Equivariant Deep Dynamical Model for Motion Prediction

Bahar Azari
Northeastern University
azari.b@northeastern.edu

Deniz Erdoğmuş
Northeastern University
Erdogmus@ece.neu.edu

Abstract

Learning representations through deep generative modeling is a powerful approach for dynamical modeling to discover the most simplified and compressed underlying description of the data, to then use it for other tasks such as prediction. Most learning tasks have intrinsic symmetries, i.e., the input transformations leave the output unchanged, or the output undergoes a similar transformation. The learning process is, however, usually uninformed of these symmetries. Therefore, the learned representations for individually transformed inputs may not be meaningfully related. In this paper, we propose an SO(3) equivariant deep dynamical model (EqDDM) for motion prediction that learns a structured representation of the input space in the sense that the embedding varies with symmetry transformations. EqDDM is equipped with equivariant networks to parameterize the state-space emission and transition models. We demonstrate the superior predictive performance of the proposed model on various motion data.

1 Introduction

Deep dynamical system models are introduced to cope with the potential non-linearities found in real-world systems. These models are constructed in such a way that they learn rich but compressed representations of the given data through structured deep generative modeling that captures the underlying complex distribution of data (Watter et al., 2015; Karl et al., 2017; Krishnan et al., 2017; Linderman et al., 2017; Fraccaro et al., 2017; Becker et al., 2019; Nassar et al., 2019; Farnoosh and Ostadabbas, 2021; Farnoosh et al., 2021; Shamsabardeh et al., 2021). The input to the models is usually multiple realizations of the system behavior we want to capture, called trajectories. Each realization is a set of spatially correlated time series consisting of measurements such as 3D coordinates indicative of the position of an object collected over time, such as the position of joints in skeletal data or the ball coordinates in swinging pendulum. Inevitably, the predictive model that is trained on these trajectories cannot be used for another one where the relative position and orientation of the coordinates system have changed. Figure. 1 shows two otherwise similar pendulums placed in two planes parallel to z-axis and rotated by θ with respect to one another. It is intuitive to see that the underlying dynamic of the swinging pendulum, i.e., its angular acceleration and velocity, remains the same regardless of its relative orientation to the coordinate system. This is because rotation is one of the symmetries of the swinging pendulum as a dynamic system. One of the main limitations of the current generative models is that they do not consider the symmetries of the model at hand.

Symmetry refers to a transformation that leaves an object (or its higher-level representation) invariant. Symmetries can also be associated with tasks. For example, translations are symmetries of the object classification task, and so where the object is inside an image should not matter to the classifier. Therefore, a model (e.g., a neural network) should process different but correspondent versions of an object under these transformations equivalently. Exploiting symmetry has a long history in physical sciences. However, recently, many studies in the literature focused on incorporating symmetries into variants of deep neural networks to learn images, sets, point clouds, and graphs, (Cohen and Welling 2017; Kondor and Trivedi 2018; Maron et al. 2019; Cohen et al. 2019; Keriven and Peyré 2019; Feige 2019; Wang et al. 2020; Walters et al. 2021).

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

In most of the studies, with the help of mathematical tools such as group theory and representation theory, a *global* architecture is designed for a learning problem in such a way that it is invariant or equivariant under various transformations of the input. In other words, the input is treated as a single object whose symmetries should be preserved throughout the network. However, to the best of our knowledge, limited works exist investigating the integration of equivariance/invariance into the more complex Markovian generative models. The complicated structure of these models, which are associated with multiple levels of latent variables connected through neural networks, renders the design of an efficient equivariant model more challenging. Specifically, in this scenario, we need to consider each *local* network connecting a part of an input to its latent, or a current latent to the future latent, etc.

In this paper, we investigate the role of symmetry in learning of a dynamic system characterized with an equivariant deep state space model. We propose an SO(3)-equivariance deep dynamical model for motion prediction, and call it EqDDM. Our model, which inherits a Markovian structure from its state-space model counterpart, is equipped with a chain of appropriately equivariant/invariant MLPs similar to those in Finzi et al. (2021). Specifically, we propose to use a hierarchical equivariant structure where we have an equivariant network from each input to its representation (latent), followed by an equivariant switching network from the current latent to the future latent, followed by an invariant switching network controlling each switch (see figure. 3 **Right**).

The structure of the paper is organized as follows. After covering the relevant studies around the subject in section. 2, we review the necessary background on group theory, representation theory, and state-space models in section. 3. We then outline the designing steps of an equivariant/invariant network in section. 4. In section. 5, we characterize our dynamic learning problem with a generative state space model. We then describe how to integrate equivariant/invariant architectures in a Markovian structure. We evaluate the performance of our model on motion prediction of a skeletal object and provide the results in section. 6.

2 Related Work

Incorporating symmetry into deep neural networks has been the focus in many studies due to its compelling promise to improve generalization and accuracy. The line of research began with the attempt to generalize the idea of the translation equivariant convolution layer in CNN to other spaces and symmetry groups, as one can associate the ability of CNN with the fact that it exploits the translational symmetry (Gens and Domingos, 2014; Olah, 2014; Dieleman et al., 2015; Guttenberg et al., 2016; Dieleman et al., 2016; Cohen and Welling, 2017; Ravanbakhsh et al., 2016, 2017; Worrall et al., 2017; Maron et al., 2020; Dym and Maron, 2021; Finzi et al., 2020; Satorras et al., 2021). Early studies focused on discrete groups for their ease of understanding (Cohen and Welling, 2016; Maron et al., 2019; Zaheer et al., 2017). Some works have been investigating equivariance to continuous groups and generalized the CNN to various spaces (Cohen et al., 2018; Kondor and Trivedi, 2018; Cohen et al., 2019; Walters et al., 2021; Azari and Erdogmus, 2021). Lately, Finzi et al. (2021) has has generalized quivariant multilayer perceptrons (MLPs) for arbitrary matrix groups.

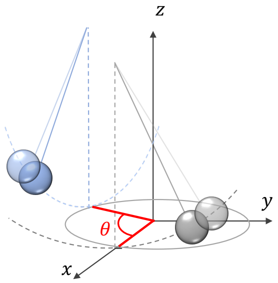


Figure 1: Rotation is one of the symmetries of the swinging pendulum as a dynamic system. Rotations of the coordinate system, although alter the relative position of the points in the collected data, leave the underlying dynamical system unchanged.

Incorporating symmetry into complex probabilistic deep generative models has not been entirely examined. Some studies tackled the problem of learning equivariant and invariant representations using variational autoencoders (Feige, 2019; Esteves et al., 2018; Qi et al., 2019; Gao et al., 2020; Köhler et al., 2020). Other studies integrated symmetries in dynamic models by defining a new equivariant convolutional layer (Walters et al., 2021; Wang et al., 2020). To the best of our knowledge, our study is the first attempt at designing an equivariant deep state-space model.

3 Background

We begin by explaining the required building blocks of our model starting from Lie groups and their representations. Note that in this paper, we use the word representation in two different contexts: representation as a compact but informative numeric feature that captures relevant information regarding an input signal, and representation as an invertible matrix, representative of a group element, that act on a vector space.

