

[38] Bradley CA Brown et al. "Verifying the Union of Manifolds Hypothesis for Image Data". In: *The Eleventh International Conference on Learning Representations*. 2023. URL: https://openreview.net/forum?id=Rvee9CAX4fi.

[39] James M Shine et al. "The low-dimensional neural architecture of cognitive complexity is related to activity in medial thalamic nuclei". en. In: *Neuron* 104.5 (Dec. 2019), 849–855.e3.

[40] Emmy Noether. "Invariante Variationsprobleme". In: *Nachr. d. König. Gesellsch. d. Wiss. zuGöttingen, Math-phys. Klasse* (1918), pp. 235–257.

[41] Dwight E Neuenschwander. *Emmy Noether's Wonderful Theorem*. Baltimore, MD: Johns Hopkins University Press, Apr. 2017.

[42] Tomaso Poggio et al. *The computational magic of the ventral stream: sketch of a theory (and why some deep architectures work)*. Tech. rep. MIT-CSAIL-TR-2012-035. MIT Computer Science and Artificial Intelligence Laboratory, 2012. URL: https://dspace.mit.edu/bitstream/handle/1721.1/76248/MIT-CSAIL-TR-2012-035.pdf?sequence=1&isAllowed=y.

[43] Fabio Anselmi and Tomaso Poggio. "Representation learning in sensory cortex: A theory". In: *IEEE Access* 10 (2022), pp. 102475–102491.

[44] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. "Data-driven discovery of Koopman eigenfunctions for control". In: (2017).

[45] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. "Discovering conservation laws from data for control". In: (2018).

[46] Samuel E Otto et al. "A unified framework to enforce, discover, and promote symmetry in machine learning". In: (2023).

[47] K Fukushima. "Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". en. In: *Biol. Cybern.* 36.4 (1980), pp. 193–202.

[48] Y LeCun et al. "Backpropagation applied to handwritten zip code recognition". en. In: *Neural Comput.* 1.4 (Dec. 1989), pp. 541–551.

[49] Kurt Hornik. "Approximation Capabilities of Multilayer Feedforward Networks". In: *Neural Networks,* 4.2 (1991), pp. 251–257.

[50] T Poggio et al. "Why and When Can Deep – but Not Shallow –Networks Avoid the Curse of Dimensionality: a Review". In: *CBMM Memo* 058 (2016).

[51] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. *Learning Functions: When Is Deep Better Than Shallow.* Tech. rep. CBMM Memo No. 045. CBBM, 2016.

[52] Rajesh Rao and Daniel Ruderman. "Learning Lie Groups for Invariant Visual Perception". In: *Advances in Neural Information Processing Systems.* Ed. by M. Kearns, S. Solla, and D. Cohn. Vol. 11. MIT Press, 1998. URL: https://proceedings.neurips.cc/paper_files/paper/1998/file/277281aada22045c03945dcb2ca6f2ec-Paper.pdf.

[53] Tomaso Poggio. "The computational magic of the ventral stream". en. In: *Derm. Helv.* (Sept. 2011).

[54] Artem Moskalev et al. "LieGG: Studying learned Lie group generators". In: (2022).

[55] John Stillwell. *Naive Lie Theory.* Springer, 2008.

[56] Brian C. Hall. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction.* Springer, 2015.

[57] G. Ruffini. "Models, networks and algorithmic complexity". In: *Starlab Technical Note - arXiv:1612.05627* TN00339 (Dec. 2016). DOI: DOI:10.13140/RG.2.2.19510.50249.

[58] Rudolf Schmid. "Infinite-dimensional Lie groups and algebras in mathematical physics". en. In: *Adv. Math. Phys.* 2010 (2010), pp. 1–35.

[59] Peter J Olver. *Applications of lie groups to differential equations.* en. 1986th ed. Graduate texts in mathematics. New York, NY: Springer, Dec. 2012.

[60] J M Selig. *Geometric fundamentals of robotics.* en. 2nd ed. Monographs in Computer Science. New York, NY: Springer, Nov. 2004.

