Structured dynamics in the algorithmic agent

G. Ruffini[1*], F. Castaldo[1], E. Lopez-Sola[1,2],
R. Sanchez-Todo[1,2] and J. Vohryzek[2,3]

[1] Brain Modeling Department, Neuroelectrics, Barcelona

[2] Computational Neuroscience Group, UPF, Barcelona

[3] Centre for Eudaimonia and Human Flourishing, Linacre College, U. of Oxford

v1. Dec 13th, 2023, v2. Oct 16th 2023[†]

Abstract

In the Kolmogorov Theory of Consciousness, agents utilize inferred compressive models to track coarse-grained data produced by simplified world models, capturing regularities that structure subjective experience and guide action planning. Here, we study the dynamical aspects of this framework by examining how the requirement of tracking natural data drives the structural and dynamical properties of the agent. We first formalize the notion of *generative model* using the language of symmetry from group theory, specifically employing Lie groups to describe the continuous transformations observed in natural data. Then, by using a neural network model to represent the agent system, we demonstrate that data tracking forces the agent to mirror the symmetry properties of the generative world model. This constrains both the constitutive parameters of the agent (structure) and its dynamical repertoire, imposing a hierarchical structure that reflects the organization of the world, aligning with the *manifold hypothesis*. Our findings connect the perspectives of algorithmic information theory (model, Kolmogorov complexity, simplicity), symmetry, and dynamics (group theory, reduced manifolds), providing new insights for identifying neural correlates of agenthood and structured experience in natural agents as well as enhancing the design of artificial intelligence and computational brain models.

[*]Corresponding author: giulio.ruffini@neuroelectrics.com

[†]Compiled on October 16, 2024.

Highlights

- We introduce generative models based on Lie groups, linking algorithmic simplicity to Lie group properties like nilpotency and hierarchical composition.
- Neural networks inherit structural constraints reflecting the symmetries in their training data.
- Symmetry, conservation laws, and world-tracking constraints are closely related, shaping how agents interact with the world.
- World-tracking compels agents, as dynamical systems, to reflect data symmetries, leading to reduced-dimensional neural manifolds.
- Hierarchical world data structures lead to coarse-grained, hierarchical constraints and manifolds.
- Mutual Algorithmic Information between the agent and the world manifests as shared symmetries in dynamic interactions.
- We explore the implications of these concepts for AI design and brain modeling.

Contents

1 Introduction **4**

2 Generative models as Lie groups **7**
2.1 Classifying model-generated cat images . . . 8
2.2 Implications of invariance under group action . . . 11

3 The agent as a dynamical system **14**
3.1 General model . . . 14
3.2 Conserved quantities and symmetries . . . 15

4 The world-tracking condition **17**
4.1 Constrained dynamics I . . . 18
4.2 Constrained dynamics II . . . 19
4.3 The fixed input problem . . . 21
4.4 Covariance of the equations . . . 21
4.5 Tracking time-varying inputs; the Comparator . . . 23
4.6 Coarse-graining, hierarchical constraints, and manifolds . . . 25

5 Discussion **28**
5.1 Connections with empirical observations . . . 28
5.2 Applications in AI and neurosynthetic agent design . . . 30

6 Conclusion **31**

A Appendix **42**
A.1 Equations for a Turing pair . . . 42
A.2 Notes on Lie groups . . . 43
A.2.1 Definitions . . . 43
A.2.2 Important theorems . . . 44
A.2.3 Action of a Lie group on a manifold . . . 45
A.2.4 Invariance and equivariance . . . 46
A.2.5 Further Notes on Lie Groups . . . 47
A.3 Groups, Turing Machines, and Generative Models . . . 50
A.4 Groups and algorithmic complexity . . . 52
A.5 Symmetry in ODEs: abstract vs. traditional definitions . . . 54
A.5.1 Olver's definition of symmetry of PDEs[1,2] . . . 54
A.5.2 Abstract Definition of Symmetry for ODEs . . . 54
A.6 Are all Lie generative models finite-dimensional? . . . 56
A.7 Conserved quantities and canonical variables . . . 58
A.7.1 Some examples . . . 59
A.7.2 A not-so-simple example of constrained dynamics . . . 60
A.7.3 Approximate symmetry after transients . . . 60
A.7.4 Transition to Canonical Coordinates . . . 62

1 Introduction

Understanding the fundamental principles underlying *agency* (cognition, planning, valence, and behavior) and shaping subjective *experience* (perceived structure and emotion) is a central challenge for both neuroscience and artificial intelligence (AI), with broad socioethical ramifications beyond *homo sapiens*.[3,4] In a series of papers[5–9] we proposed 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 ($\mathcal{K}$), Kolmogorov Theory (KT) posits that a key function of the brain is information compression.[5,6] The ability to compress information is equivalent to having access to a model of the world, provides the foundation for cognition, and is instrumental in shaping the perception of structured reality. Furthermore, to relate the framework to neurophenomenology, we hypothesized that the successful comparison of compressive models with data generates the *qualia* of structured experience in *algorithmic agents*.[6–8]

Thus, KT is both a theory of cognition (behavior, third-person observations) and *experience* (phenomenology, first-person science), where the qualitative aspects of *structured experience* ($\mathcal{S}$) are associated with the algorithmic features of agent world models and their associated valence ($\mathcal{V}$).[9] Since successful models reflect structure in world data, KT predicts that the structure of experience ultimately stems from the structure inherent in the underlying *generative model* of world data.

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 (Figure 2). This 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 AI systems combining a world modeling engine, an objective function, and a planning module.[7,8]

AIT is the generalization of Shannon information theory pioneered by Kolmogorov, Chaitin, and Solomonoff using computation theory (Turing).[10–12] The structure of models, central to KT, can be explored by instantiating the agent, a computational concept, as a dynamical system[a]. In this framework, models and other components of the agent are programs that are "run," and model and agent structure map directly onto dynamical structure,[8] where dynamics refers to the time evolution of variables in the agent program.

[a] This is natural because computation can be understood as a dynamical process.[13,14] For example, Turing machines can be viewed as discrete dynamical systems characterized by transitions (over time) between states.

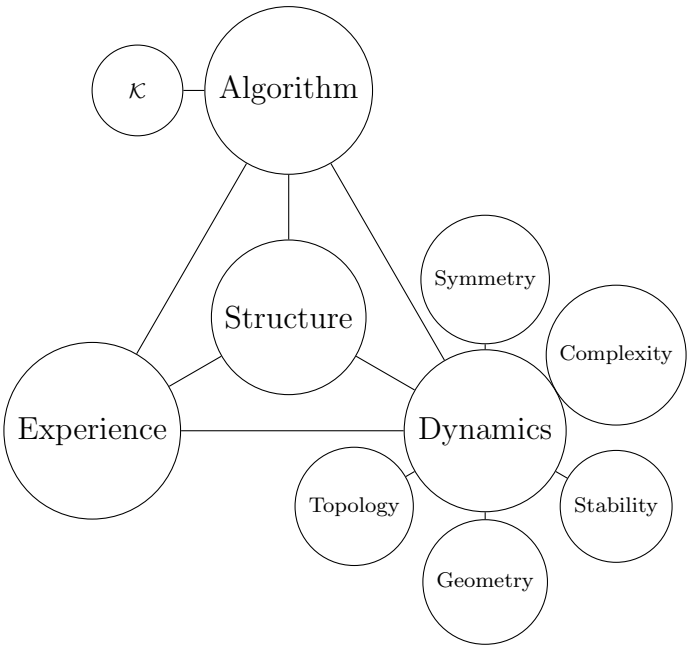

Figure 1: Diagram illustrating the relationship between underlying mathematical model Structure, Algorithm features (such as algorithmic complexity $\mathcal{K}$, i.e., program length, compositionally, recursion), Dynamics, first-person Experience (neurophenomenology), and in more detail the tools used in studying Dynamics, i.e., stability theory, bifurcation theory, chaos, geometry, topology, complexity (entropy, algorithmic), and symmetry.

The connection between algorithmic and dynamical perspectives provides powerful tools for analysis, including bifurcation theory, differential geometry, and topology, and suggests analyzing the brain's state dynamics and attractor landscapes for a deeper understanding of the mechanisms underlying neuropsychiatric phenomena and neurophenomenology, as depitect in Figure 1.

Viewing the brain through the lenses of *computation* and *compression* offers valuable insights for both theoretical and empirical research. The computational viewpoint of brain function directly suggests that the brain operates as a dynamical system with unique computational and informational characteristics emerging at the critical points of complex systems.[14–18] Criticality is increasingly recognized as a guiding principle for understanding the brain's complex behavior.[8,19–32] Neural systems are hypothesized to reach such states through evolutionary and developmental mechanisms driven by homeostatic plasticity.[33–37]

Regarding the role of compression, we have argued elsewhere that, from the perspective of natural agents, the world—defined as whatever is generating data on the other side of the agent—is, to a significant extent, simple and compressible.[7,8] This assumption, which mirrors the "unreasonable effectiveness of mathematics"[38] and Occam's razor,[11,39] underlies our rationale for the definition of agents as systems seeking to derive models of the world for survival, guided by a prior bias toward simplicity (compressibility, in the algorithmic information-theoretic sense).

AIT provides the foundational definition of compression: Kolmogorov complexity is defined as the length of the shortest program capable of generating a dataset (e.g., a stack of images of a hand[11]). As we discuss here, compression fundamentally relies on the

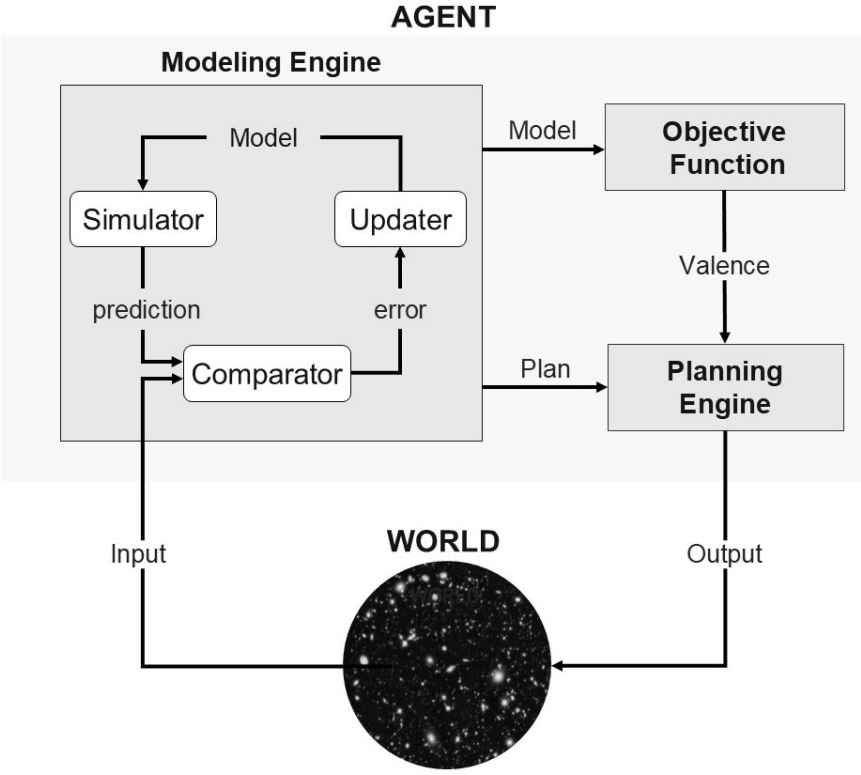

Figure 2: KT's algorithmic agent. An **algorithmic agent** is an information-processing system with an Objective Function that interacts bidirectionally with the external world, creating and running compressive models, planning, and acting to maximize its Objective Function.[7–9] The agent interacts dynamically with its environment, continuously exchanging data. The *Modeling Engine* runs the Model and makes predictions of future data (both from external interfaces as well as the agent's own actions) and then evaluates the prediction error (from the *Comparator*) to update the Model. The *Planning Engine* runs counterfactual simulations and selects plans for the next actions (agent outputs). The Updater receives prediction errors from the Comparator as inputs to improve the Model. The *Comparator* is a key agent element monitoring the success of the modeling engine in matching input data. We reflect this process mathematically as a world-tracking constraint on the dynamics (Equation 4.4, see also Section 4.5).

presence of *symmetry* or *invariance*.[5] The ability to compress a dataset is equivalent to having a model of the data, which implies the existence of patterns, invariances, or regularities that can be exploited.[40] Intuitively, the length and structure of such a minimal program or model will be closely related to the symmetries within the dataset. A model that successfully compresses data essentially reflects the underlying symmetries or invariances. In this sense, the model is the 'invariant' entity.

It is reasonable to expect that the dynamics arising from agent computations using these models—as observed in behavior or neural activity—inherit and reflect the structure derived from the models. Consequently, studying the relationship between the structure and symmetries 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 (third-person records), can shed light on the hypothesis that an agent has (or at least reports) structured experience in a way that mirrors the

structure of the compressive models it uses 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 (World) data and the computational (constitutive, structural) and dynamical features of the successful world-tracking agent*. Through this analysis, we aim to provide a new perspective on phenomena such as changes in the complexity of neural data[27–29,37,41] or the *manifold hypothesis*, which holds that natural data (including neuroimaging data) lies on lower-dimensional manifolds in its embedding space.[42–44] We include insights from the compositional (hierarchical) nature of world data and of neural processing and the associated necessity of coarse-graining.

Since invariance, structure, and symmetry are central to the problem at hand, we are naturally led to group theory and, more specifically, to the theory of continuous groups —Lie groups. The fundamental principles of symmetry and Lie groups, which have provided the foundation for modern physics, including relativity theory and quantum mechanics,[45,46] are now being adopted in machine learning (see, e.g.,[47–51]) after pioneering work in the context of convolutional networks.[52,53] This paper is a parallel development in the context of KT and, as we shall see, with applications in both computational neuroscience and machine learning.

In the next sections, we start from the mathematical framework of Lie groups to formalize the concept of *structure* and provide a definition of *generative model*. This foundation allows us to explore how the algorithmic simplicity of a model is linked to representations of compositional or *hierarchical* Lie groups, particularly those with special properties like nilpotency.

We then demonstrate that neural networks, such as feedforward networks, inherently acquire structural constraints from the symmetry properties of the data they are trained on. Translating this to the context of recurrent neural networks (RNNs) leads us to discuss the central role of *symmetry* and *conservation laws*, which provide the basis for the concept of *compatible world-tracking constraints*.