3.1 Lie Group & Infinitesimal Generator

A Lie group G is a smooth manifold equipped with the structure of a group such that the group operation and inverse-assigning operation are smooth functions. The manifold is locally represented by a chart mapping (ψ) to an underlying Euclidean space $\mathbb{R}^D$, where D is the dimensionality of the manifold. Furthermore, the chart map is defined in such a way that it associates the identity element in the group with the origin of Euclidean space. Elements of the Lie group can *act* as a transformation on the basis an n-dimensional vector space known as the *geometric space*, and change the coordinates of elements accordingly (see figure. 2 for a visualisation). We analyze Lie groups in terms of their infinitesimal generators which are the derivative of the group elements with respect to its D underlying parameters at the identity. These infinitesimal generators are $n \times n$ matrices $\{A_1, A_2, \dots, A_D\}$, that are the bases for a new vector space, or more strongly an Algebra, called the Lie Algebra $\mathfrak{g}$. We can relate an element in the Lie Algebra, which is a linear combination of the generator matrices, to an element in the Lie group through the exponential map $\exp : \mathfrak{g} \to G$ as $g = \exp\left(\sum_{i=1}^{D} \alpha_i A_i\right)$ (refer to Gilmore (2006) for more details).

3.2 Lie Algebra Representations

A linear finite dimensional group representation is a map $\rho : G \to \mathrm{GL}_n(\mathbb{R})$ from the abstract notion of a group element to an invertible matrix belonging to the general linear group $\mathrm{GL}_n(\mathbb{R})$ – the set of $n \times n$ invertible matrices. The group element represented in this form is the transformation that acts on the elements (i.e., their corresponding coordinates) in a vector space V_n called the geometric space. The representation should satisfy $\rho(g_1 g_2) = \rho(g_1 g_2) = \rho(g_1)\rho(g_2)$ and consequently $\rho(g^{-1}) = \rho(g)^{-1}$ for all $g_1, g_2 \in G$. Furthermore, Lie Algebra of a Lie group has a corresponding matrix representation denoted as $d\rho : \mathfrak{g} \to \mathfrak{gl}_n(\mathbb{R})$ that linearly maps Lie Algebra elements to the set of $n \times n$ matrices (commonly denoted as $\mathfrak{gl}_n(\mathbb{R})$). The representation of a Lie Group and the that of its Lie Algebra is related by:

$$\rho(e^A) = e^{d\rho(A)} \qquad \forall A \in \mathfrak{g}. \tag{1}$$

3.3 Lie group SO(n)

For the rotation group SO(n), the dimensionality of the group and its Lie Algebra, $D = \dim(\mathfrak{so}(n)) = \dim(\mathrm{SO}(n))$, is given by ${}^{n(n-1)}/_{2}$. The group elements are matrices $R \in \mathbb{R}^{n \times n}$ such that $R^\top R = I$ and $\det(R) = 1$, and the anti-symmetric matrices constitute its Lie Algebra $\mathfrak{so}(n) = \{A \in \mathbb{R}^{n\times n} : A^\top = -A\}$.

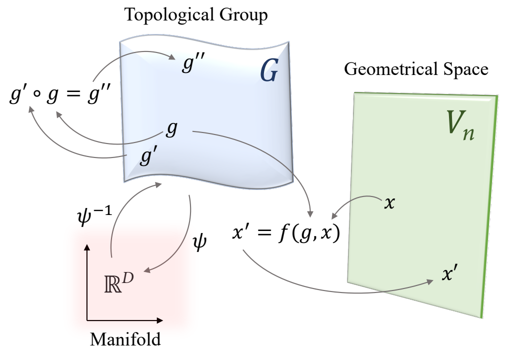


Figure 2: A Lie group of $n \times n$ matrices and its corresponding D dimensional manifold acting on a n dimensional vector space V_n.

3.4 Tensor Representations

Tensor manipulation is required to build increasingly complex representation matrices with increasingly more parameters to construct an expressive neural network function. Given a base group representation ρ acting on a vector space V, and its corresponding Lie Algebra representation $d\rho$, we can design larger and more sophisticated representations using the following tensor operations: Direct sum ($\oplus$) acts on matrices and concatenates them on the diagonal as $X \oplus Y = \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix}$. Tensor Product($\otimes$) is the standard Kronecker product. Finally, V^* is the dual space of V, and its corresponding Lie group and Lie Algebra representation are $\rho(g^{-1})^\top$ and $-d\rho(A)^\top$, respectively. Using the tensor product and dual operator, we can describe linear maps between two vector spaces. Linear maps from $V_1 \to V_2$ form the vector space $V_2 \otimes V_1^*$ and have the corresponding representation $\rho_2 \otimes \rho_1^*$. More notations used throughout the paper are as follows. We denote several copies of the same vector space $\underbrace{V \oplus V \oplus \cdots \oplus V}_{m}$ as mV. We also refer to the vector space formed by several tensor products as $T_{(p,q)} = V^{\otimes p} \otimes V^{*\otimes q}$ where $(\cdot)^{\otimes p}$is the tensor product iterated p times.

4 SO(3)-equivariant Network Design

In this section, we describe the general framework for designing an equivariant (invariant) architecture inspired by Finzi et al. (2021), to then deploy it in our deep generative model with Markovian structure. Specifically, we focus on designing an SO(3)-equivariant network since SO(3) constitutes the symmetries of the representation learning in the dynamic modeling problem at hand. We start with a general Lie group G and then provide the specific results for SO(3).

4.1 Equivariant Linear Layer

We require the group transformation that the input undergoes to traverse the network and appear in the output (e.g., latent space). This goal is achievable by defining each layer of the network to be equivariant to the group action. To formally establish the equivariance property, we define the vector spaces V_1 and V_2 (of dimensionality N_1 and N_2) to represent the input and output of a network layer, respectively. We also define the action of the group on these two vector spaces as transformation matrices $\rho_1(g) : G \to \mathrm{GL}_{N_1}(\mathbb{R})$ and $\rho_2(g) : G \to \mathrm{GL}_{N_2}(\mathbb{R})$. An equivariant linear layer is parameterized with the weight matrix $W \in \mathbb{R}^{N_2 \times N_1}$ that maps the V_1 to V_2. Equivariance implies that transforming input is equivalent to correspondingly transforming the output and since it is true for all the input $x \in V_1$, we can say:

$$\rho_2(g)W = W\rho_1(g) \qquad \forall g \in G.$$

We can simplify the equality by using the tensor product and $vec(\cdot)$ operator manipulation (see the supplementary for details) to reach to:

$$\left(\rho_2(g) \otimes \rho_1(g^{-1})^\top\right) vec(W) = vec(W) \quad \forall g \in G, \quad (2)$$

where the $vec(\cdot)$ operator creates a column vector from a matrix by stacking the its column vectors below one another. The representation $\rho_1(g^{-1})^\top$ is the dual representation $\rho_1^*(g)$. If we closely examine equation. 2, we realize that it is similar to an invariance equality where the representation $(\rho_2 \otimes \rho_1^*)(g) = \rho_2(g) \otimes \rho_1(g^{-1})^\top$ on the left-hand side is just a more intricate representation composed of two simple ones through the tensor operations. Specifically, we can say the vectorized version of matrices mapping from $V_1 \to V_2$ create the vector space $V_2 \otimes V_1^*$, and every group element g acts on it with the representation $\rho = (\rho_2 \otimes \rho_1^*)$:

$$\rho(g)v = v \quad \forall g \in G,\ \forall v \in V_2 \otimes V_1^*. \qquad (3)$$

Elements in G can be finitely generated by taking the $\exp(\cdot)$ of some linear combination of its Lie Algebra bases (the generators). Hence, we can write the representation of g in terms of the exponential of the generator matrices, $\{A_i\}_{i=1}^D$, and relate it to its Lie algebra representation as:

$$\rho(g) = \rho(\exp(\sum_{i=1}^{D} \alpha_i A_i)) = \exp(\sum_{i=1}^{D} \alpha_i d\rho(A_i)) \quad \forall \alpha_i. \qquad (4)$$