[61] Blender Online Community. *Blender - a 3D modelling and rendering package.* Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018. URL: http://www.blender.org.

[62] Xu Miao and Rajesh P N Rao. "Learning the Lie groups of visual invariance". en. In: *Neural Comput.* 19.10 (Oct. 2007), pp. 2665–2693.

[63] Artem Moskalev et al. "LieGG: Studying Learned Lie Group Generators". In: *Advances in Neural Information Processing Systems.* Ed. by Alice H. Oh et al. 2022. URL: https://openreview.net/forum?id=9sKZ60VtRmi.

[64] Mark Ibrahim, Diane Bouchacourt, and Ari S. Morcos. *Robust Self-Supervised Learning with Lie Groups.* 2023. URL: https://openreview.net/forum?id=qWt3YobXdwe.

[65] Yonatan Sanz Perl et al. "Generative Embeddings of Brain Collective Dynamics Using Variational Autoencoders". In: *Physical Review Letters* 125.23 (Dec. 2020). DOI: 10.1103/physrevlett.125.238101. URL: https://doi.org/10.1103/physrevlett.125.238101.

[66] Morris W. Hirsch. "Actions of Lie groups and Lie algebras on manifolds". In: *A Celebration of the Mathematical Legacy of Raoul Bott*. Ed. by P. R. Kotiuga. Proceedings & Lecture Notes Vol. 50. Providence, RI: Amer. Math. Soc., 2010, pp. 69–78. ISBN: 978-0-8218-4777-0.

[67] Tomaso Poggio. *Foundations of Deep Learning: Compositional Sparsity of Computable Functions*. CBMM Memo 138. Cambridge, MA, USA: Center for Brains, Minds, and Machines, MIT, Jan. 2023.

[68] Tomaso Poggio. *Compositional Sparsity: a framework for ML*. CBMM Memo 138. Cambridge, MA, USA: Center for Brains, Minds, and Machines, MIT, Oct. 2022.

[69] Kevin M Lynch and Frank C Park. *Modern robotics*. Cambridge, England: Cambridge University Press, May 2017.

[70] Devanshu Agrawal and James Ostrowski. "A Classification of $G$-invariant Shallow Neural Networks". In: (2022).

[71] C Moore. "Generalized shifts: unpredictability and undecidability in dynamical systems". In: *Nonlinearity* 4.2 (May 1991), pp. 199–230.

[72] Michael S Branicky. "Universal computation and other capabilities of hybrid and continuous dynamical systems". en. In: *Theor. Comput. Sci.* 138.1 (Feb. 1995), pp. 67–100.

[73] Daniel Graça and Ning Zhong. "Analytic one-dimensional maps and two-dimensional ordinary differential equations can robustly simulate Turing machines". In: *Computability* (Oct. 2022), pp. 1–28.

[74] Stephen Campbell, Vu Linh, and Linda Petzold. "Differential-algebraic equations". In: *Scholarpedia J.* 3.8 (2008), p. 2849.

[75] Zachary Friedenberger et al. "Silences, spikes and bursts: Three-part knot of the neural code". en. In: *J. Physiol.* (Oct. 2023).

[76] Gisela D Charó et al. "Noise-driven topological changes in chaotic dynamics". en. In: *Chaos* 31.10 (Oct. 2021), p. 103115.

[77] Peter E Hydon. *Cambridge texts in applied mathematics: Symmetry methods for differential equations: A beginner's guide series number 22*. en. Cambridge texts in applied mathematics. Cambridge, England: Cambridge University Press, Jan. 2000.

[78] D M Fradkin. "Existence of the dynamic SymmetriesO4andSU3for all classical central potential problems". In: *Prog. Theor. Phys.* 37.5 (May 1967), pp. 798–812.

[79] L D Landau and E M Lifshitz. *Mechanics*. 3rd ed. Oxford, England: Butterworth-Heinemann, Jan. 1982.

[80] Vladimir I. Arnold. *Mathematical Methods of Classical Mechanics*. 2nd. New York: Springer, 1989.