Building on this, we show that when a dynamical system aims to track the world, it is compelled to mirror the symmetries present in the data. This mirroring results in structural constraints and in the emergence of *reduced manifolds*, which are lower-dimensional spaces that encapsulate the essential features of the system while respecting the imposed constraints. The compositional nature of world data gives rise to the notion of compositional or *hierarchical constraints and manifolds*.

Our discussion is high-level and does not distinguish between natural and artificial neural networks, with implications for AI design and brain modeling. Finally, we discuss how, by integrating principles of symmetry, hierarchy, and manifold structures, we can enhance our understanding of neural processing and develop AI models that more closely emulate the efficient and hierarchical nature of biological brains.

2 Generative models as Lie groups

Prior work in computer vision has highlighted the role of symmetries in image data,[54–56] 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 $\mathrm{Sim}(n)$.[57] We recall that groups act on manifolds to effect transformations (see the Appendix for various definitions).

Artificial neural networks used for classification can be seen to represent invariance with respect to a *group of transformations*.[40] Essentially, the group provides the means for grouping instances of what is to be similarly classified. This perspective offers a principled way to understand the representations learned by the neural network and the invariance properties 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. We aim to extend these ideas to more general *generative models*—the starting point in KT.

2.1 Classifying model-generated cat images

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[b]), 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—see Figure 3. 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 can locally represent the generative model using coordinates on the manifold, which we refer to as 'patch' coordinates, denoted by $\theta \in \mathbb{R}^M$. In this context, $I_\theta = f(\theta) \in \mathbb{R}^X$ represents the image generated by the model, where M is the number of parameters defining the model, and X is the number of pixels in the generated image I_θ. 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". Suppose there is a group G that acts on the set of images of cats or, equivalently, their corresponding parameters in configuration (latent) space 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 group of transformations that can act on cat images while preserving their structure is the group of *automorphisms*, which in this case refers to all possible permutations of cat images.[40] Within this large group, there exists a specific subgroup known as the diffeomorphism group, denoted as $\mathrm{Diff}(\mathcal{C})$. This subgroup is an infinite-dimensional

[b] Real images are digital, and a pixel can take a finite number of values—we ignore such details.

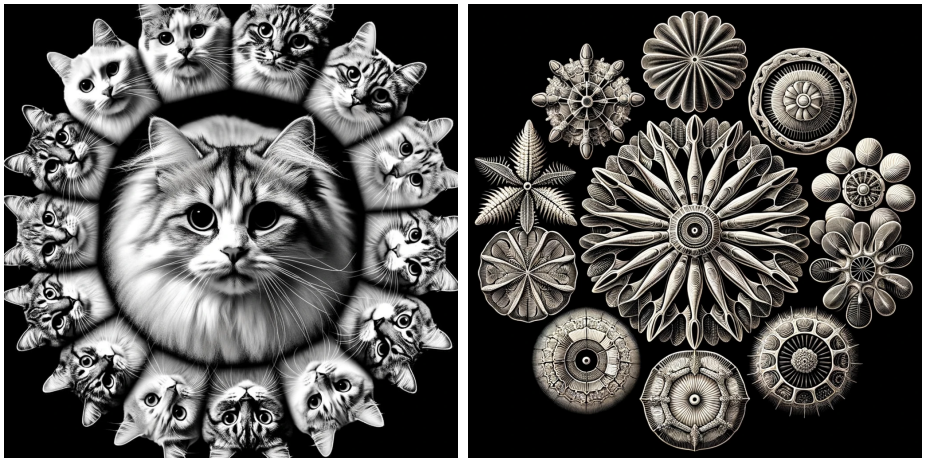

Figure 3: Variations from canonical cat (left) or diatom (right). An illustration of cat and diatom images (in Ernst Haeckel's style) derived from a central archetype (center) via the transformative action of a Lie group in latent space. In such a generative model, any image can be used as an archetype due to the transitivity of the acting group.

Lie group[58] that consists of *smooth* (i.e., continuously differentiable) transformations or deformations of the configuration space manifold, mapping cat images to other cat images through the generative model.

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 achieve this, we impose a *transitivity* requirement, meaning that the group of transformations must be able to link any cat image to any other cat image.

Since we can appeal to the infinite-dimensional group $\mathrm{Diff}(\mathcal{C})=G$, we know we can always associate a Lie group with the generative model. However, to keep the generative model simple, particularly by ensuring it has 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, 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):

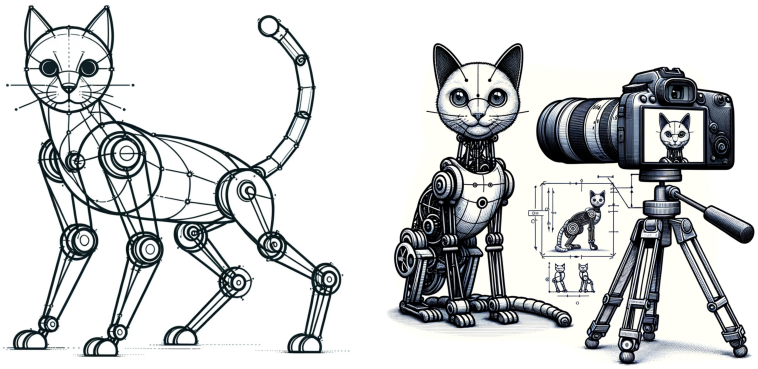

Figure 4: Cat robot and generated cat image. A representation of the generative model of a cat derived from a robotic construction using the product of exponentials formula (Eq. 2.2) in robot kinematics. The robot consists of a set of joints controlled by a Lie group. Left: cat robot using joints to control pose and expression. Right, projection into an image using a camera.

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

$$I = f(c) = f(\gamma \cdot c_0) = \gamma \cdot I_0, \quad \gamma \in G \tag{2.1}$$

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 γ_θ is a representation of the group G on that object. The action of the group element γ varies depending on the type of object it is applied to—whether a point cloud, an image, or a parameter—resulting in different *representations* of the group for the same transformation. For instance, when dealing with point cloud data and rotations, the group's action can be described by matrices, giving us a linear representation in a vector space. However, linearity of representations may not generally hold—the representations found in natural settings are often non-linear, outside the domain of classical *representation theory*.[1]

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 $\mathrm{SE}(n)$.[59] More generally, 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[60] (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 reasonably 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 (the notion can however be generalized[1]). Recent work in this direction using natural data suggests it may be possible to exploit this structure.[56,61,62] We discuss the limitations of the finite dimensional assumption in the Appendix.

A Lie group structure will allows us to navigate the configuration space manifold in a succinct manner. By using Lie groups, models can capture the essential characteristics of objects (like cats) in a mathematically rigorous and computationally efficient manner. This approach facilitates both the understanding of the learned representations in neural networks and the practical implementation of generative models.

In summary, data produced by a generative model can always be locally generated by a finite-dimensional Lie group. Some generative models admit global finite dimensional Lie-group actions and are, therefore, Lie generative models. This will hinge on the geometry of the generative model configuration space $\mathcal{C}$. It is nevertheless always possible to locally cast generative models as Lie generative models.

2.2 Implications of invariance under group action

To illustrate a Lie compositional generative model and study the implications of invariance, imagine generating cat images by taking pictures of a jointed cat robot (see Figure 4) for training a deep neural network classifier. The robot's state embodies generative compositionality and is described by the Product of Exponentials formula from robot kinematics,[63]

$$T = \prod_{n \in \text{parents}} e^{[\mathbf{S}_n]\theta_n} M \tag{2.2}$$

where $T \in SE(3)$ is the final position and orientation of the end effector (e.g., a particular cat claw) in the special Euclidean group, θ_n is the vector of joint variables for the n-th joint, $[\mathbf{S}_n]$ is the skew-symmetric matrix representing the screw axis of 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.

To represent the full state of the robot, a similar expression is applied to each joint and, by extension, to the entire robot. Each joint corresponds to a representation of a Lie group, and the calculations—including those involving the generators—are compositional. Leveraging this Lie-group compositional structure, we can generate an arbitrarily large yet highly compressible dataset of robot pose images.[40]

This principle may extend to other generative models of the world, such as more realistic images of cats. Fundamental physical laws often exhibit symmetry principles, such as covariance in general relativity or gauge invariance in quantum field theory. While it is not immediately clear how these symmetries propagate after the coarse-graining of phase space by an agent's sensory system, it is plausible that they persist in some form. Alternatively, natural selection may favor agents that coarse-grain the world in a way that allows them to exploit compositional representations.[12]

Our network classifier aims to output "1" if an image contains a cat and "0" otherwise. The Universal Approximation theorem[64] ensures that this classification function can be well approximated well by a sufficiently large neural network. Crucially, deep neural networks can learn this task more efficiently if the data is *compositional* in nature[65,66] and if the network architecture leverages this property. This efficiency generally arises when the generative model of the data is succinct—a short program in AIT terms[c].

The general feedforward network is defined by

$$x^{(l+1)} = h\left(W^{(l)}x^{(l)} + b^{(l)}\right), \quad l = 0, 1, \ldots, L-1 \tag{2.3}$$

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, h 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 $x = h(Wx + b)$, 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 and W is a block matrix describing the connections between the nodes, including the weights associated with each connection.

We require the classifier to be invariant under the action of the group G, whether it is finite or an infinite Lie group. Specifically, if the network classifies a point $\theta \in \mathcal{C}$ (via the image I_θ generated using parameters θ) 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 is sufficient to study the *finite* number of generators of the group—linear operators in the group's Lie algebra. An important property of Lie groups is that any element can be expressed in terms of actions near the identity element, although 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. The recursive nature of Lie groups ensures that to establish invariance or equivariance of a function under group action, it is sufficient to verify this property for the generators T_k. The behavior under infinitesimal transformations near the identity informs the behavior under the full group action. However, for disconnected Lie groups, invariance must be confirmed for each component separately.

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

$$x' = h(W(x + \delta x) + b) = h(W(x + \epsilon T x) + b), \tag{2.4}$$

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

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 preceding layers,

$$T_l = J_{l-1}W_{l-1} \cdot J_{l-2}W_{l-2} \cdot \ldots \cdot J_1 W_1 \cdot T_0 \tag{2.5}$$

[c] We can similarly analyze the case of a compressive autoencoder network. As a corollary, 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.

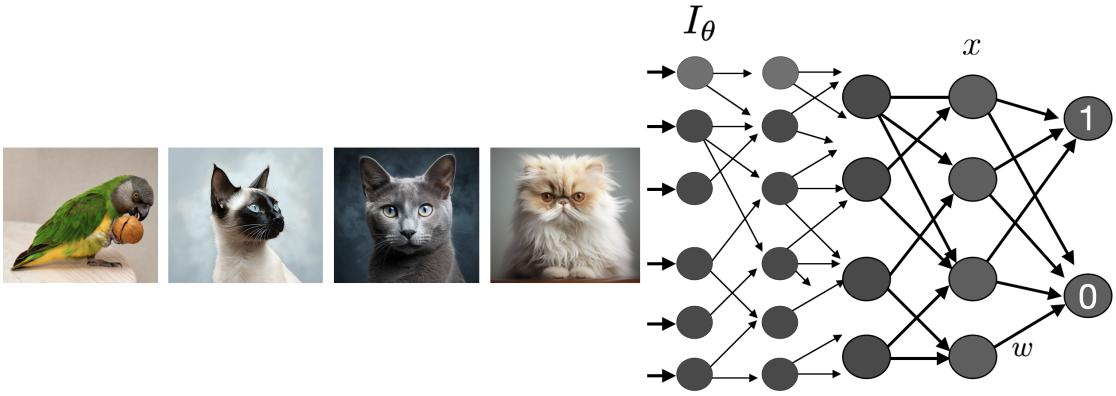

Figure 5: Image classification task. Classification of cat images can be seen as learning the invariances of a generative model. Such a network can also be implemented using an autoencoder of cat images with a skip connection to detect anomalies.

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 $I_\theta = \gamma_\theta \cdot I_0 \approx h((1 + \epsilon T_0)\theta_0)$.

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[56] for similar reasoning on the conditions imposed by symmetry). 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 propagates to higher layers, ensuring the network's output remains invariant to these transformations.

For the feedforward autoencoder, the analogous *equivariance* requirement is that the transformation of the output matches that of the input, namely,

$$T_N = J_{L-1} W_{L-1} \cdot J_{L-2} W_{L-2} \cdot \ldots \cdot J_1 W_1 \cdot T_0 = T_0 \tag{2.6}$$

when acting on cat images. This sets an M-dimensional family of constraints on W requiring that the matrix $T_N = J_{L-1} W_{L-1} \cdot J_{L-2} W_{L-2} \cdot \ldots \cdot J_1 W_1$ has the space of cat images as an eigenspace corresponding to the unit eigenvalue. These are the necessary requirements for invariance or equivariance under transformations.

Systems specifically designed for invariant behavior, such as convolutional networks with pooling layers, explicitly implement some of the required symmetries. Generalizing this idea involves designing networks that "filter" data across group orbits (e.g., such as translations or rotations), then analyze the results (e.g., computing the maximum or the average),[48] or using data augmentation techniques, or finding fundamental architectural principles for group invariance.[67] The success of hierarchical, recursive, deep architectures reflects this principle.[42,47]

In summary, requiring invariance of the classification of data produced by a generative 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—provided the generators are known.

Conversely, these principles can be useful for inferring symmetries underlying world data from the structure of networks trained on it.

3 The agent as a dynamical system

Although feedforward networks can be viewed as dynamical systems, their computational capacity is constrained by a fixed number of sequential evaluation steps. A broader perspective, bridging both algorithmic and biological paradigms, arises from the use of ordinary differential equations (ODEs), which have the potential to instantiate universal Turing machines.[68–70] Differential equations can be interpreted as recursive neural networks (RNNs), modeling continuous dynamics similarly to recursive processes in NNs. This analogy becomes particularly clear when the ODE equations are discretized.

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 mathematical details) 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*.

3.1 General model

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

$$\dot{x} = f\big(x; w, \eta(t)\big) \tag{3.1}$$

with $x \in \mathbb{R}^X$ is 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.[71] This finite-dimensional equation can be generalized to a partial differential equation (a neural field equation), but these 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 agent's state 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 received by agent sensors[d]. External inputs—which, from the perspective of a Universal Turing machine, can be viewed as reprogramming the agent—can force changes in the state trajectory or even drive the system's landscape through dramatic topological transitions.[37,72] For instance, such changes in topology can be induced by time-varying parameters associated with bifurcations in the system.

We call Equation 3.1 the "fast time" dynamics equation,[37] as it governs dynamics at short time scales (seconds), where connectivity parameters w are assumed to be constant (in reality, w may 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 parameters w.