Note that for the latter equality, we used the correspondence in (1) and the fact that $d\rho(\cdot)$ is linear. Substituting the obtained $\rho(g)$ in (3) we have:

$$\exp(\sum_{i=1}^{D} \alpha_i d\rho(A_i))v = v \quad \forall \alpha_i. \qquad (5)$$

Since (5) is true for all α_is, it is also true for its derivative with respect to α_i at $\alpha = 0$. Hence, we get D constraints collected in a larger matrix as:

$$Cv = \begin{bmatrix} d\rho(A_1) \\ \vdots \\ d\rho(A_D) \end{bmatrix} v = 0. \qquad (6)$$

Note that since all the A_is are known, and different $\rho(A_i)$ can be constructed using the tensor representation, the problem in (6) is a standard nullspace problem addressed in Finzi et al. Specifically, they used a Krylov method for efficiently solving the nullspace problem by exploiting structure in the matrices ρ and $d\rho$ (see Finzi et al. (2021) for more detail). The obtained nullspace $Q \in \mathbb{R}^{n\times r}$ can apply symmetry to the arbitrary weight matrix flattened as v, where r is the rank of the nullspace, and n is the dimensionality of v. In practice, we can parameterize a set of weights v_0 and project them onto the equivariant subspace by $v = QQ^\top v_0$, which encourages the weight-sharing scheme. Through the linear combination induced by the projection, the unique elements in v_0 are projected into a set of repeating (i.e., *shared*) elements in v. Although this structured weight sharing promotes equivariance, it comes with the caveat that the expressive power of the network is restricted in that the number of parameters is now limited. For example, for $G = \mathrm{SO}(3)$, the weight matrix W reshaped from rows of v obtained from solving (6) has only one non-zero parameter, and hence limited expressive power.

To remedy this issue, we need to use more complex representations, constructed through tensor operations, to obtain more parameters through achieving a higher nullspace rank. Considering a feature space U in a neural network, it can be a combination of tensors with different ranks. For example, if we consider the group SO(3) acting on $\mathbb{R}^3$, a complex feature space can be constructed using scalars, denoted as T_0, 3 dimensional vectors, denoted as T_1, 3×3 matrix denoted as T_2, and so on. Using the notation described in the background section, we can write $U = c_0T_0 \oplus c_1T_1 \oplus \cdots \oplus c_MT_M$. The representation for a linear map $U_1 \to U2$ between two feature space in the network with the corresponding representations $\rho_{U_1}(g) = \bigoplus_{a\in\mathcal{A}_1} \rho_a(g)$ and $\rho_{U_2}(g) = \bigoplus_{b\in\mathcal{A}_2} \rho_b(g)$, is given by:

$$\rho_2 \otimes \rho_1^* = \bigoplus_{b\in\mathcal{A}_2} \rho_b \otimes \bigoplus_{a\in\mathcal{A}_1} \rho_a^* = \bigoplus_{(b,a)\in\mathcal{A}_2\times\mathcal{A}_1} \rho_b \otimes \rho_a^*. \quad (7)$$

To design a fully equivariant MLP from the described equivariant linear layer, we use the gated nonlinearities introduced in (Weiler et al., 2018), and used in Finzi et al. (2021) as an equivariant nonlinearity. In the next section, we propose our equivariant deep dynamic model using the building blocks described here.

5 Equivariant Deep Dynamical Model

We propose an equivariant deep dynamical model (EqDDM) based on the equivariant linear map described in the section. 4. We describe its associated generative and inference model in the following.

5.1 Generative Model

Lets consider a set of N motion sequences $\{X_1, \ldots, X_N\}$, where each sequence $X_n \in \mathbb{R}^{T_n \times (D \times 3)}$ records 3D coordinates of D objects/skeletal joints over T_n time points. A switching dynamical model defines a generative distribution over this dataset according to a set of *discrete* dynamical states $\mathcal{S}_n = \{s_{n,t}\}_{t=1}^{T}$ and their corresponding *continuous* temporal latents $Z_n = \{z_{n,t} \in \mathbb{R}^K\}_{t=1}^{T}$ as follows:

$$\begin{aligned} x_{n,t} &\sim p_\theta(x_{n,t} \mid z_{n,t}), \\ z_{n,t} &\sim p_\theta(z_{n,t} \mid z_{n,t-\ell}, s_{n,t}), \\ s_{n,t} &\sim p_\theta(s_{n,t} \mid s_{n,t-1}, z_{n,t-1}), \end{aligned} \tag{8}$$

where θ collectively denotes generative distribution parameters and ℓ indicates a set of temporal lags (e.g., $\ell = \{1, 2\}$ for a second-order model). The graphical representation for our proposed generative model is depicted in 3.

Specifically, the distributions $p_\theta(s_{n,t}|s_{n,t-1}, z_{n,t-1})$ define a discrete Markovian prior over $\mathcal{S}_n$ (subscript n is dropped hereafter for brevity):

$$p_\theta(s_t \mid s_{t-1} = s, z_{t-1}) = \mathrm{Cat}\left(\boldsymbol{\pi}_{\boldsymbol{\theta}}^{s}\left(z_{t-1}\right)\right), \tag{9}$$

where $\boldsymbol{\pi}_{\boldsymbol{\theta}}^{s}(\cdot)$ is a state transition network that is set by the preceding state s_{t-1} and maps z_{t-1} to the prior distribution parameters of s_t (a.k.a. a recurrent state transition model (Linderman et al., 2017)). This probability is characterized by an invariant switching network. Note that the state of the system should be invariant to the translations of the latent variables.

The distributions $p_\theta(z_{n,t} \mid z_{n,t-\ell}, s_{n,t})$ define a switching dynamical autoregressive prior over Z_n (a.k.a. a transition model):

$$p_\theta(z_t|z_{t-\ell}, s_t = s) = \mathrm{Norm}\Big(\boldsymbol{\mu}_{\boldsymbol{\theta}}^{s}(z_{t-\ell}), \boldsymbol{\sigma}_{\boldsymbol{\theta}}^{s}(z_{t-\ell})\Big), \tag{10}$$

where state-specific $\boldsymbol{\mu}_{\boldsymbol{\theta}}^{s}(\cdot)$ and $\boldsymbol{\sigma}_{\boldsymbol{\theta}}^{s}(\cdot)$ are nonlinear mappings that parameterize the mean and covariance of the Gaussians, respectively, from the preceding continuous latents $z_{t-\ell}$. The mean of this probability is characterized by a simultaneously equivariant and invariant architecture. Specifically, we want the transformation of past latent z_{t-1} to be preserved and conveyed to the present latent z_t through one of the S equivariant networks. However, the selection of the equivariant network is controlled invariantly by the state switch s_t. The diagonal covariance should, however, be characterized with an invariant network.

Finally, Gaussian distributions are defined for $p_\theta(x_{n,t} \mid z_{n,t})$ to map $z_{n,t}$ to the observation space $x_{n,t}$:

$$p_\theta(x_t \mid z_t) = \mathrm{Norm}\big(\boldsymbol{\mu}_{\boldsymbol{\theta}}^{\mathbf{x}}(z_t), \boldsymbol{\sigma}^{\mathbf{x}}\mathrm{I}\big), \tag{11}$$

where $\boldsymbol{\mu}_{\boldsymbol{\theta}}^{\mathbf{x}}(\cdot)$ is a nonlinear mapping and $\boldsymbol{\sigma}^{\mathbf{x}}$ denotes the observation noise. $\boldsymbol{\mu}_{\boldsymbol{\theta}}^{\mathbf{x}}(\cdot)$ is characterized by an equivariant network that preserves the transformations between input and latent (The architecture of our networks are described in Supplementary.)