[81] H. Goldstein. *Classical Mechanics*. Second Edition. Reading, MA: Addison-Wesley Publishing, 1980.

[82] Yuwei Li et al. *NIMBLE: A Non-rigid Hand Model with Bones and Muscles*. 2022. DOI: 10.48550/ARXIV.2202.04533. URL: https://arxiv.org/abs/2202.04533.

[83] Raul M Sanchez, Gary R Dunkelberger, and Harry A Quigley. "The number and diameter distribution of axons in the monkey optic nerve". In: *Investigative Ophthalmology & Visual Science* 27.9 (1986), pp. 1342–1350. ISSN: 0146-0404.

[84] Herbert Jaeger. *The "Echo State" Approach to Analysing and Training Recurrent Neural Networks*. Tech. rep. 148. German National Research Center for Information Technology GMD Technical Report, 2001. URL: http://www.faculty.jacobs-university.de/hjaeger/pubs/EchoStatesTechRep.pdf.

[85] Mantas Lukoševičius. "A Practical Guide to Applying Echo State Networks". In: *Neural Networks: Tricks of the Trade* (2012), pp. 659–686. DOI: 10.1007/978-3-642-35289-8_36.

[86] Gouhei Tanaka et al. "Recent advances in physical reservoir computing: A review". In: *Neural Networks* 115 (2019), pp. 100–123. DOI: 10.1016/j.neunet.2019.02.013.

[87] Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[88] Ellen Boven et al. "Cerebro-cerebellar networks facilitate learning through feedback decoupling". en. In: *Nat. Commun.* 14.1 (Jan. 2023), p. 51.

[89] Lawrence Cayton. *Algorithms for manifold learning*. Tech. rep. UC San Diego: Department of Computer Science & Engineering, 2008. URL: https://escholarship.org/uc/item/8969r8tc.

[90] Anne E Urai et al. "Large-scale neural recordings call for new insights to link brain and behavior". en. In: *Nat. Neurosci.* 25.1 (Jan. 2022), pp. 11–19.

[91] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. 2nd ed. John Wiley & sons, 2006.

[92] Yonatan Sanz Perl et al. "Model-based whole-brain perturbational landscape of neurodegenerative diseases". In: *eLife* 12 (Mar. 2023). Ed. by Muireann Irish, Timothy E Behrens, and Jordi A Matias-Guiu, e83970. ISSN: 2050-084X. DOI: 10.7554/eLife.83970. URL: https://doi.org/10.7554/eLife.83970.

[93] Nadine Farnes et al. "Increased signal diversity/complexity of spontaneous EEG, but not evoked EEG responses, in ketamine-induced psychedelic state in humans". en. In: *PLoS One* 15.11 (Nov. 2020), e0242056.

[94] R P Rao and D H Ballard. "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects". en. In: *Nat. Neurosci.* 2.1 (Jan. 1999), pp. 79–87.

[95] Maurice Weiler et al. *Equivariant and Coordinate Independent Convolutional Networks. A Gauge Field Theory of Neural Networks*. 2023. URL: https://maurice-weiler.gitlab.io/cnn_book/EquivariantAndCoordinateIndependentCNNs.pdf.

[96] Giulio Ruffini et al. "Targeting brain networks with multichannel transcranial current stimulation (tCS)". en. In: *Curr. Opin. Biomed. Eng.* 8 (Dec. 2018), pp. 70–77.

[97] Johannes Borgqvist, Fredrik Ohlsson, and Ruth E Baker. "Symmetries of systems of first order ODEs: Symbolic symmetry computations, mechanistic model construction and applications in biology". In: (2022).

[98] Ian Nicholas Stewart. *Galois Theory*. 4th ed. Oakville, MO: Apple Academic Press, Mar. 2015.

[99] Martin Golubitsky and Ian Stewart. "Symmetry methods in mathematical biology". en. In: *São Paulo J. Math. Sci.* 9.1 (June 2015), pp. 1–36.