3.2 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.

Since the only indeterminacy in the solutions to the ODEs are the initial conditions, we can also see symmetries as the action of a group on the initial conditions manifold (integration constants). It is, in fact, possible to reach any solution from any other through a group action. Again, this may be done globally with a potentially infinite-dimensional Lie group or locally with a finite-dimensional one.

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, with labeling within each class provided by the group element used.

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,[45,46,73] 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).[73] Furthermore, there is a deep connection between symmetries and conservation quantities, as we now describe.

A *conservation law* is a map from solutions 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.

[d] In the Appendix we provide a generalization of this model to include inputs and outputs reflecting interaction with World dynamics.

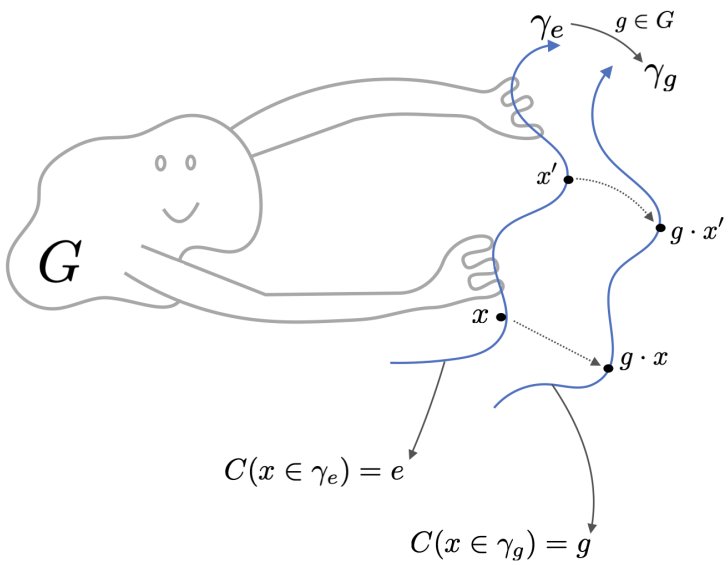

Figure 6: Group acting on solutions space and the (generalized) Noether's theorem. The action of a group on ODE trajectories can be used to provide a labeling system for them. This gives rise to conserved quantities (functions from phase space to group elements after the choice of some reference solutions—convention). [Figure inspired by Tom Leinster's Galois Theory.[74]]

If a system is invariant under the action of a group, we can transform one solution into another by applying a group element. Moreover, we can index (label) every solution using the group elements and a reference solution (which corresponds to the identity element of the group, $e \in G$, see Figure 6). Furthermore, the solution to the ODE corresponds to a set of points in phase space and under reasonable conditions (guaranteed by the Picard-Lindelöf Theorem, also known as the Cauchy-Lipschitz Theorem[75]), these solutions do not intersect (i.e., they do not share points) unless they are essentially the same solution after time translation (for autonomous ODEs). Therefore, all the points in a solution (the state trajectory) can be unambiguously mapped to a group element (except for the group elements corresponding to time translation in autonomous ODEs). This mapping defines a function from phase space to group elements and initial condition classes, and this function remains constant along the entire solution. This constant information represents the conserved quantity associated with the system's symmetry.

Thus, we can think of this process as the act of creating a *structured labeling system* for the solution space and then phase space, using the group. *The group labels, as the output of functions from phase space to group elements, are the conserved quantities.*

To illustrate these ideas, consider the equation $\dot{x} = 0$ with $x \in \mathbb{R}$. It displays translational symmetry: if $x(t)$ is a solution, then $x(t) \to \tilde{x}(t) = x(t) + \delta x$ is also a solution. The group's action is to translate solutions along the x-axis (phase space is 1D in this example). The conserved quantity is $C(x) = x = x_0$, which is shifted by the action of the group, $C \to C + \delta x$. We provide more simple examples of symmetry transformations that shift associated conserved quantities in Appendix A.7.

Finally, a well-known example in classical mechanics is the Kepler problem (two-body problem in celestial mechanics), an autonomous ODE in six-dimensional phase space, where 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).[76]

More generally, N-dimensional ODEs have $N-1$ conserved quantities (essentially functions of the constants of the motion, see Appendix A.7). An ODE with constraints (an algebraic differential equation) has solutions 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 and act as (group) labels of solutions. The symmetry of the equation is associated with transformations that shift these constants.

4 The world-tracking condition

Because of the time-dependent input term, it seems difficult to say very much about the behavior of the fast-time equation (3.1). However, there are two important elements of the model that we can leverage. Namely, we assume that

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 now analyze the mathematical consequences of these statements.

First, the assumption that world data is the product of a simple generative model—as discussed in the previous sections—implies that the constrained dynamics of the agent can be studied from the point of view of symmetry. We may think of the external world as providing stochastic but structured input to the dynamic system.

Second, 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 task is carried out by the *Comparator* in the KT agent model, see Figure 2).

To make this comparison meaningful, there must be a readout from the agent's internal state that matches the tracked world data. This necessity imposes a dynamical constraint on the system: the agent's internal states must evolve so that their readouts correspond to the external inputs. This constraint ensures that the agent's internal dynamics are aligned with the environment, enabling accurate tracking and adaptation.

As we will see, the implications of world-tracking are, in fact, twofold. On the one hand, the dynamics of the system are constrained to lie in reduced manifolds. On the

other, the architecture of the equations implementing world tracking needs to satisfy some requirements, as we already saw above in the simpler case of a feedforward classifier.

4.1 Constrained dynamics I

We begin with a simple toy model to study the role of symmetry and conservation laws. Consider the constrained system of differential equations (or *differential algebraic equation*[77]),

$$\begin{aligned} \dot{x} &= 0 \\ \dot{y} &= 1 \\ x &= C \end{aligned} \tag{4.1}$$

The first two equations are regular ODEs, and the last is a constraint (C is an arbitrary parameter), a toy world-tracking condition. These equations happen to be solvable for any value of C, because the general solution to the ODE subsystem is

$$\begin{aligned} x &= A \\ y &= t + B \end{aligned} \tag{4.2}$$

Thus, it is possible to select the *integration constants* A and B that satisfy the constraint. This occurs because the constraint is not independent of the conserved quantities of the system. The same conclusion follows if the third equation in 4.1 is 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.

In fact, Equation 4.1 has two conserved quantities (corresponding to two constants of the motion, or integration constants) and two symmetries associated with two generators[e] associated with them,

$$\begin{aligned} \Phi_1 = x, \quad & x \to x + \epsilon,\ y \to y, \quad T_1 = \partial/\partial x \\ \Phi_2 = y - xt, \quad & x \to x,\ y \to y + \epsilon, \quad T_2 = \partial/\partial y \end{aligned} \tag{4.3}$$

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. Further simple examples are provided in Appendix A.7.

[e]The operator notation, e.g., $T_1 = \partial/\partial x$ corresponds to the generator of a local Lie group, since

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

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

4.2 Constrained dynamics II

Consider next an agent watching a moving human hand generated by some simple physics-based computational model, as we described in[8]—see Figure 8.

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[78]) 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,[79] 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 1.6×10^{10} neurons and about 1.5×10^{14} synapses in the human cerebral cortex[80,81]). The dimensions of the brain, pixel, and model space are drastically different. There are many more neurons (or synapses) 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 or the algorithmic pattern of "hand."

We return to Equation 3.1, which becomes $\dot{x} = f\big(x; w, I_{\theta(t)}\big)$, where we have expressed the model inputs as $\eta(t) = I_{\theta(t)}$ (the image may be preprocessed in some way before its input to the network—e.g., 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 θ 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 agent's task, 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)*

$$\begin{aligned} \dot{x} &= f\big(x; w, I_{\theta(t)}\big) \\ g(x) &\approx I_{\theta(t)} \end{aligned} \tag{4.4}$$

with the inputs $I_{\theta(t)}$ produced by the generative model and some parameter dynamics (see Figure 7).

This a *differential algebraic equation* (it consists both of 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.

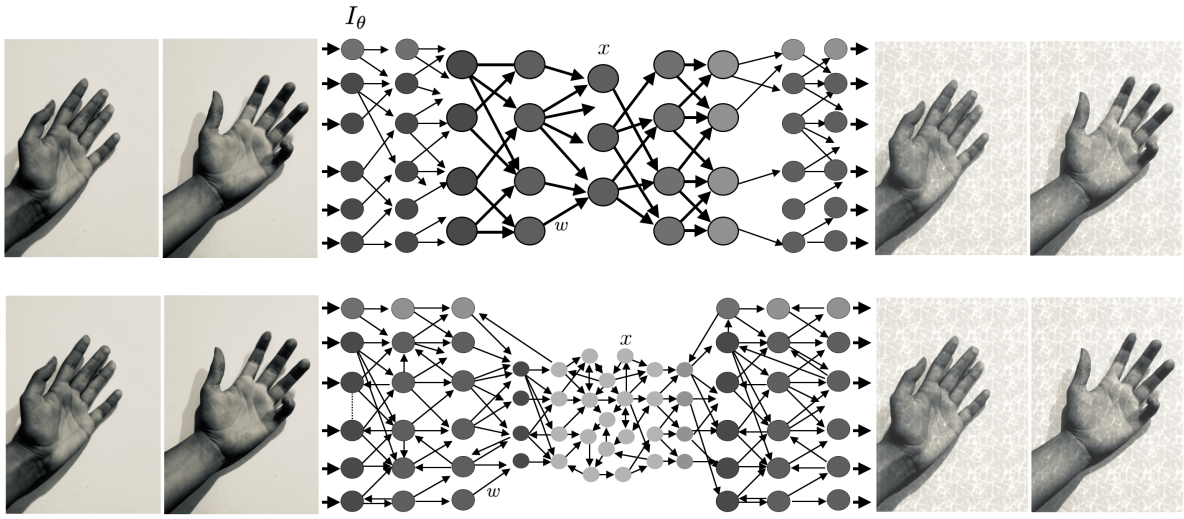

Figure 7: World-tracking neurodynamics. Tracking the world, represented here by a set of frames of a moving hand, is essentially the task of a compressive autoencoder, which can be described by Equation 4.4. The top panel displays a feedforward autoencoder, while the bottom provides a more general recurrent (RNN) 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. The RNN displayed is generic, e.g., the role of the Comparator is not made explicit.

In fact, even in the case of a static input, the only way to satisfy the constraint is if the constraint is a conserved quantity in disguise,

$$\frac{d}{dt}(g(x) - I_\theta) = 0 \tag{4.5}$$

We may relax the problem by asking for approximate tracking, as indicated by the "≈" symbol in the second equation. This may be implemented by replacing the constraint with an equation of the form $||I_{\theta(t)} - g(x)||^2 < \epsilon$. We may also relax the problem by asking for approximate tracking after transient behavior (the constraint is then seen as a Lyapunov function of the ODE, see 4.5).

Another way to view Equations 4.4 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 7. The network implementation is recurrent and reminiscent of architectures in Reservoir Computing,[82–84] but it can also represent a feedforward system such as a variational autoencoder[85]—or something in between, as in Figure 7. In this machine-learning context, we rewrite the WTNE autoencoder equation as

$$\begin{aligned} \dot{x} &= f\big(x; w, I_{\theta(t)}\big) \\ w &= \arg\min_w ||g(x(t; w)) - I_{\theta(t)}||^2 \end{aligned} \tag{4.6}$$

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 4.4 rather than Equation 4.6 for the sake of simplicity, although we should keep in mind that, in a real scenario, only approximate solutions will be possible.

4.3 The fixed input problem

The projection operator g in Equations 4.4 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

$$g(x) = I_\theta \tag{4.7}$$

This implies that the dynamics are characterized by a large set of conserved quantities $g_n(x) = C_n$, $n = 1, ..., Y$ (which may not all be independent, of course). Since g is a variable selector, this 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 $g(x) = C$.

The existence of conserved quantities (constraints) implies the ODE has symmetry associated with them (see Appendix A.7). 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 4.4 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 (local) 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.

Finally, Equation 4.7 has to hold for any value of θ. Since this constraint is associated with the dynamics, $\dot{x} = f(x; w, I_\theta)$, this imposes conditions on the w structure of the equations.

4.4 Covariance of the equations

To analyze the general case in more detail, we now split the variables x into two blocks, $x = (z, u)$, where u are the variables involved in the world-tracking constraint (see Figure 8). 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 explicit reference to u as

$$\begin{aligned} \dot{z} &= f_z\big(z; w, I_\theta\big) \\ I_\theta &\approx u = \sigma(w_u z) \end{aligned} \tag{4.8}$$

where for simplicity, we assume a feedforward read-out layer with a pointwise nonlinearity, with $g(x) = \sigma(w_u z) = u$ (recall that g is a variable selector operator selecting u, a small 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, given the input I_θ, z is a

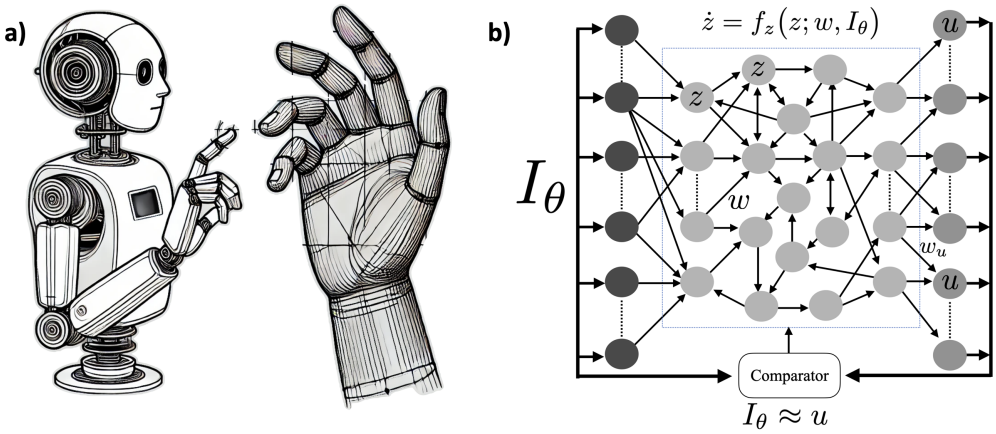

Figure 8: Agent tracking a computer-generated moving hand. a) An agent is observing and tracking a moving human hand generated by some simple physics-based computational model (such as Blender[79]). b) World tracking neural network, where the input is compared to the read out—see Equation 4.4. We don't detail here how the Comparator is implemented in the network or how its output is used to regulate the network—see Section 4.5 for a more in-depth discussion.

solution, then so is $\gamma \cdot z$ with input $\gamma \cdot I_\theta$, that is,

$$\begin{aligned} \frac{d}{dt}(\gamma \cdot z) &= f_z\big(\gamma \cdot z; w, \gamma \cdot I_\theta\big) \\ \gamma \cdot I_0 &\approx \sigma(w_u \gamma \cdot z) \end{aligned} \tag{4.9}$$

This is simply a consequence of the input tracking constraints and the Lie-generative nature of the world model. The agent is able, by definition, to track any version of the hand images (recognizing them as "hands").