5.2 Inference Model

The posterior distribution of our proposed model $p_\theta(\mathcal{S}, Z \mid X)$ is intractable. Therefore, we employ stochastic variational methods (Hoffman et al., 2013; Ranganath et al., 2013) to learn the parameters of our model, in which the posterior of latents are approximated with a variational distribution $q_\phi(\mathcal{S}, Z)$, by maximizing a lower bound on the likelihood of data, a.k.a. ELBO:

$$\begin{aligned} \mathcal{L}(\theta, \phi) &= \mathbb{E}_{q_\phi(\mathcal{S},Z)}\left[\log \frac{p_\theta(X, \mathcal{S}, Z)}{q_\phi(\mathcal{S}, Z)}\right] \\ &= \log p_\theta(X) - \mathrm{KL}\left(q_\phi(\mathcal{S}, Z) \,\|\, p_\theta(\mathcal{S}, Z|X)\right) \end{aligned} \tag{12}$$

By maximizing this bound with respect to the parameters θ, we learn a generative model that defines a distribution over data $p_\theta(X)$. Similarly, by maximizing it over the parameters ϕ, we perform Bayesian inference.

We assume the following factorized variational distribution for the latents $\mathcal{S}, Z$:

$$q_\phi(\mathcal{S}, Z) = \prod_{n=1}^{N} \prod_{t=1}^{T} q_\phi(s_{n,t}) q_\phi(z_{n,t}), \tag{13}$$

where $q_\phi(z_{n,t}) = \mathrm{Norm}(\mu_\phi^{n,t}, \sigma_\phi^{n,t})$ and the categorical distributions $q_\phi(s_{n,t})$ are approximated from the posteriors $p(s_{n,t}|\tilde{z}_{n,t})$ using the Bayes' rule, where $\tilde{z}_{n,t} \sim q_\phi(z_{n,t})$, to relieve the information loss from

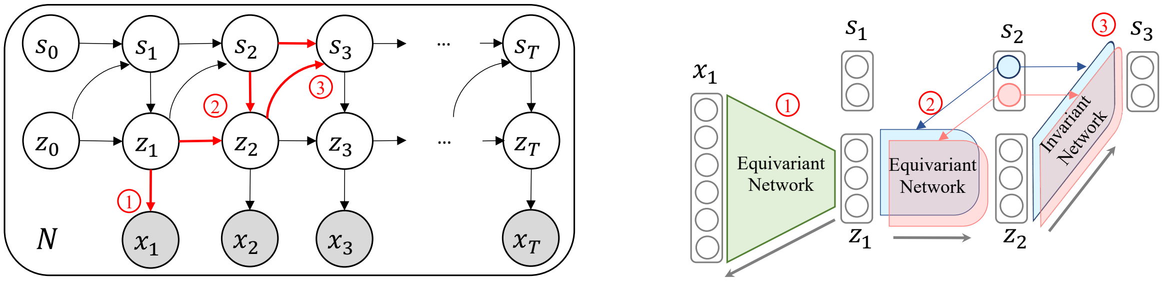


Figure 3: **Left:** Probabilistic graphical model of equivariant deep state space model. **Right:** The generative equivariant and invariant networks are for edges in red. To avoid crowding the figure, we only draw some of the networks in the generative model on the left, but the pattern repeats throughout the chain. The label (1) is the equivariant generative network for constructing x_t using the latent z_t, label (2) is the equivariant generative network informing the current latent z_t of the past latent z_{t-1} governed by the current state s_t, and label (3) is the invariant network influencing future state s_{t+1} using the current latent z_t and state s_t. Arrows show the direction of the forward pass of the networks.

mean-field approximation:

$$\begin{aligned} q_\phi(s_t = s) &\simeq p(s_t = s|\tilde{z}_t) \\ &= \frac{p(s_t = s)p(\tilde{z}_t|s_t = s)}{\sum_s p(s_t = s)p(\tilde{z}_t|s_t = s)} \end{aligned} \tag{14}$$

After defining the variational structures, we insert into equation 12 the generative and variational distributions from equation 8 and equation 13, respectively and derive the following ELBO by some algebraic manipulations.

$$\begin{aligned} &\mathcal{L}_t(\theta, \phi) = \\ &\quad -\mathbb{E}_{q_\phi(z_t)}\Big[\big\|x_t - \boldsymbol{\mu}^{\mathbf{x}}_{\boldsymbol{\theta}}(z_t)\big\|^2_{\mathrm{F}}\Big] - \\ &\quad \mathbb{E}_{q_\phi(s_{t-1})q_\phi(z_{t-1})}\Big[\mathrm{KL}\big(q_\phi(s_t)||p_\theta(s_t|s_{t-1}, z_{t-1})\big)\Big] - \\ &\quad \mathbb{E}_{q_\phi(s_t)q_\phi(z_{t-\ell})}\Big[\mathrm{KL}\big(q_\phi(z_t)\|p_\theta\big(z_t|z_{t-\ell}, s_t\big)\big)\Big], \end{aligned}$$

where the first term corresponds to the reconstruction loss, the second term is the discrete latent loss, and the last term is the continuous latent loss. The ELBO gradients w.r.t. θ and ϕ are estimated using a reparameterized sample from $q_\phi(z_t)$ (Kingma and Welling, 2014), i.e., $z_t = \mu^t_\phi + \sigma^t_\phi \epsilon$, where $\epsilon \sim \mathrm{Norm}(0, \mathrm{I})$, and by enumerating over the possible states in $q_\phi(s_t)$.

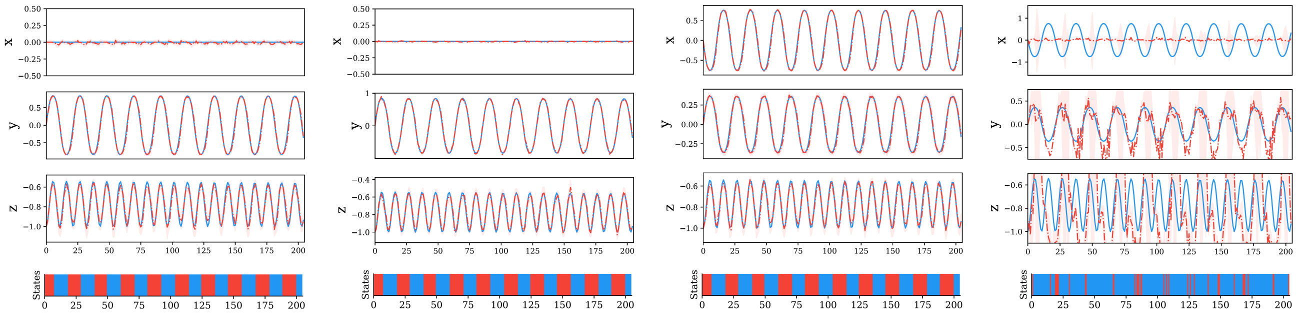
5.2.1 Why do variational distributions preserve equivariance/invariance?