# 6 Appendix: notes of Lie groups

In the context of Ordinary Differential Equations (ODEs) [59, 77, 97, 99], **a symmetry** is a transformation that leaves the set of solutions of the differential equation invariant — a bijection of the set of solutions. This implies that if a function is a solution to the differential equation, then its transformation under the symmetry is also a solution and that this map is one-to-one (injective) and onto (subjective).

This definition is based on the notion of invariance under a transformation, a fundamental idea in symmetry analysis. Symmetries in ODEs can be continuous (such as scaling or rotation) or discrete (such as reflection), and they help in simplifying and solving differential equations as well as understanding their underlying properties.

## Definitions

**Definition 6.1.** A **group** is a set $G$ together with a group operation, usually called multiplication, such that for any two elements $g$ and $h$ of $G$, the product $g \cdot h$ is again an element of $G$. The group operation is required to satisfy the following axioms:

1. **Associativity.** If $g$, $h$, and $k$ are elements of $G$, then
$$g \cdot (h \cdot k) = (g \cdot h) \cdot k.$$
2. **Identity Element.** There is a distinguished element $e$ of $G$, called the identity element, which has the property that
$$e \cdot g = g = g \cdot e$$
for all $g$ in $G$.
3. **Inverses.** For each $g$ in $G$, there is an inverse, denoted $g^{-1}$, with the property
$$g \cdot g^{-1} = e = g^{-1} \cdot g.$$

**Definition 6.2.** An $m$**-dimensional manifold** is a set $M$, together with a countable collection of subsets $U_\alpha \subseteq M$, called *coordinate charts*, and one-to-one functions $x_\alpha : U_\alpha \to V_\alpha$ onto connected open subsets $V_\alpha \subseteq \mathbb{R}^m$, called *local coordinate maps*, which satisfy the following properties:

1. The coordinate charts cover $M$:
$$\bigcup_\alpha U_\alpha = M.$$
2. On the overlap of any pair of coordinate charts $U_\alpha \cap U_\beta$, the composite map
$$x_\beta \circ x_\alpha^{-1} : x_\alpha(U_\alpha \cap U_\beta) \to x_\beta(U_\alpha \cap U_\beta)$$
is a smooth (infinitely differentiable) function.

3. If $x \in U_\alpha$, $x' \in U_\beta$ are distinct points of $M$, then there exist open subsets $W \subseteq V_\alpha$, $W' \subseteq V_\beta$ with $x_\alpha(x) \in W$, $x_\beta(x') \in W'$, satisfying

$$x_\alpha^{-1}(W) \cap x_\beta^{-1}(W') = \emptyset.$$

**Definition 6.3.** An $r$-parameter **Lie group** is a group $G$ which also carries the structure of an $r$-dimensional smooth manifold in such a way that both the group operation

$$m : G \times G \to G, \quad m(g,h) = g \cdot h, \quad g,h \in G,$$

and the inversion

$$i : G \to G, \quad i(g) = g^{-1}, \quad g \in G,$$

are smooth maps between manifolds.

In general, a Lie group $G$ will be realized as a group of transformations of some manifold $M$ with each group element $g \in G$ associated a map from $M$ to itself [59]. It is important *not to restrict* our attention solely to linear transformations. Moreover, the group may act only locally, meaning that the group transformations may not be defined for all elements of the group nor for all points on the manifold.

**Definition 6.4.** Let $M$ be a smooth manifold. A local **group of transformations** acting on $M$ is given by a (local) Lie group $G$, an open subset $\mathcal{U}$, with

$$\{e\} \times M \subset \mathcal{U} \subset G \times M,$$

which is the domain of definition of the group action and a smooth map

$$\Psi : \mathcal{U} \to M$$

with the following properties:

1. If $(h,x) \in \mathcal{U}$, $(g, \Psi(h,x)) \in \mathcal{U}$, and also $(g \cdot h, x) \in \mathcal{U}$, then

$$\Psi(g, \Psi(h,x)) = \Psi(g \cdot h, x).$$

2. For all $x \in M$,

$$\Psi(e,x) = x.$$

3. If $(g,x) \in \mathcal{U}$, then $(g^{-1}, \Psi(g,x)) \in \mathcal{U}$ and

$$\Psi(g^{-1}, \Psi(g,x)) = x.$$

(Note that except for the assumption of the form of the domain $\mathcal{U}$, part (c) follows directly from parts (a) and (b).)