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[51] for a more formal mathematical discussion.). These operators commute with time differentiation ($T_k(z)$ act on configuration space), hence

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

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 ϵ is arbitrary and the generators linearly independent.

Covariance under the Lie group transformations, Equation 4.10, imposes specific conditions on the weights w. These conditions ensure that the dynamics $\dot{z} = f_z(z; w, I_\theta)$ are consistent with the symmetry properties of the system and the desired conservation laws (world tracking constraints).

As a consequence of the world tracking constraints associated with inputs from a generative model, we see that the number of conservation laws or constraints is proportional to the number of generators of the group. This is because each generator of the Lie group introduces constraints through its action on the output variables u. A group with a large

number of generators will impose tighter constraints (and more conservation laws) on the possible w space.

4.5 Tracking time-varying inputs; the Comparator

Here, we consider the situation where the inputs change in time but at a long time scale compared to neural dynamics to allow for transient phenomena to complete. As the input image moves slowly through changes in the M-dimensional parameters θ 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 θ 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_θ 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 in Section A.7.3, 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.

So far, we have not discussed *how* the agent system can track changing inputs. To formalize this, we can provide more details on the world-tracking equations to represent the role of the Comparator in the agent, which computes the error between the model and the data (see Figure 2) and feeds it back to adjust the dynamics so that the constraint is met after a transient period. Recall the general *world-tracking neurodynamics equations (WTNE)*, Equation 4.4,

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

with the inputs $I_{\theta(t)}$ produced by the generative model and some parameter dynamics (see Figure 7). We can better describe the dynamics and the approximate constraint using the notion of the Lyapunov function. Lyapunov stability is a method for analyzing the behavior of dynamical systems near an equilibrium. It uses a Lyapunov function, which acts like an "energy" measure. If this function is non-increasing over time, the system is stable: small disturbances won't cause it to diverge far from equilibrium. When the Lyapunov function strictly decreases, the system converges to a stable point or a minimum. However, if the function remains constant along certain trajectories, the system may not settle at a point but instead exhibit periodic or orbiting behavior.

Using this, we define the dynamic world-tracking problem:

The world-tracking agent dynamics satisfy the equations

$$\dot{x} = f(x; w, I_{\theta(t)}) \tag{4.11}$$
$$\dot{V} \leq 0 \tag{4.12}$$

where $V(t)$ is a Lyapunov function capturing the role of the Comparator,

$$V = \frac{1}{2}E^T T \tag{4.13}$$
$$E(t) = g(x) - I_{\theta(t)} \tag{4.14}$$

Here x represents the state variables of the agent system, w denotes the system parameters or weights, $I_{\theta(t)}$ is the Lie-generated time-varying input from the environment, parameterized by $\theta(t)$, $g(x)$ is the model's output (readout) based on the current state x, and $E(t)$ is the error computed by the Comparator. The goal is to achieve world-tracking after transient behavior.

As usual, these equations are to be equivariant under the action of the Lie group. The Lie group control structure of time-varying input plays a key role in making the problem feasible.

We can provide a more detailed form of the tracking equation assuming that only the error is used by the network using proportional feedback,[86]

$$\dot{x} = h\big(x; w\big) + \hat{K}E(t) \tag{4.15}$$

where $\hat{K}$ is a linear feedback gain operator that determines the influence of the error on the system's dynamics.

The time derivative of V is $\dot{V} = E^T\dot{E}$. Substituting the dynamics of $\dot{E}$ from Equation (4.15):

$$\dot{E} = \frac{\partial g}{\partial x}\Big[h\big(x; w\big) + \hat{K}E(t)\Big] - \frac{dI_{\theta(t)}}{dt}$$

To ensure $\dot{V} \leq 0$, the feedback gain $\hat{K}$ must be designed such that

$$E^T\left(\frac{\partial g}{\partial x}\Big[h\big(x; w, I_{\theta(t)}\big) + \hat{K}E(t)\Big] - \frac{dI_{\theta(t)}}{dt}\right) \leq 0$$

Proper selection of $\hat{K}$ ensures that $V(E)$ is non-increasing, thereby driving $E(t)$ towards zero asymptotically. The choice of these matrices will reflect the Lie group structure of the system. Indeed, the feasibility of finding a solution to this problem with the right K matrices is likely to require symmetry. We leave this analysis for further work.

By integrating a feedback mechanism into the world-tracking equations, we enable the agent system to adjust dynamically and minimize the error $E(t)$ between the model's output and the environmental input. This formalization ensures the agent can effectively

track and adapt to changing inputs, maintaining alignment between the model and the observed data.

Finally, these equations correspond to the fast-time equations in Ruffini et al., 2024.[37] Comparator errors $E(t)$ will feed the slow-time equations (*connectodynamics*[37]) to govern the slow change of the structural parameters w, $\dot{w} = W(w, x, E(t))$, with $\dot{w} = W(w, x, 0) = 0$.

4.6 Coarse-graining, hierarchical constraints, and manifolds

A central idea in the agent framework is that all cognitive processes—modeling, planning, and objective function evaluation—are inherently hierarchical,[8] forming a layered structure of decision-making and goal-setting affecting models, objectives, and plans, creating a continuous hierarchy from simple actions to complex behaviors. This hierarchical organization naturally finds its origin in the multi-scale structure of real-world data and the processing required for effective modeling and decision-making. To ensure survival, and due to its limited resources, the agent must employ coarse-graining to transform data that initially appears uncompressible into a compressible form while retaining a non-trivial structure.[12] Coarse-graining may include various forms of data reorganization, such as spatiotemporal averaging, spectral decomposition, compressive sensing, and dimensionality reduction techniques.

This perspective aligns with the view of the brain processing information through *hierarchical coarse-graining*, in line with research findings in the visual[87,88] and auditory systems.[89] The same approach is employed by convolutional neural networks.[47,52]

At each processing level, details are aggregated and abstracted to form higher-level representations. World-tracking at different levels corresponds to different scales of coarse-graining, progressively reducing the complexity of the system while preserving essential features.

Since lower-level constraints (e.g., "it's furry") must be compatible with higher-level constraints ("it's a cat"), we can construct a nested structure of constraints and corresponding manifolds. Following this logic, world-tracking is achieved by imposing a hierarchy of constraints $\{\mathcal{C}_i\}_{i=1}^k$, each corresponding to a specific coarse-graining level. The constraints are nested and compatible, meaning each lower-level constraint operates within the subset of the state space defined by the higher-level constraints.

We can formalize this concept in a simplified manner (reality is likely more complex, involving parallel threads of constraints, for example) through a sequence of *coarse-graining operators* $\{\mathcal{G}_i\}_{i=1}^k$, where each operator $\mathcal{G}_i : \mathbb{R}^n \to \mathbb{R}^{n_i}$ maps the fine-grained state vector $\mathbf{x}$ to a coarse-grained state vector $\mathbf{y}_i$,

$$\mathbf{y}_i = \mathcal{G}_i(\mathbf{x}), \quad \text{with} \quad n_i < n. \tag{4.16}$$

Each $\mathbf{y}_i$ represents the system at a specific level of abstraction, capturing variables relevant at that scale. For each level i, we impose a *hierarchical constraint* $\mathcal{C}_i$ on $\mathbf{y}_i$,

$$\mathcal{C}_i(\mathbf{y}_i) = 0. \tag{4.17}$$

These constraints reflect the agent's world model at different levels, ensuring that the dynamics of the coarse-grained variables align with expected behaviors.

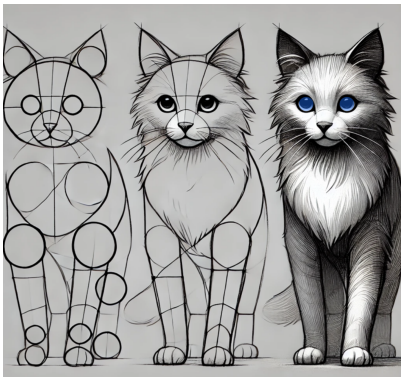

Figure 9: Hierarchical modeling: step-by-step drawing of a cat demonstrating hierarchical constraints: starting with the basic form of a cat ("it's a cat"), refining it with more specific features, and finally adding details like "has white fur and blue eyes," progressively narrowing the state space to match more specific characteristics.

Lower-level constraints must be *compatible* with higher-level constraints. This means that the solution set of a lower-level constraint $\mathcal{C}_{i+1}$ is a subset of the solution set of the higher-level constraint $\mathcal{C}_i$,

$$\{\mathbf{x} \in \mathbb{R}^n \mid \mathcal{C}_{i+1}(\mathcal{G}_{i+1}(\mathbf{x})) = 0\} \subseteq \{\mathbf{x} \in \mathbb{R}^n \mid \mathcal{C}_i(\mathcal{G}_i(\mathbf{x})) = 0\} \tag{4.18}$$

This nesting ensures that as we move to lower levels (higher indices), the constraints become more specific, refining the state space within the context defined by higher-level constraints.

As an illustrative example, consider a visual perception task with a high-level constraint $\mathcal{C}_1$: the agent recognizes that the object is a *cat*—see Figure 9. This constraint reduces the state space to the manifold $\mathcal{M}_1$ of all possible cats, which is still high-dimensional. To this, we can add a lower-level constraint $\mathcal{C}_2$: the agent further discerns that the cat is *white with blue eyes*. This constraint reduces the state space to a submanifold $\mathcal{M}_2$ within $\mathcal{M}_1$, containing only white cats with blue eyes. Here, the lower-level constraint $\mathcal{C}_2$ is compatible with the higher-level constraint $\mathcal{C}_1$, as all white cats with blue eyes are indeed cats, $\mathcal{M}_2 \subseteq \mathcal{M}_1$. Thus, higher-level constraints define broad categories or contexts (e.g., recognizing an object as a cat), while lower-level constraints provide additional specificity (e.g., identifying the cat's color, hair type, etc). The nested structure ensures that all lower-level processing is coherent with the higher-level understanding, leading to efficient and accurate modeling of the world.

The hierarchical framework of constraints can be related to compositional Lie groups characterizing the invariant properties of symmetries of the equations. In our context, the hierarchical constraints ($\mathcal{C}_i$) on agent dynamics correspond to conserved quantities associated with symmetries of the ODEs represented by a hierarchical Lie group structure.

The structure of nested, compatible constraints has several important implications. First, the state space is progressively reduced at each level, focusing the agent's modeling and computation on increasingly specific subsets of the environment. Second, compatibility ensures that lower-level details enhance rather than conflict with higher-level understanding, leading to coherent perception and action. Third, by operating within the context defined by higher-level constraints, lower-level processing can be more targeted and effi-

cient. Fourth, the nested constraints facilitate the emergence of complex behaviors from the interaction of simple rules applied at different scales. Finally, the agent can flexibly adjust its level of abstraction depending on the task requirements, adding or relaxing constraints as needed.

By imposing these hierarchical, compatible constraints, we restrict the original high-dimensional phase space $\mathbb{R}^n$ to a sequence of *nested manifolds*, forming what we may call a *hierarchical manifold*,

$$\mathbb{R}^n = \mathcal{M}_0 \xrightarrow{\mathcal{C}_1} \mathcal{M}_1 \xrightarrow{\mathcal{C}_2} \mathcal{M}_2 \xrightarrow{\cdots} \mathcal{M}_k, \tag{4.19}$$

where each manifold is defined recursively as

$$\mathcal{M}_i = \{\mathbf{x} \in \mathcal{M}_{i-1} \mid \mathcal{C}_i(\mathcal{G}_i(\mathbf{x})) = 0\}, \quad \text{for } i = 1, 2, \ldots, k, \tag{4.20}$$

with $\mathcal{M}_0 = \mathbb{R}^n$, or equivalently as

$$\mathcal{M}_i = \{\mathbf{y}_{i-1} \in \mathcal{M}_{i-1} \mid \mathcal{C}_i(\mathcal{G}_i(\mathbf{y}_{i-1})) = 0\} \subseteq \{\mathbf{y}_{i-1} \in \mathcal{M}_{i-1}\} \tag{4.21}$$

Each manifold $\mathcal{M}_i$ satisfies all constraints up to level i, ensuring that the agent's internal states remain consistent with both high-level goals and fine-grained observations. This recursive structure ensures that $\mathcal{M}_i \subseteq \mathcal{M}_{i-1}$,, reflecting the increased specificity of constraints at lower levels.

The sequence of hierarchical, compatible constraints leads, in turn, to a sequence of nested manifolds, each representing the state space under the accumulated constraints up to that level. The agent's dynamics are thus confined to a reduced hierarchical manifold, ensuring that its internal states remain consistent with both high-level goals and fine-grained observations. This mirrors the hierarchical organization of the brain or deep artificial networks, where a nested structure of coarse-grained models synthesizes a multilevel emergent view of the world.

In summary, we assumed that the agent inputs are generated by simple rules (i.e., that the world is inherently simple in algorithmic terms), i.e., by a hierarchical Lie generative model, and that the agent can track its inputs, i.e., encodes an internal generative model to approximately match data, perhaps after transients. As a consequence,

1. As a dynamical system, the successful agent displays conserved quantities—with dynamics in a reduced hierarchical manifold—corresponding to the world-tracking constraints generated by a Lie group. For dynamical inputs, this occurs after a transient period and is captured by a Lyapunov function (see Section 4.5).
2. The constraints force *structural* symmetries in the dynamical system constitutive equations that meet them: the agent carries the symmetries of the world generative model. This is encoded in the structural elements of the dynamical system (the ws in our formulation).

Finally, constraint (symmetry) breaking, as detected by the agent in the Comparator (Figure 2), allows the agent to identify anomalies or novel patterns, prompting updates to its internal models and engaging in learning and adaptation.

5 Discussion

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 specialized, symmetric networks, as the governing equations must maintain (approximately) conserved quantities that are compatible with the required outputs.

If the inputs are not varying, dynamics will reside within a reduced manifold of minimal dimension, determined by the constraints of world tracking. As the inputs begin to vary—following the rules of a generative world model—the dimension of the reduced manifold increases, but only slightly compared to the embedding neural dimension. Conversely, if inputs violate the world model, the dynamics will no longer adhere to the world-tracking constraints, tracking will fail, and the dynamics may no longer be constrained.