While our generative design is equivariant in essence, in order to have an equivariant framework, this property needs also to be preserved in the inference design. The reconstruction term of ELBO $\big\|x_t - \boldsymbol{\mu}^{\mathbf{x}}_{\boldsymbol{\theta}}(\tilde{z}_t)\big\|_{\mathrm{F}}$, where $\tilde{z}_t \sim q_\phi(z_t)$, in conjunction with the equivariance of $\boldsymbol{\mu}^{\mathbf{x}}_{\boldsymbol{\theta}}(\cdot)$ encourage estimation of equivariant variational parameters (i.e., μ^t_ϕ, σ^t_ϕ) such that the resulting posterior samples $\tilde{z}_t$ go through the same transformation as observed x_t. For the discrete states, estimation of their variational parameters involve computing $p(\tilde{z}_t|s_t = s)$ (see equation 14), which is proportional to $\exp(\|\tilde{z}_t - \boldsymbol{\mu}^s_{\boldsymbol{\theta}}(\tilde{z}_{t-\ell})\|^2_{\mathrm{F}})$ (see equation 10), the Euclidean distance between the posterior $\tilde{z}_t$ and prior mean $\boldsymbol{\mu}^s_{\boldsymbol{\theta}}(\tilde{z}_{t-\ell})$, which stays invariant under the SO(3) equivariance of $\boldsymbol{\mu}^s_{\boldsymbol{\theta}}(\cdot)$

6 Experiments

6.1 Performance Metric

To quantify the performance of our dynamical generative model, we follow the approach of Farnoosh et al. (2021); Linderman et al. (2017) and compute the temporal predictive error on a test set trajectory in a rolling manner. We predict the next time point on a test set $\hat{x}_{t+1}$ using the generative model learned on the train set: $\hat{x}_{t+1} = \mu^{\mathbf{x}}_\theta(\hat{z}_{t+1})$, where $\hat{z}_{t+1} \sim p(\hat{z}_{t+1}|z_{t+1-\ell}, \hat{s}_{t+1})$ and $\hat{s}_{t+1} \sim p(\hat{s}_{t+1}|s_t, z_t)$. We then run inference on x_{t+1}, the actual observation at $t+1$, to obtain z_{t+1} and s_{t+1}, and add them to the historical data for prediction of the next time point $\hat{x}_{t+2}$ in a similar manner. Repeating these steps to make predictions in a rolling manner across a test set, we report their normalized root-mean-square error (NRMSE%). The generative model remains fixed during the entire prediction procedure. It is worthwhile to note that the test set prediction NRMSE% is related to the expected *negative test-set log-likelihood* for our case of Gaussian distributions with a multiplicative/additive constant.



(a) EqDDM (b) DSARF (c) EqDDM (rotated) (d) DSARF (rotated)

Figure 4: Test set predictions (red curve) along with ground-truth (blue curve) for the pendulum experiment. The figures in the bottom row encode the inferred states: clockwise rotation (blue) and anticlockwise rotation (red). (a), (c) EqDDM successfully generalized to both the original and rotated test sets. (b), (d) DSARF performed similarly on the original test set but entirely failed on the rotated test set. This is expected as DSARF (and other baselines) are not aware of the symmetries in this dataset and overfit on the train set trajectory. As shown in the bottom row, EqDDM and DSARF decomposed the pendulum motion into two states of clockwise and anticlockwise rotation. While these states stayed unchanged for EqDDM in the rotated test set, DSARF failed to preserve its states. The red shaded regions indicate uncertainty intervals.

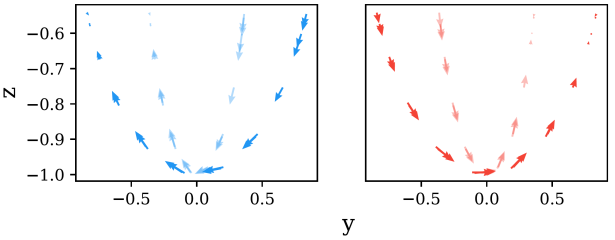


Figure 5: Dynamical trajectories of each state in the pendulum dataset (projected on the y-z plane) visualized for the original test set (solid colors) and rotated test set (shaded colors). These trajectories are computed from the generative model of EqDDM and confirm its generalizability and our interpretation of clockwise (blue) and anticlockwise (red) rotations.

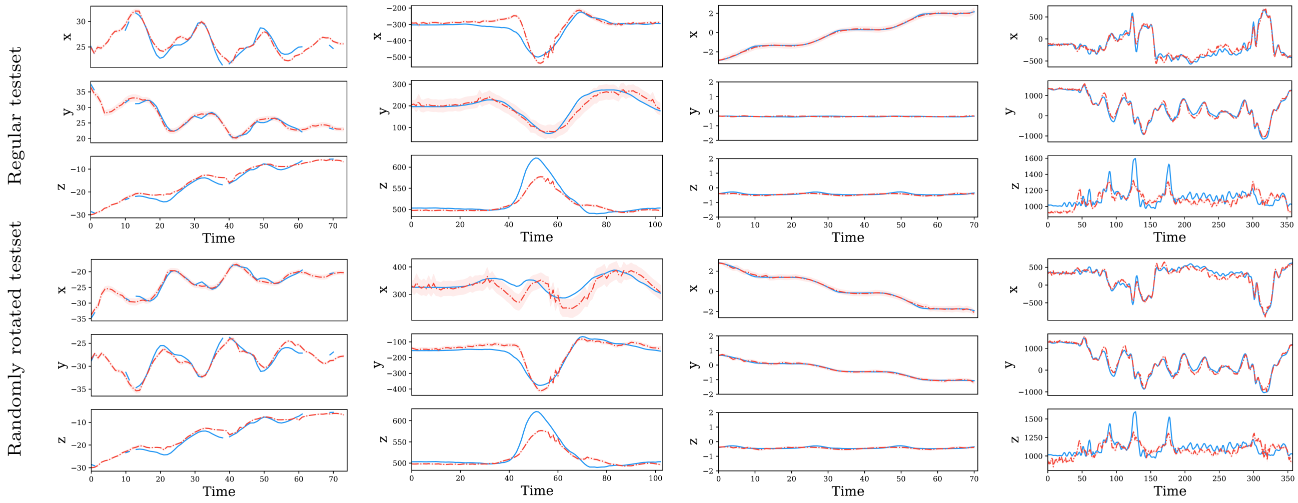
6.2 Comparison Baselines

We compared our model with three state-of-the-art Bayesian switching dynamical models, deep switching autoregressive factorization (DSARF) (Farnoosh et al., 2021), recurrent switching linear dynamical systems (rSLDS) (Nassar et al., 2019), and switching linear dynamical systems (SLDS) (Fox et al., 2009), a state-of-the-art deep state-space model, recurrent Kalman networks (RKN) (Becker et al., 2019), and a deep forecasting model, long- and short-term time-series network (LSTNet) (Lai et al., 2018) throughout the experiments. We also compared our model with the *non-equivariant* version of our EqDDM model, in which all neworks were replaced with regular MLPs, trained with augmented dataset (see DDM+Aug. in table 1). Finally, we implemented and compared with the equivariant version of Hamiltonian neural network of Greydanus et al. (2019) (see EqHNN in table. 1).

6.3 Datasets

Pendulum: We simulated a pendulum system on the y-z plane for $T = 410$ time points and recorded its 3D coordinates. We trained the models on the first half and tested on the second half. **Bat flight:** This dataset (Bergou et al., 2015) includes 3D coordinates of 34 joints on a bat skeleton recorded for T between 33 to 87 time points (every 165 msec) during a landing/falling maneuver for 10 experimental runs with 32.55% missing values. We kept two runs for the test. **Golf:** This dataset from CMU MoCap[1] includes 30 trials of motion recordings from a subject while performing typical actions in a golf game. We kept two trials for the test. **Walk:** This MoCap dataset contains 3D motion recordings from a subject for 34 trials of walking/running. We kept two trials for the test. **Salsa dance:** This MoCap dataset contains 3D coordinates of 19 joints recorded for T between 200 to 571 time points for 15 trials of salsa dancing. We kept one trial for the test and only used the woman dancer data. **Rotated test sets:** Additionally, for each dataset we formed test sets by randomly rotating the original test set (about z-axis) for 10 different angles. These test sets are prefixed by $\mathcal{R}$.