For brevity, we will denote $\Psi(g,x)$ by $g \cdot x$, and the conditions of this definition take the simpler form:

$$g' \cdot (h \cdot x) = (g' \cdot h) \cdot x, \quad g,h \in G, \quad x \in M,$$

whenever both sides of this equation make sense.

## Important theorems

This theorem shows that smoothness implies that the behavior of the group is already specified in any small patch containing the identity. In a sense, Lie groups are "cyclical": all the elements can be obtained by repeated action of a subset.

**Proposition 1.24.** [59] Let $G$ be a connected Lie group and $U \subseteq G$ a neighborhood of the identity. Also, let

$$U^k = \{g_1 \cdot g_2 \cdot \ldots \cdot g_k : g_i \in U\}$$

be the set of $k$-fold products of elements of $U$. Then

$$G = \bigcup_{k=1}^{\infty} U^k.$$

In other words, every group element $g \in G$ can be written as a finite product of elements of $U$.

Furthermore, if $G$ is a *connected*, *compact* matrix Lie group, the exponential map for $G$ is subjective (covers the entire group) [56].

Related to this, one can show that there is a one-to-one relationship between a local Lie group and the Lie algebra that describes its behavior near the identity (v. Theorem 1.54 in [59]). The latter can be fully described by a set of constants (the structure constants of the Lie algebra). Thus, finite Lie groups are intrinsically simple objects.

We next address the issue of local transitivity.

*Proposition* 1 (Local Transitivity). Let $C$ be the configuration space manifold. For any point $p \in C$, there exists a neighborhood $U_p$ around $p$ and a Lie group $G_p$ such that $G_p$ acts transitively on $U_p$ and transforms the local configurations of images within this neighborhood.

*Proof.* We need to show the existence of a Lie group that acts transitively on a local patch around any point $p$ in $C$. At any point $p$ in $C$, there exists a tangent space $T_pC$, which captures the infinitesimal displacements around $p$. This tangent space is a vector space, and its dimension is the same as the dimension of $C$. The general linear group $GL(n, \mathbb{R})$, where $n$ is the dimension of $T_pC$, can act on $T_pC$ by matrix multiplication. This action is transitive since for any two vectors $v, w$ in $T_pC$, there exists a matrix $M$ in $GL(n, \mathbb{R})$ such that $M \cdot v = w$. By the above point, around every $p$, there exists a local action of a Lie subgroup of $GL(n, \mathbb{R})$ that acts transitively on a local patch $U_p$ around $p$ in $C$. We can take $G_p$ to be this Lie subgroup. Hence, for every point $p$ in $C$, we have found a Lie group $G_p$ that acts transitively on a local patch $U_p$ around $p$. □

## Action of a Lie group on a manifold

The implications of Lie groups being generated by "repeating an operation many times" are profound. They imply that group elements can be written as exponentials of operators. Such

exponential operators, in turn, have important properties.

For example, if $f(z)$ is invariant under an infinitesimal transformation $(1+\epsilon T)$, so that

$$f([1+\epsilon T]z) = f(z)$$

then for small $\epsilon$, the function $f$ does not change to first order in $\epsilon$. This implies that the Lie derivative of $f$ with respect to the generator $T$ is zero, $\mathcal{L}_T f = T \cdot \nabla f = 0$. If this invariance holds for all infinitesimal transformations, then by repeating the process infinitely many times, which can be conceptualized as taking the limit as $\epsilon$ becomes infinitesimally small and the number of applications goes to infinity, one would sees $f$ to be invariant under the finite transformation generated by exponentiating $T$, that is $e^{\epsilon T}$:

$$\lim_{n\to\infty} f([1+\frac{\epsilon}{n}T]^n z) = f(e^{\epsilon T} z) = f(z)$$

More generally,

$$f(e^{\epsilon T} z) = f(e^{\epsilon z \cdot T \cdot \nabla} z) = e^{\epsilon z \cdot T \cdot \nabla} f(z)$$

The second equality can be checked using a Taylor expansion, or can be justified by checking it to first order and using the limit expression of the exponential as a repeated action of the operator. The first, which provides the connection with the tangent space of the Lie group, follows trivially from $\nabla_i z^j = \delta_i^j$, so that $\epsilon z \cdot T \cdot \nabla z = \epsilon z \cdot T$.