Our analysis thus supports the *manifold hypothesis*. 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. The manifold hypothesis posits that high-dimensional data, such as world data collected from sensory systems (e.g., images) or neuroimaging, can be compressed into a reduced number of parameters (latent space[90]) due to the presence of a low-dimensional manifold within the high-dimensional phase space.[42,44,91,92] Such manifolds can take various forms, such as attractors and unstable manifolds.

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 the current paper, we have studied the implications of the world-tracking condition with a focus on static or slowly varying inputs. We have not discussed *how* the agent can track time-varying inputs, where the input dynamics are controlled through the Lie group structure. We formalize this problem, which belongs to the branch of mathematics of Control Theory, in the Appendix.

Finally, we expect reduced manifolds to reflect the compositional nature of World data. This has led us to the definition of hierarchical manifold.

> A reduced hierarchical data manifold embodies the inherent simplicity of the underlying generative models that produce the observed data, as well as the effectiveness of a world-tracking system in learning these models. This observation supports the **manifold hypothesis**.

5.1 Connections with empirical observations

Symmetry is an example of an algorithmic feature that systems that compute the same function must share, regardless of their specific implementation. This implies that natural agents such as human brains implement symmetries inherent in world data within 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. Techniques like topological data analysis may help uncover further correlates of structured experience with applications in consciousness research in natural and artificial systems.

The fundamental predictions from studying the role of symmetry in agent dynamics pertain to the structural constraints of its constitutive equations and to the emergence of dynamical reduced manifolds, reflecting the structure of compositional data the agent is tracking.

The simplicity associated with a generative model may manifest in different ways. For instance, an efficient autoencoder of neural data from a brain exposed to time-varying world inputs could regenerate the dynamics using a latent parameter space whose dimension approximately matches the dimension of the world-generative model configuration space.

Furthermore, the Comparator mechanism establishes the emergence of Mutual Algorithmic Information between the agent and the world, which KT postulates.[7,8] By aligning internal models with external data, the Comparator highlights how constraints in world-tracking impact the structure of reduced, invariant manifolds in neural dynamics. These manifold structures, in turn, are expected to vary with brain states and environmental conditions, influencing neuroimaging data across different states of consciousness (awake, REM, NREM),[90] under anesthesia, minimally conscious (MCS) or unresponsive wakeful state (UWS), Locked-In Syndrome (LIS), epilepsy or Alzheimer's disease (AD),[93] 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 and motivates the hypothesis that structured experience is associated with the successful match of data and model at the Comparator.[7–9]

Constrained dynamics may also provide the context for phenomena such as increased EEG slow-frequency power in the eyes-closed state (and reduced complexity[94]), anesthesia, or completely locked-in syndrome (CLIS, where patients shift toward slower frequencies and lower signal complexity),[95,96] all of which are characterized by clamping down (setting to a null fixed value) world inputs. This may either reflect suppression of Comparator outputs or the active tracking of a predicted static input. On the other extreme, the perspective of constrained dynamics may explain how psychedelics, which interfere with sensory processing, disrupt the Comparator and modeling process (Equations 4.11–4.14), with network dynamics departing from the normative invariant manifold. Such a departure manifests as an increase in complexity and reduction in alpha power.[27,28,97–102]

For data analysis, approaches such as deep learning autoencoders are well-suited for uncovering latent spaces associated with brain dynamics. Complementing this, topological data analysis can reveal structural features of reduced manifolds. These techniques could provide new insights into how different states compress or expand neural dynamics, providing insights into underlying brain symmetries and their breakdown in pathology. Complexity measures, like the Lempel-Ziv algorithm, can help quantify the intrinsic dimensionality of these manifolds, revealing symmetry-related information processing characteristics.[11,103]

Finally, while we have focused here on how compositional data-tracking gives rise to constrained dynamics, the same logic applies to the generation of *compositional outputs*

by the agent, such as motor actions. Recent research has uncovered low-dimensional neural manifolds that capture significant neural variability across various brain regions and experimental conditions.[104–113] These manifolds, defined by correlated neural activity patterns of neural activity —referred to as "neural modes,"— are proposed to generate motor behavior through their time-dependent activation.[104,108]

This research will benefit from a similar analysis of artificial systems, as discussed next.

5.2 Applications in AI and neurosynthetic agent design

In *artificial intelligence* (AI) and *computational neuroscience*, symmetry constraints are pivotal in guiding model design by imposing structural limitations that reflect the symmetries inherent in World interaction tasks. These symmetry principles can be incorporated as regularization terms to reduce parameter degeneracy or structurally (e.g., convolutional networks), enhancing model robustness and efficiency. From the earliest developments of deep networks[47,52,53,114] to current advancements, symmetry-driven approaches continue to shape and fuel research into more adaptable and efficient models.[48,50,56,61,115]

Whole-brain computational models and digital twins[116] are *neurosynthetic* neural networks characterized by a very large number of parameters that are difficult to adjust given the limited available neuroimaging data. Just as with artificial neural networks, enforcing symmetry and conserved quantities with low dimensionality, we can reduce the parameter space's degeneracy when fitting models to data.[117] Additionally, when parameters are known, the system's dynamics can be explored through its symmetry properties. Analyzing the number, dimension, structure, and stability of the system's attractors (using eigenvalue and bifurcation analysis) provides insight into its long-term behavior[1,73] and computational capabilities, including multistability, metastability, and criticality.[19,25,118,119]

For instance, whole-brain models need to reflect the spatial-temporal symmetries and sensory-motor contingencies inherent in the data they process.[120] This includes fundamental transformations like translations and rotations. Such considerations will become increasingly important as whole-brain models are developed to interact with external inputs and generate outputs, whether in classification tasks or motor actions.

An important part of this program is to characterize more general world symmetries 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.[49,51,56] The discovery of these attractors can be used to explore the mutual algorithmic information between data and dynamical attractors in the artificial system (or in the human brain).

Finally, symmetry considerations can serve as a regularization principle, instrumentalizing a bias towards simplicity. By demanding that models remain as symmetric as possible while fitting empirical data, we implement a form of Occam's razor, promoting simpler, more generalizable models. This principle is particularly relevant for brain-inspired networks, where preserving computational simplicity can improve interpretability and robustness.[1]

6 Conclusion

In past work, we put forth the idea that because agents possess and run models of the World, there is significant Mutual Algorithmic Information (MAI) between the agent and the world.[7] In this paper, this idea is made concrete in dynamical terms using the language of group theory: the agent, viewed as a dynamical system, encodes symmetries that reflect the structure of world data.

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 initial development by Évariste Galois and Paolo Ruffini[121–123]—who, to unveil the algebraic symmetries governing polynomial equations, created the seeds of group theory—, to the works of Sophus Lie,[1,73] who extended these concepts into the role of continuous groups in differential equations, group theory is rich in its fruits. The subsequent epoch, marked by Emmy Noether's transformative theorem,[45,46] cemented the profound connection between symmetries and conservation laws, further elucidating the inherent simplicity underlying complex natural phenomena and driving the development of General Relativity and the Standard Model in physics.

These historical milestones underscore the enduring quest for understanding through the lens of symmetry, to which we hope this paper contributes. As we have discussed here through the model of the algorithmic agent (KT), the convergent principles of algorithmic simplicity and symmetry can yield useful insights into the nature of artificial and natural intelligence and the deep connections between mathematics and the phenomenology of structured experience.

References

[1] Peter J Olver. *Applications of lie groups to differential equations*. Graduate texts in mathematics. Springer, New York, NY, 1986 edition, December 2012.

[2] Peter J. Olver. Symmetry and explicit solutions of partial differential equations. *Applied Numerical Mathematics*, 10(3):307–324, September 1992.

[3] Thomas Metzinger. Artificial suffering: An argument for a global moratorium on synthetic phenomenology. *Journal of Artificial Intelligence and Consciousness*, 08(01):43–66, February 2021. Publisher: World Scientific Pub Co Pte Lt.

[4] Ethics of Artificial Intelligence | Internet Encyclopedia of Philosophy, 2023. ISSN: 2161-0002 tex.howpublished: hhttps://iep.utm.edu/ethics-of-artificial-intelligence/.

[5] Giulio Ruffini. Information, complexity, brains and reality ("Kolmogorov Manifesto"). *arXiv*, 2007. tex.date-added: 2009-03-01 11:09:48 +0100 tex.date-modified: 2009-06-06 18:41:44 +0200.

[6] Giulio Ruffini. Reality as Simplicity, June 2009. arXiv:0903.1193 [physics].

[7] Giulio Ruffini. An algorithmic information theory of consciousness. *Neuroscience of Consciousness*, 2017(1):nix019, 2017.

[8] Giulio Ruffini and Edmundo Lopez-Sola. AIT foundations of structured experience. *Journal of Artificial Intelligence and Consciousness*, 9(2):153–191, September 2022. Publisher: World Scientific Pub Co Pte Ltd.

[9] Giulio Ruffini, Francesca Castaldo, Edmundo Lopez-Sola, Roser Sanchez-Todo, and Jakub Vohryzek. The algorithmic agent perspective and computational neuropsychiatry: from etiology to advanced therapy in major depressive disorder. *PsyArXiv*, March 2024.

[10] Ming Li and Paul M.B. Vitanyi. Applications of algorithmic information theory. *Scholarpedia*, 2(5):2658, 2007. tex.date-added: 2016-10-15 14:12:39 +0000 tex.date-modified: 2016-10-22 13:13:18 +0000.

[11] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & sons, 2 edition, 2006. tex.date-added: 2016-10-20 21:00:29 +0000 tex.date-modified: 2016-10-20 21:00:58 +0000.

[12] Giulio Ruffini. Navigating Complexity: How Resource-Limited Agents Derive Probability and Generate Emergence, September 2024.

[13] Richard P. Feynman. *Feynman lectures on computation*. Advanced book program. Perseus, Cambridge, Mass, paperback print., [repr.] edition, 2001.

[14] Chris G. Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1-3):12–37, June 1990.

[15] Woodrow L Shew and Dietmar Plenz. The functional benefits of criticality in the cortex. *The Neuroscientist : a review journal bringing neurobiology, neurology and psychiatry*, 19(1):88–100, February 2013. Publisher: SAGE Publications.

[16] E Erten, Joseph Lizier, Mahendra Piraveenan, and Mikhail Prokopenko. Criticality and information dynamics in epidemiological models. *Entropy (Basel, Switzerland)*, 19(5):194, April 2017. Publisher: MDPI AG.

[17] Mike Li, Yinuo Han, Matthew J Aburn, Michael Breakspear, Russell A Poldrack, James M Shine, and Joseph T Lizier. Transitions in information processing dynamics at the whole-brain network level are driven by alterations in neural gain. *PLOS Computational Biology*, 15(10):1–27, 2019. Publisher: Public Library of Science.

[18] Pedro A M Mediano, Fernando E Rosas, Juan Carlos Farah, Murray Shanahan, Daniel Bor, and Adam B Barrett. Integrated information as a common signature of dynamical and information-processing complexity. *Chaos (Woodbury, N.Y.)*, 32(1):13115, January 2022. Publisher: AIP Publishing.

[19] G. Deco and V. K. Jirsa. Ongoing cortical activity at rest: Criticality, multistability, and ghost attractors. *Journal of Neuroscience*, 32(10):3366–3375, March 2012. Publisher: Society for Neuroscience.

[20] Gustavo Deco, Viktor Jirsa, A R McIntosh, Olaf Sporns, and Rolf Kötter. Key role of coupling, delay, and noise in resting brain fluctuations. *Proceedings of the National Academy of Sciences*, 106(25):10302–10307, 2009. Publisher: National Academy of Sciences tex.date-added: 2018-10-03 21:12:52 +0000 tex.date-modified: 2018-10-03 21:12:52 +0000.

[21] Filippo Cieri, Xiaowei Zhuang, Jessica Z. K. Caldwell, and Dietmar Cordes. Brain entropy during aging through a free energy principle approach. *Frontiers in Human Neuroscience*, 15:139, 2021. tex.date-added: 2021-11-18 09:45:28 +0100 tex.date-modified: 2021-11-18 09:45:38 +0100.

[22] Enzo Tagliazucchi, Pablo Balenzuela, Daniel Fraiman, and Dante R Chialvo. Criticality in large-scale brain FMRI dynamics unveiled by a novel point process analysis. *Frontiers in Physiology*, 3:15, February 2012. Publisher: Frontiers Media SA.

[23] Dante R Chialvo. Emergent complex neural dynamics. *Nature Physics*, 6(10):744–750, October 2010. Publisher: Springer Science and Business Media LLC.

[24] Ariel Haimovici, Enzo Tagliazucchi, Pablo Balenzuela, and Dante R Chialvo. Brain organization into resting state networks emerges at criticality on a model of the human connectome. *Physical Review Letters*, 110(17):178101, April 2013. Publisher: American Physical Society (APS).

[25] D Marinazzo, M Pellicoro, G Wu, L Angelini, J.M. Cortés, and S Stramaglia. Information transfer and criticality in the ising model on the human connectome. *PLoS ONE*, 9(4), 2014. tex.date-added: 2020-02-11 15:12:35 -0500 tex.date-modified: 2020-02-11 15:31:53 -0500.

[26] Fran Hancock, Fernando Rosas, Pedro Mediano, Andrea Luppi, Joana Cabral, Ottavia Dipasquale, and Federico Turkheim. May the 4c's be with you: An overview of complexity-inspired frameworks for analyzing resting-state neuroimaging data. Publication title: Preprints, March 2022.

[27] G. Ruffini and G. Deco. The 2D Ising model, criticality and AIT, October 2021. Pages: 2021.10.21.465265 Section: New Results.

[28] Giulio Ruffini, Giada Damiani, Diego Lozano-Soldevilla, Nikolas Deco, Fernando E Rosas, Narsis A Kiani, Adrián Ponce-Alvarez, Morten L Kringelbach, Robin Carhart-Harris, and Gustavo Deco. LSD-induced increase of Ising temperature and algorithmic complexity of brain dynamics. *Plos Computational Biology*, 19(2):e1010811, February 2023.

[29] R L Carhart-Harris and K J Friston. REBUS and the anarchic brain: Toward a unified model of the brain action of psychedelics. *Pharmacological Reviews*, 71(3):316–344, July 2019. Publisher: American Society for Pharmacology & Experimental Therapeutics (ASPET).

[30] Henrik Jeldtoft Jensen. *Self-organized criticality: Emergent complex behavior in physical and biological systems.* Cambridge lecture notes in physics. Cambridge University Press, 1998.