[1] http://mocap.cs.cmu.edu/



(b) bat (c) golf (d) walk (e) salsa

Figure 6: Predictions of the regular (top row) and rotated test sets (bottom row) for a sample joint from each of the real datasets. Test set predictions are shown as red curves along with ground-truth shown as blue curve. EqDDM has successfully generalized on the rotated test sets. Also, note that our model fills in the missing values in the bat dataset. The red shaded regions indicate uncertainty intervals.

Table 1: Comparison of prediction error (NRMSE%) on the regular and rotated ($\mathcal{R}$) test sets. Our model outperforms the baselines particularly on the rotated test sets in which the baselines completely fail.

Dataset	EqDDM	DSARF	rSLDS	SLDS	RKN	LSTNet	DDM+Aug.	EqHNN
Pen.	5.13	**4.66**	24.29	27.73	7.48	7.33	–	–
$\mathcal{R}$Pen.	**5.29**	72.37	89.88	88.08	78.22	79.58	19.37	15.04
Bat	**7.61**	8.82	11.39	12.26	19.02	18.75	–	–
$\mathcal{R}$Bat	**7.40**	47.89	57.46	46.93	66.55	69.82	17.03	20.82
Golf	**8.48**	10.92	10.60	13.99	12.95	18.22	–	–
$\mathcal{R}$Golf	**9.40**	29.06	28.73	36.48	32.09	45.01	16.98	12.81
Walk	**3.86**	4.72	12.85	13.31	13.88	8.70	–	–
$\mathcal{R}$Walk	**4.53**	37.21	43.71	41.29	38.62	42.94	10.44	8.03
Salsa	11.27	**10.94**	13.25	13.09	13.38	11.91	–	–
$\mathcal{R}$Salsa	**11.42**	16.78	18.93	19.58	18.97	16.83	15.39	15.17

Best results are highlighted in bold fonts.

6.4 Experimental results

Experimental settings: We set the number of states $S = 2$, temporal lags $\ell = \{1, 2\}$, the latent dimension $K = 3$ for pendulum and $K = 6$ for other experiments, and set the network dimensions (i.e., hidden layers) accordingly to match the number of generative parameters among comparison methods for a fair evaluation (see the Supplementary for details).

Results: We have summarized our experimental results in table 1 and figures 4, 5, and 6. For the pendulum experiment, as depicted in figure 4, EqDDM successfully generalized to both the original and rotated test sets. EqDDM performed at par with DSARF on the original test set, however, all the baselines (including DSARF) completely failed on the rotated test set (see table 1). This is expected as the baselines are not aware of the symmetries in this dataset and overfit on the train set trajectory. As shown in the bottom row of figure 4, EqDDM (and DSARF) decomposed the pendulum motion into two states: clockwise and anticlockwise rotation. While these states stayed unchanged for EqDDM in the rotated test set, DSARF failed to preserve its states. We have also computed the dynamical trajectories of each state purely from our learned generative model and visualized them for both test sets in figure 5, which confirms our interpretation of each state. For the bat, golf, and walk datasets, EqDDM consistently outperformed all the baselines and preserved its performance on the rotated test sets by exploiting the symmetries in the datasets, however, competing baselines completely failed to generalize. As reported in table 1, on the rotated test sets, EqDDM achieved 7.40%, 9.40%, and 4.53%, respectively, while the best performing non-equivariant baseline without data augmentation only achieved 47.89%, 28.73%, and 37.21%, respectively. For the salsa dataset, EqDDM closely follows DSARF on the original test set with 11.27% versus 10.94%, and outperforms the other baselines. EqDDM preserves its performance on the rotated test set and surpasses all the baselines. In contrast to the other datasets in which the baselines completely failed to generalize on the rotated test set, for salsa dance these baselines achieve an acceptable performance. It is because the motions in the salsa dance are diverse enough for the models to see and memorize various rotations. However, note that EqDDM still significantly performs better than these baselines due to its inherent SO(3) equivariant design. We have visualized predictions of the regular and rotated test sets for a sample joint from each of the real datasets in figure 6 which confirms the generalization capacity of EqDDM. Also, note that our model fills in the missing values in the bat dataset. As for the EqHNN model and non-equivariant version our model trained with augmented dataset (DDM+Aug.), it is intuitive to see that data augmentation helps with generalization to some extent. Nevertheless, the main shortcoming is that generalization is not guaranteed. Unlike the discrete transformation group with finite cardinality, data augmentation for the continuous groups, such as SO(3), requires introducing many transformations sampled from the continuous group.

7 Conclusion

We proposed an SO(3) equivariant deep dynamical model for motion prediction. Our model is equipped with equivariant/invariant networks that preserve the rotational symmetry. We showcased the generalization of our model to arbitrary rotations of various motion data.

Acknowledgements

We would like to thank Dr. Amirreza Farnoosh for providing extensive insight and expertise into this work.

References

Azari, B. and Erdogmus, D. (2021). Circular-symmetric correlation layer based on fft. *arXiv preprint arXiv:2107.12480.*

Becker, P., Pandya, H., Gebhardt, G., Zhao, C., Taylor, C. J., and Neumann, G. (2019). Recurrent kalman networks: Factorized inference in high-dimensional deep feature spaces. In *International Conference on Machine Learning*, pages 544–552.

Bergou, A. J., Swartz, S. M., Vejdani, H., Riskin, D. K., Reimnitz, L., Taubin, G., and Breuer, K. S. (2015). Falling with style: bats perform complex aerial rotations by adjusting wing inertia. *PLoS Biol*, 13(11):e1002297.

Cohen, T. and Welling, M. (2016). Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR.

Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. (2018). Spherical cnns. In *International Conference on Learning Representations.*

Cohen, T. S., Geiger, M., and Weiler, M. (2019). A general theory of equivariant cnns on homogeneous spaces. *In Advances in Neural Information Processing Systems (NeurIPS)*, 32.

Cohen, T. S. and Welling, M. (2017). Steerable cnns. In *International Conference on Learning Representations.*

Dieleman, S., De Fauw, J., and Kavukcuoglu, K. (2016). Exploiting cyclic symmetry in convolutional neural networks. In *International conference on machine learning*, pages 1889–1898. PMLR.

Dieleman, S., Willett, K. W., and Dambre, J. (2015). Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly notices of the royal astronomical society*, 450(2):1441–1459.

Dym, N. and Maron, H. (2021). On the universality of rotation equivariant point cloud networks. In *International Conference on Learning Representations.*

Esteves, C., Allen-Blanchette, C., Makadia, A., and Daniilidis, K. (2018). Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68.

Farnoosh, A., Azari, B., and Ostadabbas, S. (2021). Deep switching auto-regressive factorization: Application to time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7394–7403.

Farnoosh, A. and Ostadabbas, S. (2021). Dynamical deep generative latent modeling of 3d skeletal motion. *arXiv preprint arXiv:2106.10393*.

Feige, I. (2019). Invariant-equivariant representation learning for multi-class data. In *International Conference on Machine Learning*, pages 1882–1891. PMLR.

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. (2020). Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3165–3176. PMLR.

Finzi, M., Welling, M., and Wilson, A. G. G. (2021). A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3318–3328. PMLR.

Fox, E., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2009). Nonparametric bayesian learning of switching linear dynamical systems. In *Advances in neural information processing systems*, pages 457–464.