Thus, one can inspect the behavior of a function under infinitesimal transformations to infer some properties of the finite ones. Let $\hat{v}$ be the operator associated with the vector field $v$, as in $v = \sum_n \xi^n \partial/\partial z_n$, then the Lie series of a function $f(z)$ at a point $z$ can be expressed as:

$$f(e^v z) = \sum_{n=0}^{\infty} \frac{1}{n!} v^n f(z) \sim e^{\hat{v}} f(z)$$

where $\hat{v}^n$ denotes the $n$-th power of the Lie derivative operator $\hat{v}$ acting on the function $f$, and the symbol $\sim$ denotes that the series on the right is the Lie series expansion of the left-hand side with some abuse of notation.

The 1D case is a special instance where the manifold is $\mathbb{R}$, and the Lie group is the group of translations on $\mathbb{R}$. The Lie algebra element is $T = \frac{\partial}{\partial x}$, and the exponential map corresponds to adding $\lambda$ to the argument of $f$:

$$e^{\lambda \frac{\partial}{\partial x}} f(x) = f(e^{\lambda \frac{\partial}{\partial x}} x) = f(x+\lambda).$$

This is the standard Taylor series expansion of $f$ about $x$, which is the flow of $f$ under the action of the translation group.

### Invariance and equivariance

To describe the coordinated behavior of transformations induced by a group on different spaces, the terms *invariance* and *equivariance* are often used. The latter is the most general one, indicating that objects in different spaces transform appropriately under the action of the group. Specifically, they may carry different representations, but there exists a commutative diagram linking the action of the group in both scenarios. This can be represented as a commutative diagram,

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "g\cdot"'] & Y \arrow[d, "g\cdot"] \\ X' \arrow[r, "f'"'] & Y' \end{tikzcd}$$

In the diagram, $X$ and $Y$ are spaces, $f$ is a function from $X$ to $Y$, and $g$ represents the group action. The transformed spaces and functions are denoted by $X'$ and $f'$ respectively. The diagram commutes, meaning that starting from any object and following any path through the diagram results in the same outcome.

In more detail, for a function $f : X \to Y$ and a group action $g$, the function $f$ is said to be *equivariant* under the action of the group if the following relationship holds:

$$f(g \cdot x) = g \cdot f(x)$$

This means that applying the group action $g$ to the input $x$ and then applying the function $f$ is equivalent to first applying the function $f$ to $x$ and then applying the group action $g$ to the result.

For a function $f : X \to Y$ and a group action $g$, the function $f$ is said to be *invariant* under the action of the group if the following relationship holds:

$$f(g \cdot x) = f(x)$$

This means that applying the group action $g$ to the input $x$ does not change the output of the function $f$. In other words, the function's value remains unchanged or "invariant" under the transformation. It is worth noting that invariance can be considered a special case of equivariance where the transformation on the output space is the identity operation.

## Further Notes on Lie Groups

The original inspiration sources for Lie were differential equations. But this is now unfortunately mostly forgotten [59]: the associated Lie groups are not particularly elegant. E.g., a) non-semi-simple nor solvable, b) typically acting nonlinearly on the underlying space, outside (linear) representation theory, and c) may even be only locally defined near identity.

The full range of applicability of Lie groups to differential equations is yet to be determined [59].

Lie discovered that for continuous groups, the complicated nonlinear conditions of invariance of a system under group transformations can be replaced by equivalent but far simpler linear conditions associated with infinitesimal transformations under the generators of the group [59].