[31] Gunnar Pruessner. *Self-organised criticality.* Cambridge University Press, Cambridge, England, August 2012.

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

[33] Gina G Turrigiano and Sacha B Nelson. Homeostatic plasticity in the developing nervous system. *Nature reviews neuroscience*, 5(2):97–107, 2004. Publisher: Nature Publishing Group UK London.

[34] Dietmar Plenz, Tiago L Ribeiro, Stephanie R Miller, Patrick A Kells, Ali Vakili, and Elliott L Capek. Self-organized criticality in the brain. *Frontiers in Physics*, 9, 2021.

[35] Janina Hesse and Thilo Gross. Self-organized criticality as a fundamental property of neural systems. *Frontiers in Systems Neuroscience*, 8, 2014.

[36] RL Carhart-Harris. The entropic brain — revisited. *Neuropharmacology*, 142(167–178), 2018. tex.date-added: 2020-01-18 13:35:38 -0500 tex.date-modified: 2020-01-18 13:36:55 -0500.

[37] Giulio Ruffini, Edmundo Lopez-Sola, Jakub Vohryzek, and Roser Sanchez-Todo. Neural geometrodynamics, complexity, and plasticity: a psychedelics perspective. *Entropy*, 26(1):90, 2024. Publisher: MDPI.

[38] 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. *Communications on Pure and Applied Mathematics*, 13(1):1–14, 1960. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160130102.

[39] Ming Li and Paul Vitanyi. *An introduction to Kolmogorov Complexity and its applications*. Springer, 1997. tex.date-added: 2016-05-30 14:14:47 +0000 tex.date-modified: 2016-05-30 14:16:04 +0000.

[40] Giulio Ruffini. Models, networks and algorithmic complexity. *arxiv*, 2016. Publisher: arXiv.

[41] Robin L Carhart-Harris, Robert Leech, Peter J Hellyer, Murray Shanahan, Amanda Feilding, Enzo Tagliazucchi, Dante R Chialvo, and David Nutt. The entropic brain: a theory of conscious states informed by neuroimaging research with psychedelic drugs. *Frontiers in Human Neuroscience*, 8:20, February 2014. Publisher: Frontiers Media SA.

[42] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *arxiv*, 2012. Publisher: arXiv.

[43] Bradley CA Brown, Anthony L. Caterini, Brendan Leigh Ross, Jesse C Cresswell, and Gabriel Loaiza-Ganem. Verifying the union of manifolds hypothesis for image data. In *The eleventh international conference on learning representations*, 2023.

[44] James M Shine, Luke J Hearne, Michael Breakspear, Kai Hwang, Eli J Müller, Olaf Sporns, Russell A Poldrack, Jason B Mattingley, and Luca Cocchi. The low-dimensional neural architecture of cognitive complexity is related to activity in medial thalamic nuclei. *Neuron*, 104(5):849–855.e3, December 2019. Publisher: Elsevier BV.

[45] Emmy Noether. Invariante variationsprobleme. *Nachr. d. König. Gesellsch. d. Wiss. zuGöttingen, Math-phys. Klasse*, pages 235–257, 1918. tex.date-added: 2009-05-06 00:19:56 +0200 tex.date-modified: 2009-05-06 00:21:55 +0200.

[46] Dwight E Neuenschwander. *Emmy noether's wonderful theorem*. Johns Hopkins University Press, Baltimore, MD, April 2017.

[47] Tomaso Poggio, Jim Mutch, Joel Leibo, Lorenzo Rosasco, and Andrea Tacchetti. The computational magic of the ventral stream: sketch of a theory (and why some deep architectures work). Technical report, MIT Computer Science and Artificial Intelligence Laboratory, 2012. Issue: MIT-CSAIL-TR-2012-035.

[48] Fabio Anselmi and Tomaso Poggio. Representation learning in sensory cortex: A theory. *IEEE access : practical innovations, open solutions*, 10:102475–102491, 2022. Publisher: Institute of Electrical and Electronics Engineers (IEEE).

[49] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of Koopman eigenfunctions for control. *arxiv*, 2017. Publisher: arXiv.

[50] Eurika Kaiser, J Nathan Kutz, and Steven L Brunton. Discovering conservation laws from data for control. *arxiv*, 2018. Publisher: arXiv.

[51] Samuel E Otto, Nicholas Zolman, J Nathan Kutz, and Steven L Brunton. A unified framework to enforce, discover, and promote symmetry in machine learning. *arxiv*, 2023. Publisher: arXiv.

[52] K Fukushima. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980. Publisher: Springer Nature America, Inc.

[53] Y LeCun, B Boser, J S Denker, D Henderson, R E Howard, W Hubbard, and L D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, December 1989. Publisher: MIT Press - Journals.

[54] Rajesh Rao and Daniel Ruderman. Learning lie groups for invariant visual perception. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in neural information processing systems*, volume 11. MIT Press, 1998.

[55] Tomaso Poggio. The computational magic of the ventral stream. *Nature Precedings*, pages 1–1, March 2012. Publisher: Nature Publishing Group.

[56] Artem Moskalev, Anna Sepliarskaia, Ivan Sosnovik, and Arnold Smeulders. LieGG: Studying learned Lie group generators. *arXiv*, 2022. Publisher: arXiv.

[57] John Stillwell. *Naive lie theory*. Springer, 2008.

[58] Rudolf Schmid. Infinite-dimensional Lie groups and algebras in mathematical physics. *Advances in Mathematical Physics*, 2010:1–35, 2010. Publisher: Hindawi Limited.

[59] J M Selig. *Geometric fundamentals of robotics*. Monographs in computer science. Springer, New York, NY, 2 edition, November 2004.

[60] Blender Online Community. Blender - a 3D modelling and rendering package. manual, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[61] Xu Miao and Rajesh P N Rao. Learning the Lie groups of visual invariance. *Neural Computation*, 19(10):2665–2693, October 2007. Publisher: MIT Press - Journals.

[62] Mark Ibrahim, Diane Bouchacourt, and Ari S. Morcos. Robust self-supervised learning with lie groups, 2023.

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

[64] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks,*, 4(2):251–257, 1991. tex.date-added: 2016-11-02 19:04:43 +0000 tex.date-modified: 2016-11-02 19:05:32 +0000.

[65] T Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep – but not shallow –Networks avoid the curse of dimensionality: a review. *CBMM Memo*, (058), 2016. tex.date-added: 2016-12-04 00:19:36 +0000 tex.date-modified: 2016-12-04 00:21:03 +0000.

[66] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. Learning functions: When is deep better than shallow. Technical report, CBBM, 2016. Issue: CBMM Memo No. 045 tex.date-added: 2016-10-28 19:30:42 +0000 tex.date-modified: 2016-10-28 19:31:50 +0000.

[67] Devanshu Agrawal and James Ostrowski. A classification of G-invariant shallow neural networks. *arxiv*, 2022. Publisher: arXiv.

[68] C Moore. Generalized shifts: unpredictability and undecidability in dynamical systems. *Nonlinearity*, 4(2):199–230, May 1991. Publisher: IOP Publishing.

[69] Michael S Branicky. Universal computation and other capabilities of hybrid and continuous dynamical systems. *Theoretical Computer Science*, 138(1):67–100, February 1995. Publisher: Elsevier BV.

[70] Daniel Graça and Ning Zhong. Analytic one-dimensional maps and two-dimensional ordinary differential equations can robustly simulate Turing machines. *Computability. The Journal of the Association CiE*, pages 1–28, October 2022. Publisher: IOS Press.

[71] Zachary Friedenberger, Emerson Harkin, Katalin Tóth, and Richard Naud. Silences, spikes and bursts: Three-part knot of the neural code. *The Journal of Physiology*, October 2023.

[72] Gisela D Charó, Mickaël D Chekroun, Denisse Sciamarella, and Michael Ghil. Noise-driven topological changes in chaotic dynamics. *Chaos (Woodbury, N.Y.)*, 31(10):103115, October 2021. Publisher: AIP Publishing.

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

[74] Tom Leinster. Galois Theory, August 2024. arXiv:2408.07499 [math].

[75] Earl A. Coddington. *An introduction to ordinary differential equations*. Dover Books on mathematics. Dover Publ, New York, repr edition, 1989.

[76] D M Fradkin. Existence of the dynamic SymmetriesO4andSU3for all classical central potential problems. *Progress of Theoretical Physics*, 37(5):798–812, May 1967. Publisher: Oxford University Press (OUP).

[77] Stephen Campbell, Vu Linh, and Linda Petzold. Differential-algebraic equations. *Scholarpedia J.*, 3(8):2849, 2008. Publisher: Scholarpedia.

[78] Raul M Sanchez, Gary R Dunkelberger, and Harry A Quigley. The number and diameter distribution of axons in the monkey optic nerve. *Investigative Ophthalmology & Visual Science*, 27(9):1342–1350, 1986.

[79] Yuwei Li, Longwen Zhang, Zesong Qiu, Yingwenqi Jiang, Yuyao Zhang, Nianyi Li, Yuexin Ma, Lan Xu, and Jingyi Yu. NIMBLE: A non-rigid hand model with bones and muscles, 2022. tex.copyright: Creative Commons Attribution 4.0 International.

[80] Suzana Herculano-Houzel. The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost. *Proceedings of the National Academy of Sciences of the United States of America*, 109 Suppl 1(Suppl 1):10661–10668, June 2012.

[81] Bente Pakkenberg, Dorte Pelvig, Lisbeth Marner, Mads J. Bundgaard, Hans Jørgen G. Gundersen, Jens R. Nyengaard, and Lisbeth Regeur. Aging and the human neocortex. *Experimental Gerontology*, 38(1):95–99, January 2003.

[82] Herbert Jaeger. The "Echo State" approach to analysing and training recurrent neural networks. Technical report, German National Research Center for Information Technology GMD Technical Report, 2001. Issue: 148.

[83] Mantas Lukoševičius. A practical guide to applying echo state networks. *Neural Networks: Tricks of the Trade*, pages 659–686, 2012. Publisher: Springer.

[84] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoît Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hirokazu Numata, Daiju Nakano, and Akio Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, 2019. Publisher: Elsevier.

[85] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[86] Hassan K. Khalil. *Nonlinear systems. Hauptbd.* Prentice Hall, Upper Saddle River, NJ, 3. ed edition, 2002.

[87] James J. DiCarlo, Davide Zoccolan, and Nicole C. Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434, February 2012.

[88] Kalanit Grill-Spector and Kevin S. Weiner. The functional architecture of the ventral temporal cortex and its role in categorization. *Nature Reviews Neuroscience*, 15(8):536–548, August 2014. Publisher: Nature Publishing Group.

[89] Jennifer K. Bizley and Yale E. Cohen. The what, where and how of auditory-object perception. *Nature Reviews Neuroscience*, 14(10):693–707, October 2013. Publisher: Nature Publishing Group.

[90] Yonatan Sanz Perl, Hernán Bocaccio, Ignacio Pérez-Ipiña, Federico Zamberlán, Juan Piccinini, Helmut Laufs, Morten Kringelbach, Gustavo Deco, and Enzo Tagliazucchi. Generative embeddings of brain collective dynamics using variational autoencoders. *Physical Review Letters*, 125(23), December 2020. Publisher: American Physical Society (APS).

[91] Lawrence Cayton. Algorithms for manifold learning. Technical report, UC San Diego: Department of Computer Science & Engineering, 2008.

[92] Anne E Urai, Brent Doiron, Andrew M Leifer, and Anne K Churchland. Large-scale neural recordings call for new insights to link brain and behavior. *Nature Neuroscience*, 25(1):11–19, January 2022. Publisher: Springer Science and Business Media LLC.

[93] Yonatan Sanz Perl, Sebastian Geli, Eider Pérez-Ordoyo, Lou Zonca, Sebastian Idesis, Jakub Vohryzek, Viktor K. Jirsa, Morten L. Kringelbach, Enzo Tagliazucchi, and Gustavo Deco. Whole-brain modelling of low-dimensional manifold modes reveals organising principle of brain dynamics, November 2023. Pages: 2023.11.20.567824 Section: New Results.

[94] Nadine Farnes, Bjørn E Juel, André S Nilsen, Luis G Romundstad, and Johan F Storm. Increased signal diversity/complexity of spontaneous EEG, but not evoked EEG responses, in ketamine-induced psychedelic state in humans. *PLoS One*, 15(11):e0242056, November 2020. Publisher: Public Library of Science (PLoS).

[95] Khalili-Ardali M, Wu S, Tonin A, Birbaumer N, and Chaudhary U. Neurophysiological aspects of the completely locked-in syndrome in patients with advanced amyotrophic lateral sclerosis. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 132(5), May 2021. Publisher: Clin Neurophysiol.

[96] Federico Zilio, Javier Gomez-Pilar, Ujwal Chaudhary, Stuart Fogel, Tatiana Fomina, Matthis Synofzik, Ludger Schöls, Shumei Cao, Jun Zhang, Zirui Huang, Niels Birbaumer, and Georg Northoff. Altered brain dynamics index levels of arousal in complete locked-in syndrome. *Communications Biology*, 6(1):1–19, July 2023. Publisher: Nature Publishing Group.

[97] Robin L. Carhart-Harris. The entropic brain - revisited. *Neuropharmacology*, 2018. tex.date-added: 2021-01-25 10:56:48 +0100 tex.date-modified: 2021-01-25 10:56:48 +0100.

[98] T. F. Varley, M. Craig, R. Adapa, P. Finoia, G. Williams, J. Allanson, and E. A. Stamatakis. Fractal dimension of cortical functional connectivity networks & severity of disorders of consciousness. *PloS one*, 2020. tex.date-added: 2021-01-25 10:56:48 +0100 tex.date-modified: 2021-01-25 10:56:48 +0100.

[99] Andrea I. Luppi, Michael M. Craig, Ioannis Pappas, Paola Finoia, Guy B. Williams, Judith Allanson, John D. Pickard, Adrian M. Owen, Lorina Naci, David K. Menon, and Emmanuel A. Stamatakis. Consciousness-specific dynamic interactions of brain integration and functional diversity. *Nature Communications*, 10(4616), 2019. tex.date-added: 2021-01-25 10:56:48 +0100 tex.date-modified: 2021-01-25 10:56:48 +0100.

[100] Viol A., Palhano-Fontes Fernanda, Onias heloisa, de Araujo Drauilo B, and Viswanathan G M. Shannon entropy of brain functional complex networks under the influence of the psychedelic Ayahuasca. *Sci Rep.*, 7(1)::7388, 2017. tex.date-added: 2021-01-25 10:56:48 +0100 tex.date-modified: 2021-01-25 10:56:48 +0100.