Fraccaro, M., Kamronn, S., Paquet, U., and Winther, O. (2017). A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems*, pages 3601–3610.

Gao, X., Hu, W., and Qi, G.-J. (2020). Graphter: Unsupervised learning of graph transformation equivariant representations via auto-encoding node-wise transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172.

Gens, R. and Domingos, P. M. (2014). Deep symmetry networks. *Advances in neural information processing systems*, 27:2537–2545.

Gilmore, R. (2006). *Lie Groups, Lie Algebras, and Some of Their Applications*. Courier Corporation.

Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32.

Guttenberg, N., Virgo, N., Witkowski, O., Aoki, H., and Kanai, R. (2016). Permutation-equivariant neural networks applied to dynamics prediction. *arXiv preprint arXiv:1612.04530*.

Hazewinkel, M., Gubareni, N., and Kirichenko, V. V. (2004). *Algebras, rings and modules*, volume 1. Springer Science & Business Media.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Karl, M., Soelch, M., Bayer, J., and van der Smagt, P. (2017). Deep variational bayes filters: Unsupervised learning of state space models from raw data. *stat*, 1050:3.

Keriven, N. and Peyré, G. (2019). Universal invariant and equivariant graph neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *stat*, 1050:1.

Köhler, J., Klein, L., and Noé, F. (2020). Equivariant flows: exact likelihood generative learning for symmetric densities. In *International Conference on Machine Learning*, pages 5361–5370. PMLR.

Kondor, R. and Trivedi, S. (2018). On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pages 2747–2755.

Krishnan, R. G., Shalit, U., and Sontag, D. (2017). Structured inference networks for nonlinear state space models. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. (2018). Modeling long-and short-term temporal patterns with deep neural networks. In *ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104.

Linderman, S., Johnson, M., Miller, A., Adams, R., Blei, D., and Paninski, L. (2017). Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial Intelligence and Statistics*, pages 914–922.

Maron, H., Ben-Hamu, H., Shamir, N., and Lipman, Y. (2019). Invariant and equivariant graph networks. In *International Conference on Learning Representations*.

Maron, H., Litany, O., Chechik, G., and Fetaya, E. (2020). On learning sets of symmetric elements. In *International Conference on Machine Learning*, pages 6734–6744. PMLR.

Nassar, J., Linderman, S., Bugallo, M., and Park, I. (2019). Tree-structured recurrent switching linear dynamical systems for multi-scale modeling. In *International Conference on Learning Representations (ICLR)*.

Olah, C. (2014). Groups and group convolutions.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch.

Qi, G.-J., Zhang, L., Chen, C. W., and Tian, Q. (2019). Avt: Unsupervised learning of transformation equivariant representations by autoencoding variational transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8130–8139.

Ranganath, R., Wang, C., David, B., and Xing, E. (2013). An adaptive learning rate for stochastic variational inference. In *International Conference on Machine Learning*, pages 298–306.

Ravanbakhsh, S., Schneider, J., and Poczos, B. (2016). Deep learning with sets and point clouds. *arXiv preprint arXiv:1611.04500*.

Ravanbakhsh, S., Schneider, J., and Poczos, B. (2017). Equivariance through parameter-sharing. In *International Conference on Machine Learning*, pages 2892–2901. PMLR.

Satorras, V. G., Hoogeboom, E., and Welling, M. (2021). E (n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*.

Shamsabardeh, M., Azari, B., and Martínez-López, B. (2021). Prrs outbreak prediction via deep switching auto-regressive factorization modeling. *arXiv preprint arXiv:2110.03147*.

Walters, R., Li, J., and Yu, R. (2021). Trajectory prediction using equivariant continuous convolution. In *International Conference on Learning Representations*.

Wang, R., Walters, R., and Yu, R. (2020). Incorporating symmetry into deep dynamics models for improved generalization. In *International Conference on Learning Representations*.

Watter, M., Springenberg, J., Boedecker, J., and Riedmiller, M. (2015). Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pages 2746–2754.

Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. (2018). 3d steerable cnns: learning rotationally equivariant features in volumetric data. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10402–10413.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. (2017). Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5028–5037.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. (2017). Deep sets. In *Advances in neural information processing systems*, pages 3391–3401.

Supplementary Material: Equivariant Deep Dynamical Model for Motion Prediction

A Background on Group Theory

In this paper, we use numerous concepts in abstract algebra, group theory, and representation theory. We provide a wider range of details on the notations and definitions we employed in the paper.

Symmetry: A symmetry is a set of The transformations should preserve the properties of the structure. Generally, it is presumed that the transformations must be invertible, i.e., for each transformation there is another transformation, called its inverse, which reverses its effect. Symmetry is thus can be stated mathematically as an operator acting on an object, are modeled by **Groups**.

Group: Let G be a non-empty set with a binary operation defined as $\circ : G \times G \mapsto G$. We call the pair $(G; \circ)$ a group if it has the following properties: G is closed under its binary operation (Closure), the group operation is associative –i.e., $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$ for $g_1, g_2, g_3 \in G$ (Associativity axiom), there exists an identity $e \in G$ such that $g \circ e = e \circ g = g$ for all $g \in G$ (Identity axiom), every element $g \in G$ has an inverse $g^{-1} \in G$, such that $g \circ g^{-1} = g^{-1} \circ g = e$ (Inverse axiom).

Subgroup: A non-empty subset H of G is called a subgroup, if H is a group equipped with the same binary operation of as in G. We show this as $H \leq G$. H is called a proper subgroup of if $H \neq G$ and we show it as $H < G$.

Group action: We say a group G *acts* on a set $\mathcal{X}$ if there exist a map $\phi : G \times \mathcal{X} \to \mathcal{X}$ such that: (1) $\phi(e, x) = x$, where e is the identity element of G, and (ii) $\phi(g, \phi(h, x)) = \phi(gh, x)$ for all $g, h \in G$ and $x \in \mathcal{X}$. In this case, G is called a transformation group, $\mathcal{X}$ is a called a G-set, and ϕ is called the group action.

Lie group and infinitesimal generator: A Lie group G is a smooth manifold equipped with the structure of a group such that the group operation and inverse-assigning operation are smooth functions (Gilmore, 2006). This means that the group operation and inverse operation are continuous on the manifold, and they can be expressed in terms of the coordinates. The manifold is locally represented by a chart mapping to an underlying Euclidean space $\mathbb{R}^D$, where D is the dimensionality of the manifold. Furthermore, the chart map is defined in such a way that it associates the identity element in the group with the origin of Euclidean space. Elements of the Lie group can *act* as a transformation on the basis an n-dimensional vector space known as the *geometric space*, and change the coordinates of elements accordingly. We analyze Lie groups in terms of their infinitesimal generators which are the derivative of the group elements with respect to its D underlying parameters at the identity. These infinitesimal generators are the basis for a new vector space, called the *Lie Algebra*.

Lie algebra: Lie algebra, denoted as $\mathfrak{g}$, is the first order infinitesimal approximation to a Lie group, and can be interpreted as a tangent space at the identity $\mathfrak{g} \coloneqq T_{\text{id}}G \subseteq \mathbb{R}^{n \times n}$. In general, an algebra over a field is a vector space equipped with a bilinear product. Thus, it consists of a set together with operations of multiplication and addition and scalar multiplication by elements of a field and satisfies the axioms of vector space and bilinear form (Hazewinkel et al., 2004).