Infinitesimal symmetry conditions — the defining equations of the symmetry group — can usually be solved [59]. Once this is done, one can a) construct new solutions from known ones (this provides the means for defining equivalent classes of solutions), b) classify families of differential equations that depend on parameters or other form factors [59].

Normally, all other things being equal, one seeks the most symmetric equations possible (simplicity) [59].

In the case of ODEs, invariance under a one-parameter symmetry group implies that the order of the ODE can be reduced by one (in the first-order case, it can be solved by quadrature). Multiparameter symmetry engenders further reductions in order, but quadrature alone is not sufficient to solve them [59].

Symmetry can lead to unique bifurcation phenomena. Hopf bifurcations in symmetric systems can result in synchronized oscillations among a group of neurons due to symmetry constraints. This synchronization can be crucial for understanding phenomena like coherent neural oscillations in the brain.

In essence, the presence of symmetry in neural networks can impact the conditions and characteristics of Hopf bifurcations. The idea is that the network's symmetries play a role in shaping the bifurcation phenomena and the resulting synchronized oscillations observed in neural systems. The exact mathematical details of how symmetries influence Hopf bifurcations would depend on the specific network structure and equations governing the neural dynamics.

**Degeneracy of Eigenvalues and Symmetry:** When a dynamical system possesses certain symmetries, it can constrain the eigenvectors of the Jacobian matrix A. Symmetry can force some of the eigenvalues of A to be degenerate, meaning they have repeated values. The reason for this degeneracy lies in the symmetry-induced constraints on the dynamics. Symmetry often implies that there are multiple equivalent directions or modes of perturbation around an equilibrium point. These equivalent directions correspond to eigenvectors associated with the degenerate eigenvalues. Mathematically, the degenerate eigenvalues represent directions in which the linearized system's behavior is indeterminate due to the symmetry constraints. They indicate that there are multiple possible linear combinations of these eigenvectors that result in equivalent perturbations. Eigenvalue degeneracy can be associated with bifurcation points where the stability of the equilibrium point changes, and new dynamical behaviors emerge as system parameters vary. The degeneracy of eigenvalues can lead to complex dynamics, such as limit cycles or pattern formation.

It is important to distinguish symmetry in the equations from symmetries in the state or the solution, which may be affected by asymmetry in the initial conditions or external inputs (*symmetry breaking*).

**Symmetry in Equations** (Mathematical Symmetry): This refers to the inherent symmetries or invariances present in the mathematical equations that describe the system's dynamics. These symmetries are independent of initial conditions (ICs) or external inputs and are related to the structure of the equations themselves. For example, if the equations exhibit translational symmetry, it means that the system's behavior remains the same when spatial coordinates are shifted.

**Symmetries in the State or Solutions** (Physical Symmetry): This relates to the actual configurations or states that the system can adopt as solutions to the equations. These symmetries are observable and depend on the initial conditions, external inputs, and the system's dynamics. Symmetry in the state implies that the observable patterns or configurations of the system exhibit certain symmetrical properties. These symmetries can be influenced by asymmetries in ICs or external inputs.

The key point is that while the equations themselves may possess symmetries, the actual behavior or state of the system may or may not exhibit these symmetries, depending on the specific conditions and inputs. External factors, such as uneven initial conditions or spatially varying external inputs, can break the symmetry of the system's state, leading to the emergence of patterns and symmetry-breaking phenomena like Turing bifurcations.

**Symmetry Breaking** is a fundamental concept in understanding how complexity and spatial patterns emerge in natural systems, particularly in the context of spatially extended systems and phenomena like Turing bifurcations. It refers to a phenomenon where the physical state or solutions of a system deviate from or do not exhibit the symmetries that are present in the mathematical equations governing the system. In classical systems[9], symmetry breaking can arise due to the interplay between asymmetric initial conditions (ICs) or external inputs and the system's response to these factors. This response is often governed by intrinsic nonlinear dynamics within the system. However, it is crucial to note that the nonlinearities in the equations themselves do not inherently break symmetry; rather, they can amplify or manifest asymmetries introduced by the initial conditions or external inputs.

[9] In quantum systems, quantum fluctuations can also break the symmetry.