[101] Christopher Timmermann, Leor Roseman, Michael Schartner, Raphael Milliere, Luke T. J. Williams, David Erritzoe, Suresh Muthukumaraswamy, Michael Ashton, Adam Bendrioua, Okdeep Kaur, Samuel Turton, Matthew M. Nour, Camilla M. Day, Robert Leech, David J. Nutt, and Robin L. Carhart-Harris. Neural correlates of the DMT experience assessed with multivariate EEG. *Scientific reports*, 2019. tex.date-added: 2021-01-25 10:56:48 +0100 tex.date-modified: 2021-01-25 10:56:48 +0100.

[102] Inês Hipólito, Jonas Mago, Fernando E Rosas, and Robin Carhart-Harris. Pattern breaking: a complex systems approach to psychedelic medicine. *Neuroscience of Consciousness*, 2023(1):niad017, January 2023.

[103] Hector Zenil. A review of methods for estimating algorithmic complexity: Options, challenges, and new directions. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies*, 22(6):612, May 2020. Publisher: MDPI AG.

[104] Juan A. Gallego, Matthew G. Perich, Lee E. Miller, and Sara A. Solla. Neural Manifolds for the Control of Movement. *Neuron*, 94(5):978–984, June 2017. Publisher: Elsevier.

[105] Rufus Mitchell-Heggs, Seigfred Prado, Giuseppe P. Gava, Mary Ann Go, and Simon R. Schultz. Neural manifold analysis of brain circuit dynamics in health and disease. *Journal of Computational Neuroscience*, 51(1):1–21, 2023.

[106] Saxena S, Russo Aa, Cunningham J, and Churchland Mm. Motor cortex activity across movement speeds is predicted by network-level strategies for generating muscle activity. *eLife*, 11, May 2022. Publisher: Elife.

[107] Connor Brennan and Alexander Proekt. A quantitative model of conserved macroscopic dynamics predicts future motor commands. *eLife*, 8:e46814, July 2019. Publisher: eLife Sciences Publications, Ltd.

[108] Cátia Fortunato, Jorge Bennasar-Vázquez, Junchol Park, Joanna C. Chang, Lee E. Miller, Joshua T. Dudman, Matthew G. Perich, and Juan A. Gallego. Nonlinear manifolds underlie neural population activity during behaviour, April 2024. Pages: 2023.07.18.549575 Section: New Results.

[109] Rishidev Chaudhuri, Berk Gerçek, Biraj Pandey, Adrien Peyrache, and Ila Fiete. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nature Neuroscience*, 22(9):1512–1520, September 2019. Number: 9 Publisher: Nature Publishing Group.

[110] Mark Stopfer, Vivek Jayaraman, and Gilles Laurent. Intensity versus Identity Coding in an Olfactory System. *Neuron*, 39(6):991–1004, September 2003. Publisher: Elsevier.

[111] Richard J. Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A. Baas, Benjamin A. Dunn, May-Britt Moser, and Edvard I. Moser. Toroidal topology of population activity in grid cells. *Nature*, 602(7895):123–128, February 2022. Publisher: Nature Publishing Group.

[112] Sung Soo Kim, Hervé Rouault, Shaul Druckmann, and Vivek Jayaraman. Ring attractor dynamics in the Drosophila central brain. *Science*, 356(6340):849–853, May 2017. Publisher: American Association for the Advancement of Science.

[113] Luigi Petrucco, Hagar Lavian, You Kure Wu, Fabian Svara, Vilim Štih, and Ruben Portugues. Neural dynamics and architecture of the heading direction circuit in zebrafish. *Nature Neuroscience*, 26(5):765–773, May 2023. Publisher: Nature Publishing Group.

[114] R P Rao and D H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, January 1999. Publisher: Springer Science and Business Media LLC.

[115] Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. *Equivariant and coordinate independent convolutional networks. A Gauge Field Theory of Neural Networks*. 2023.

[116] Giulio Ruffini, Fabrice Wendling, Roser Sanchez-Todo, and Emiliano Santarnecchi. Targeting brain networks with multichannel transcranial current stimulation (tCS). *Current Opinion in Biomedical Engineering*, 8:70–77, December 2018. Publisher: Elsevier BV.

[117] 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. *arxiv*, 2022. Publisher: arXiv.

[118] Mathieu Golos, Viktor Jirsa, and Emmanuel Daucé. Multistability in Large Scale Models of Brain Activity. *PLOS Computational Biology*, 11(12):e1004644, December 2015. Publisher: Public Library of Science.

[119] Beggs JM and Timme N. Being critical of criticality in the brain. *Frontiers in Physiology.*, 2012. tex.date-added: 2021-01-25 10:56:48 +0100 tex.date-modified: 2021-01-25 10:56:48 +0100.

[120] D Philipona, JK O'Regan, JP Nadal, and O Coenen. Perception of the structure of the physical world using unknown multimodal sensors and effector. In *Advances in neural information processing systems*, pages 945–952. MIT Press, 2004. tex.date-added: 2009-02-26 19:04:16 +0100 tex.date-modified: 2009-03-01 22:15:54 +0100.

[121] Paolo Ruffini. *Teoria generale delle equazioni, in cui si dimostra impossibile la soluzione algebraica delle equazioni generali di grado superiore al quarto.* Bologna, stamperia di s. Tommaso d'aquino, 1799.

[122] Evariste Galois. Mémoire sur les conditions de résolubilité des équations par radicaux. *Journal de mathématiques pures et appliquées*, pages 417–433, 1830.

[123] Ian Nicholas Stewart. *Galois theory.* Apple Academic Press, Oakville, MO, 4 edition, March 2015.

[124] Brian C. Hall. *Lie groups, lie algebras, and representations: An elementary introduction.* Springer, 2015.

[125] Stephen Wolfram. *A new kind of science.* Wolfram Media, 2002. tex.date-added: 2016-05-06 23:17:17 +0000 tex.date-modified: 2016-05-06 23:21:31 +0000.

[126] Tomaso Poggio. Foundations of deep learning: Compositional sparsity of computable functions. CBMM memo, Center for Brains, Minds, and Machines, MIT, Cambridge, MA, USA, January 2023. Issue: 138.

[127] Tomaso Poggio. Compositional sparsity: a framework for ML. CBMM memo, Center for Brains, Minds, and Machines, MIT, Cambridge, MA, USA, October 2022. Issue: 138.

[128] Morris W. Hirsch. Actions of Lie groups and Lie algebras on manifolds. In P. R. Kotiuga, editor, *A celebration of the mathematical legacy of raoul bott*, Proceedings & lecture notes vol. 50, pages 69–78, Providence, RI, 2010. Amer. Math. Soc.

[129] L D Landau and E M Lifshitz. *Mechanics*. Butterworth-Heinemann, Oxford, England, 3 edition, January 1982.

[130] Vladimir I. Arnold. *Mathematical methods of classical mechanics*. Springer, New York, 2 edition, 1989.

[131] H. Goldstein. *Classical mechanics*. Reading, MA: Addison-Wesley Publishing, second edition edition, 1980. tex.date-added: 2009-05-06 00:13:43 +0200 tex.date-modified: 2009-05-06 00:15:16 +0200.

A Appendix

A.1 Equations for a Turing pair

The general equations for a *Turing pair*—a description of the interaction of the agent and world—implemented as a network dynamical system are given by the autonomous system

$$\begin{aligned}
\dot{x} &= f_x\big(x; w_x, \eta_y\big) && \text{(A.1)}\\
\eta_x &= o_x(x) && \text{(A.2)}\\
\eta_y &\approx g_x(x) && \text{(A.3)}\\
\dot{y} &= f_y\big(y; w_y, \eta_x\big) && \text{(A.4)}\\
\eta_y &= o_y(y) && \text{(A.5)}
\end{aligned}$$

Or, explicitly providing the Compartor equation with the Lyupanov function,

$$\begin{aligned}
\dot{x} &= f_x\big(x; w_x, \eta_y\big) && \text{(A.6)}\\
E &= \eta_y - g_x(x) && \text{(A.7)}\\
\dot{V} &\leq 0,\ V = E^T E && \text{(A.8)}\\
\dot{y} &= f_y\big(y; w_y, \eta_x\big) && \text{(A.9)}\\
\eta_y &= o_y(y) && \text{(A.10)}
\end{aligned}$$

These equations express the idea that the outputs of a subsystem are the inputs of the other. They include the agent's world-tracking condition. Here, x is a multidimensional variable capturing the state of the agent, and y captures the state of the world. The ws are connectivity weights in the network (or other parameters), and the ηs 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. Finally, g_x is a variable selector (or, more generally, readout function) in the agent that is to match the inputs from the world (the world-tracking subsystem). The third equation, $\eta_y = g_x(x)$, or its Lyupanov version (allowing for transients), captures the world-tracking condition.

A.2 Notes on Lie groups

A.2.1 Definitions

Definition A.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 A.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 A.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 with a map from M to itself.[1] 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 A.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.

A.2.2 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.[1] 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).[124]

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[1]). 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. □

A.2.3 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 ϵ, the function f does not change to first order in ϵ. 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 ϵ 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 λ 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.

A.2.4 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 a commutative diagram linking the actions of the group in both scenarios exists. 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.

A.2.5 Further Notes on Lie Groups

The original inspiration sources for Lie were differential equations. But this is now unfortunately mostly forgotten:[1] 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.[1]

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.[1]

Infinitesimal symmetry conditions—the defining equations of the symmetry group—can usually be solved.[1] 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.[1]

Normally, all other things being equal, one seeks the most symmetric equations possible (simplicity).[1]

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.[1]

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-related 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[f], symmetry breaking can arise due to the interplay between asymmetric initial

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

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.

A.3 Groups, Turing Machines, and Generative Models

The exploration of group theory in the context of Turing machines and its extension to generative models like LLMs opens a novel theoretical perspective in understanding computational processes. Can we think of Turing machines, and hence computation, as a branch of group theory? While the practicality and exact mapping pose significant challenges, the conceptual framework offers intriguing possibilities for future research in computational theory and artificial intelligence.

Large Language Models (LLMs) like GPT-3 or GPT-4 can be seen as sophisticated generative models. These models generate new tokens (words or characters) based on a sequence of previous tokens, analogous to the way a Turing machine processes input to generate outputs driven by a complex set of rules (akin to a Turing machine's transition function).

Group theory, a branch of abstract algebra, deals with the study of groups that are sets equipped with a binary operation that satisfies certain axioms: closure, associativity, identity, and invertibility. This mathematical framework can be applied to studying computational models, such as Turing machines, particularly in understanding the symmetry and structure of computational processes.

To begin analyzing this, we will use the three-tape Turing machine with a finite private tape, which can be conceptualized as a generative model. In this model, the machine's states and the states of the private tape could be interpreted as elements of a group. The transition from one state to another, governed by the machine's rules and the symbols on the tapes, can be analogized to a group operation.

A three-tape Turing machine is an extension of the standard Turing machine model with three separate tapes: the input tape, the output tape, and a private (or work) tape. Each tape has its own tape head for reading and writing. It can be formally defined as a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ where:

- Q is a finite set of states.
- Σ is a finite input alphabet that does not contain the blank symbol $\sqcup$.
- Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$.
- $\delta : Q \times \Gamma^3 \to Q \times \Gamma^3 \times \{L, R, S\}^3$ is the transition function. Here, L, R, and S denote left shift, right shift, and no shift on the tapes, respectively.
- $q_0 \in Q$ is the initial state.
- $q_{\text{accept}} \in Q$ is the accept state.
- $q_{\text{reject}} \in Q$ is the reject state, distinct from the accept state.

The machine operates as follows:

- The input tape contains the input string and is read-only.
- The output tape is used to write the output and is write-only.
- The private tape is used for intermediate computations and can be both read from and written to.

- The transition function δ dictates the machine's actions based on the current state and the symbols under the tape heads. It specifies the next state, the symbols to write on each tape, and the movements of the tape heads.
- The computation begins in the initial state q_0 and proceeds according to the transition function until the machine enters either the accept state q_{accept} or the reject state q_{reject}.

The challenges in mapping Turing machines to group theory are

- **Closure:** Ensuring that the combination of any two states (or actions) results in another valid state within the system.
- **Associativity:** The sequence of transitions must not affect the final state of the machine, a non-trivial property to verify in computational processes.
- **Identity and Invertibility:** Identifying a state that acts as an identity element, and for each state, an inverse state, can be complex in the context of a Turing machine.

The minimal requirement is that of composition. We can relax the other two, and instead of groups, we have more general algebraic structures. A **magma**, or **groupoid**, is a foundational concept where a set is equipped with a binary operation, with the only requirement being closure. An extension of this is a **semigroup**, which is an associative magma, meaning the binary operation is associative. Building upon this, a **monoid** is essentially a semigroup that includes an identity element but does not necessarily have inverses for all its elements. Finally, a **group** is a monoid where every element has an inverse, thus completing the hierarchy from the most general structure (magma) to the more specialized one (group). While the concepts of Lie algebras and exponential maps are not directly applicable to magmas, semigroups, and monoids in the way they are to Lie groups, there are efforts and research in mathematics to explore analogous or related structures in these broader contexts.

For example, the concept of monoids becomes necessary when dealing with computational systems where inverses are not available, such as cellular automata like Rule 110.[125] A monoid is an algebraic structure similar to a group but does not require every element to have an inverse. In Rule 110, the evolution operator T that updates the cellular configuration lacks an inverse because the process is generally not reversible—the previous state cannot be uniquely determined from the current state.

The power of recognizing the monoid structure in Rule 110 lies in understanding how the repeated application of a simple rule can generate highly complex behavior. Despite the simplicity of the rule (a single generator T), the iterative process T^n produces rich dynamics that are capable of universal computation (Rule 110 is Turing complete). This highlights that even without the full framework of group theory, the presence of an underlying algebraic structure like a monoid is sufficient to capture the essence of the system's generative process.

A.4 Groups and algorithmic complexity

Can the association of group action with generative models be generalized to general compression? Under what conditions does the existence of a highly compressive generative model imply the presence of underlying symmetries described by a group? Although rigorously answering this will require bridging abstract algebra and algorithmic information theory[g], the intuition is straightforward: a compressive program capable of generating a large dataset will involve repeating an operation or a sequence of operations multiple times. The recursive property of Lie groups, where complex elements are constructed through repeated applications of simple algebraic operations, mirrors foundational concepts in algorithmic information theory and recursive function theory. The discrete analogs of Lie groups are finitely generated groups, where a finite set of generators produces all elements, with cyclic groups representing cases generated by a single element[h].

To explore this, let's first consider the converse: 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 be manipulated using the Lie group's generators. This approach can be highly efficient. By generating a single instance within our model, we can apply the Lie group's transformations in small, discrete steps to produce a diverse set of outputs. This process is inherently recursive and compressive, suggesting that an entire class of data, such as images of cats, can be generated from a relatively short program. The program only needs 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 illustrate, recall that a small transformation generated by a Lie group element γ can be described as a perturbation around the identity element as $\gamma = 1_G + \epsilon T$, where γ is a group element representing a small transformation, 1_G is the identity element representing no transformation, ϵ 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 γ in the vicinity of the identity element, utilizing the Lie algebra structure derived from the Lie group. Similar to the exponential function's definition via a limit,

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

elements of Lie groups can be expressed as exponentials of Lie algebra elements with profound implications (see 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 $\gamma = 1 + \sum_k \epsilon_k T^k$,

$$g = \lim_{n\to\infty} \left(1 + \frac{1}{n}\sum_k \theta_k T^k\right)^n = \exp[\sum_k \theta_k T^k]. \tag{A.11}$$

[g]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 have limited resources).

[h]In Section A.3, 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.

where θ_k are parameters and T^k are the generators of the Lie algebra. If G is a connected, compact matrix Lie group, the exponential map for G is surjective (covers the entire group).[124] This *recursive* expression encapsulates the fundamental property of Lie groups that any group element can be decomposed into an (infinite) series of infinitesimal transformations.

This concept extends to any representation or action of the group on a manifold. The action preserves the group structure and the number of its generators, and the simplicity of the underlying Lie group is maintained for functions on a manifold under the action of the Lie group. The Lie algebra's infinitesimal generators T_i (z) correspond to vector fields on the target manifold, and the group action on a point θ is given by the exponential map. This map can be seen 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. While this computation involves higher-order derivatives of f, the linearity of these operators and the structure constants of the Lie algebra may simplify the evaluation. Recurrence patterns and algebraic properties such as *nilpotency* can reduce the number of distinct terms that need to be computed, depending on the Lie's group characteristics.

In the other direction, if a model can efficiently and globally encapsulate the complex and continuous variability of a dataset using a limited set of parameters and a short program length, it's plausible to conjecture that the parameters correspond to a Lie algebra. This is especially relevant in the context of compositional generative functions, where one can write

$$I(\theta) = f_n(f_{n-1}(...f_1(\theta))) \tag{A.12}$$

The compositional nature of the function—a form of recursion—is associated with a shorter set of computations to evaluate it.[65] Thus, the computation of the generators of the transformations of the object I becomes simpler. An explicit form of a compositional Lie-generative model is (a special case of Eq. 2.1), with $\gamma = (\gamma_1, ..., \gamma_M)$,

$$I(\gamma) = f_M(\gamma_M \cdot f_{M-1}(...\gamma_2 \cdot f_1(\gamma_1 \cdot x_0))) \tag{A.13}$$

This hierarchical functional structure relates to Poggio's conjecture,[126] which states that the *compositional sparsity* of the underlying target function (i.e., the world) —the task to be learned—is the key principle behind the success of machine learning, particularly 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,[127] is equivalent to efficient computability, that is, computability in polynomial time.[126] Thus, algorithmic and computational complexity notions may converge on the role of Lie action in generative (compressive) models.

A.5 Symmetry in ODEs: abstract vs. traditional definitions

Symmetry plays a fundamental role in the analysis and solution of ordinary differential equations (ODEs). In this section, we present an abstract definition of symmetry based on transformations of initial conditions (ICs) and compare it with the traditional definition that involves transformations of variables. We demonstrate that both definitions are equivalent in their ability to map solutions to solutions while preserving the form of the ODE and explore the structure of the resulting symmetry groups, highlighting that they can be either abelian or non-abelian, depending on the system.

A.5.1 Olver's definition of symmetry of PDEs[1,2]

Consider a general system of nth order (partial) differential equations

$$\Delta_\nu(x, u^{(n)}) = 0, \quad \nu = 1, \ldots, m,$$

in p independent variables $x = (x^1, \ldots, x^p)$, and q dependent variables $u = (u^1, \ldots, u^q)$, with $u^{(n)}$ denoting the derivatives of the u's with respect to the x's up to order n. In general, by a *symmetry of the system (1)*, we mean a transformation that takes solutions to solutions. The most basic type of symmetry is a (locally defined) invertible map on the space of independent and dependent variables:

$$(\bar{x}, \bar{u}) = g \cdot (x, u) = (\Xi(x, u), \Phi(x, u)).$$

Such transformations act on solutions $u = f(x)$ by pointwise transforming their graphs; in other words, if $\Gamma_f = \{(x, f(x))\}$ denotes the graph of f, then the transformed function $\bar{f} = g \cdot f$ will have the graph

$$\Gamma_{\bar{f}} = \{(\bar{x}, \bar{f}(\bar{x}))\} = g \cdot \Gamma_f \equiv \{g \cdot (x, f(x))\}. \tag{2}$$

Definition A.5. A local Lie group of transformations G is called a *symmetry group* of the system of partial differential equations (1) if $\bar{f} = g \cdot f$ is a solution whenever f is.

A.5.2 Abstract Definition of Symmetry for ODEs

Here we provide a generalized definition of symmetry. As it turns out, it is closely aligned with that of Olver for PDEs.[1,2] Consider an autonomous ODE of the form:

$$\dot{x} = f(x), \tag{A.14}$$

where $x \in \mathbb{R}^n$ is a vector of dependent variables, and $f : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function.

Let $x(t; x_0)$ denote the unique solution of the ODE with the initial condition $x(0) = x_0$. The mapping from the initial condition x_0 to the solution $x(t; x_0)$ establishes a one-to-one correspondence between initial conditions and solutions due to the existence and uniqueness theorems for ODEs.

Definition (abstract definition of symmetry): A *symmetry* of the ODE is any **bijective transformation** γ acting on the initial conditions x_0, such that:

$$x_0 \mapsto x_0' = \gamma \cdot x_0, \tag{A.15}$$

which induces a transformation of the solution:

$$x(t; x_0) \mapsto x'(t) = x(t; x_0') = x(t; \gamma \cdot x_0). \tag{A.16}$$

Since each initial condition corresponds to a unique solution, any bijective transformation γ on x_0 automatically maps a solution to another solution that satisfies the same ODE,

$$\dot{x}' = f(x'). \tag{A.17}$$

Thus, the transformation γ preserves the form of the ODE by ensuring that $x'(t)$ satisfies the original differential equation.

The set of these transformations forms a Lie group, the group of diffeomorphisms of the manifold of initial conditions. Under some conditions, the Lie group has a finite number of generators.

A.6 Are all Lie generative models finite-dimensional?

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 itself or its model can both 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 notion that a generative model can be described by a finite Lie group highlights the importance of symmetry and smoothness in modeling natural data. When a model exhibits these properties, it allows us to understand and navigate the configuration space in a structured and efficient manner, which is particularly useful in practical applications.

If natural data can be captured in a latent space—a lower-dimensional representation where complex relationships are simplified—we can choose an arbitrary point in this latent space as a reference. This is the case with neuroimaging data, which includes complex brain activity patterns obtained from techniques like fMRI or EEG. Embedding such data in a latent space (e.g., processed using a variational autoencoder[90]) reveals underlying structures and patterns not immediately apparent in the high-dimensional raw data. By studying the relationship between displacements in different directions around the reference point in the latent space, we can gain insights into how variations in the data correspond to meaningful changes in brain activity. 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 C be a smooth manifold. We say that C admits a *point-transitive finite-dimensional Lie subgroup* of $\mathrm{Diff}(C)$ if there exists a finite-dimensional Lie subgroup $G \subseteq \mathrm{Diff}(C)$ such that for any two points $x, y \in C$, there exists $g \in G$ such that $g(x) = y$. In the case of **1D manifolds**, every connected, smooth 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)**.[128] According to the theorem, a 2D manifold C without boundary admits a point-transitive Lie group action if and only if C 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.

A.7 Conserved quantities and canonical variables

In this section, we provide some background on the theory of symmetries and constraints in ODEs. Let us consider, for simplicity, a well-behaved autonomous system of X equations and unknowns,

$$\dot{x} = f(x; w) \tag{A.18}$$

(with a smooth function f). The solution involves X integration constants. Paraphrasing the classical argument in Landau and Lifschitz,[129] 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 (at least locally) 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)[i], 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,

$$\Phi_n(x) = x_n - x_n(x_0, C) = 0, \;\; n = 1, ..., X - 1 \tag{A.19}$$

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 A.19 to express the remaining Cs as a function of x. We thus obtain $X - 1$ independent constants of the motion,

$$C_n = Q_n(x) \;\; n = 1, ..., X - 1 \tag{A.20}$$

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[j], which results, as we already knew from the start, in one-dimensional trajec-

[i] 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 $x_0 = t + t_0$.

[j] *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[130] 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.

tories.

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

When it is possible to find such canonical variables globally, the system is called integrable. An *integrable system* 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.

A.7.1 Some examples

Consider the forced non-relativistic particle, $\ddot{x} = a$, or, in first-order form

$$\dot{x} = p \tag{A.21}$$

$$\dot{p} = a \tag{A.22}$$

If $x(t)$ is a solution, then $x(t) \to \tilde{x}(t) = x(t) + \delta x$ is also a solution. The transformation $p(t) \to \tilde{p}(t) = p(t) + \delta p$, $x(t) \to \tilde{x}(t) = x(t) + \delta p\, t$, is also a symmetry, since

$$\dot{\tilde{x}} = \dot{x} + \delta p = \tilde{p} \tag{A.23}$$

$$\dot{\tilde{p}} = a \tag{A.24}$$

so, the overall symmetry of the ODE system is

$$x(t) \quad \longrightarrow \quad \tilde{x}(t) = x(t) + \delta p t + \delta x \tag{A.25}$$

$$p(t) \quad \longrightarrow \quad \tilde{p}(t) = p(t) + \delta p \tag{A.26}$$

The solution to the equations with $a = 0$ is $p = p_0$ and $x = t + t_0$, for arbitrary constants p_0 and t_0. Following the argument above, we have $t + t_0 = x$ (the x coordinate can be used as a natural clock) and the conservation law $C(x, p) = p = p_0$. The symmetry transformation shifts C by δp.

It is instructive to consider the case $a \neq 0$. Here, we have translational invariance without momentum conservation. Again, we use a variable as a clock, this time $p/a = t + t_0$, and

express the solution as a function of the clock and constants of integration,

$$x = \int p\,dt = \int a(t+t_0)\,dt = \frac{1}{2}at^2 + at_0 t + \tilde{C} = \frac{1}{2a}p^2 + C$$

The conserved quantity is $C = x - \frac{1}{2a}p^2$. The symmetry transformation shifts C by $\tilde{C} - C = \Delta C = \delta x - \frac{3(\delta p)^2}{2a}$.

A.7.2 A not-so-simple example of constrained dynamics

Consider the constrained ODE (differential algebraic equation)

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 \\ x_1^2 + x_2^2 &= C^2 \end{aligned} \tag{A.27}$$

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

$$\begin{aligned} x_1 &= A\sin(t+t_0) \\ x_2 &= A\cos(t+t_0) \end{aligned} \tag{A.28}$$

for any constant A, which automatically guarantees that $x_1^2+x_2^2 = A^2$. Hence, because the constraint is compatible with the conserved quantities of the ODEs, finding a solution to the three equations is possible. Adding an incompatible constraint, e.g., $x_1 = 1$, 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),

$$\begin{aligned} \dot{\phi} &= 1 \\ \dot{r} &= 0 \\ r^2 &= C^2 \end{aligned} \tag{A.29}$$

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, $r \to r + \epsilon$—with its associated conserved variable r. Again, the conserved variable corresponds to a center manifold (zero real part eigenvalue of the system).

A.7.3 Approximate symmetry after transients

If the second equation were instead $\dot{r} = -(r - C)$, we would have the constraint holding in steady state after transient dynamics,

$$\begin{aligned} \dot{\phi} &= 1 \\ \dot{r} &= -(r - C) \\ r^2 &\to C^2 \end{aligned} \tag{A.30}$$

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 time scale, the equations will still hold. The input, instead of the initial conditions, implements the group control on the solution space.

This equation has a Lyupanov function, which is also invariant under a simultaneous translation of r and C, namely $V = (r - C)^2/2 \geq 0$, since $\dot{V} = \dot{r}(r - C) = -(r - C)^2 \leq 0$.

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

$$\begin{aligned} \dot{\phi} &= 1 \\ \dot{r} &= y \\ \dot{y} &= -y \\ r^2 &\to C^2 \end{aligned} \tag{A.31}$$

which is equivalent to

$$\begin{aligned} \dot{\phi} &= 1 \\ \dot{r} &= ae^{-t} \\ r^2 &\to C^2 \end{aligned} \tag{A.32}$$

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. The Lyupanov function is $V = r^2/2 \geq 0$, since $\dot{V} = \dot{r}r = ae^{-t}r \leq 0$ for $a < 0$.

A.7.4 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

$$\dot{Q}_i = 0, \quad i = 1, \ldots, X-1, \tag{A.33}$$

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

$$Q_i \to Q_i + \epsilon_i, \quad i = 1, \ldots, X-1, \tag{A.34}$$

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,

$$V_i = \frac{\partial}{\partial Q_i}. \tag{A.35}$$

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

$$[V_i, V_j] = 0, \tag{A.36}$$

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

$$Q \to Q' = (1 + \epsilon V_n) \cdot Q_i = Q_i + \epsilon \delta_{ni} \tag{A.37}$$

with δ_{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.[131]

Next, we impose a first hierarchical constraint $\mathcal{C}_1$, $\mathcal{C}_1(\mathbf{y}_1) = r - 1 = 0$, which fixes $r = 1$. This constraint reduces the state space to the surface of a unit sphere. The manifold at this level is defined as

$$\mathcal{M}_1 = \left\{ \mathbf{x} \in \mathbb{R}^3 \;\middle|\; \mathcal{C}_1\big(\mathcal{G}_1(\mathbf{x})\big) = 0 \right\} = \left\{ \mathbf{x} \in \mathbb{R}^3 \;\middle|\; r(\mathbf{x}) = 1 \right\}.$$

Since we have discarded ϕ, the remaining variable is $\theta \in [0, \pi]$. However, because ϕ is not specified, for each value of θ, there is a circle of points (a line of latitude) on the sphere. Therefore, our manifold $\mathcal{M}_1$ corresponds to the set of all such circles on the unit sphere.