Lie algebra representation: Each element of the Lie group can also be understood as a transformation on some other vector space such as what we call a Lie group of *transformations* (A N dimensional vector space, geometric space G_N). Every point has its own coordinate. We have some basis e_μ, and each point in the vector space has its coordinate which will just be its component relative to the e_μ basis. Each element of the Lie group represent a transformation of the basis which changes the coordinate of every point. The effect of the group elements on the elements of the underlying geometric space is describe by a function denoted as $y = f(\mu, x)$. every Lie group is its own Lie group of transformation – geometric space G_N.

B Useful Tensor Manipulations

If A is an $n \times p$ matrix and B an $m \times q$ matrix, the $mn \times pq$ matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1p}B \\ a_{21}B & a_{22}B & \dots & a_{2p}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \dots & a_{np}B \end{bmatrix}$$

is the Kronecker product of A and B. It is also called the tensor product. The $vec(\cdot)$ operator creates a column vector from a matrix A by stacking the column vectors of $A = [a_1, a_2, \dots, a_n]$ below one another as:

$$vec(A) = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

Kronecker Product and the $\boldsymbol{vec}(\cdot)$ Operator: For given matrices A, B, and X we have:

$$vec(AXB) = (B^\top \otimes A)vec(X). \tag{15}$$

C Network architecture and experimental settings

The network architectures for $\boldsymbol{\pi}^s_{\boldsymbol{\theta}}$, $\boldsymbol{\mu}^s_{\boldsymbol{\theta}}$, $\boldsymbol{\sigma}^s_{\boldsymbol{\theta}}$, and $\boldsymbol{\mu}^{\mathbf{x}}_{\boldsymbol{\theta}}$ are provided in table 2. For the pendulum experiment, we set the number of states $S = 2$, latent dimension $K = 3$, latent representation $U_{\mathrm{z}} = T_1$, state representation $U_{\mathrm{s}} = 2T_0$, observation representation $U_{\mathrm{x}} = T_1$, $\boldsymbol{\pi}^s_{\boldsymbol{\theta}}$ hidden representation $U_{h_s} = 3T_0 \oplus 2T_1$, $\boldsymbol{\mu}^s_{\boldsymbol{\theta}}$ and $\boldsymbol{\sigma}^s_{\boldsymbol{\theta}}$ hidden representation $U_{h_z} = 3T_0 \oplus T_1 \oplus T_2$, and $\boldsymbol{\mu}^{\mathbf{x}}_{\boldsymbol{\theta}}$ hidden representation $U_{h_x} = 3T_0 \oplus T_1$. For other experiments, we set the number of states $S = 2$, latent dimension $K = 6$, latent representation $U_{\mathrm{z}} = 3T_0 \oplus T_1$, state representation $U_{\mathrm{s}} = 2T_0$, observation representation $U_{\mathrm{x}} = D\,T_1$, $\boldsymbol{\pi}^s_{\boldsymbol{\theta}}$ hidden representation $U_{h_s} = 3T_0 \oplus 2T_1 \oplus T_2$, $\boldsymbol{\mu}^s_{\boldsymbol{\theta}}$ and $\boldsymbol{\sigma}^s_{\boldsymbol{\theta}}$ hidden representation $U_{h_z} = 3T_0 \oplus 3T_1 \oplus 2T_2$, and $\boldsymbol{\mu}^{\mathbf{x}}_{\boldsymbol{\theta}}$ hidden representation $U_{h_x} = 3T_0 \oplus 3T_1$. This setting of hidden representations and dimensions roughly match the generative parameter count of EqDDM with that of the baselines for a fair comparison. We set the latent dimension and number of states (if applicable) for the baselines accordingly.

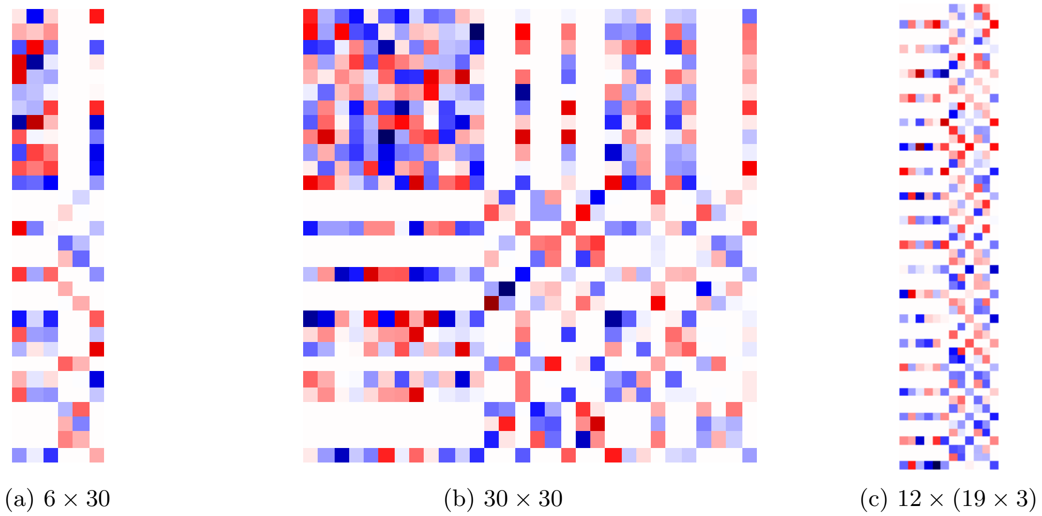
We have visualized the three equivariant linear layers with dimensions $K \times 5K$, $5K \times 5K$, and $2K \times 3D$, respectively in figure 7 for $K = 6$ and $D = 19$.

Table 2: Network architectures for the nonlinear mappings in EqDDM. ELL: Equivariant Linear Layer, ILL: Invariant Linear Layer, ENL: Equivariant Nonlinearity

Network	$\boldsymbol{\pi}^s_{\boldsymbol{\theta}} : \mathbb{R}^K \to \mathbb{R}^S$	$\boldsymbol{\mu}^s_{\boldsymbol{\theta}}$, $\boldsymbol{\sigma}^s_{\boldsymbol{\theta}} : \mathbb{R}^{\lvert\ell\rvert \times K} \to \mathbb{R}^{K,K}$	$\boldsymbol{\mu}^{\mathbf{x}}_{\boldsymbol{\theta}} : \mathbb{R}^K \to \mathbb{R}^{3D}$
Input	$z_{t-1} \in \mathbb{R}^K$	$z_{t-\ell} \in \mathbb{R}^{\lvert\ell\rvert \times K}$	$z_t \in \mathbb{R}^K$
1	ELL $K \times 3K$ ENL	ELL $\lvert\ell\rvert \times K \times 5K$ ENL	ELL $K \times 2K$ ENL
2	ELL $3K \times 3K$ ENL	ELL $\lvert\ell\rvert \times 5K \times 5K$ ENL	ELL $2K \times 2K$ ENL
3	ILL $3K \times S$ Softmax	AvgPool($\lvert\ell\rvert$)	ELL $2K \times 2K$ ENL
4		ELL $5K \times 5K$ ENL	ELL $2K \times 3D$
5		ELL $5K \times K$ ILL $5K \times K$	

D Computational Resources

We implemented EqDDM with PyTorch v1.8 (Paszke et al., 2017) and used the Adam optimizer (Kingma and Ba, 2014) with learning rate of 0.01. We initialized all the parameters randomly. We performed all the experiments



(a) 6×30

(b) 30×30

(c) $12 \times (19 \times 3)$

Figure 7: Visual representations of three equivariant linear layers for dimensions (a) 6×30, (b) 30×30, and (c) $12 \times (19 \times 3)$. For each representation, similar colors denote shared parameters.

on an Intel Core i9 CPU@3.6GHz with 32 GB of RAM. Per-epoch training time varied from 200 msec in smaller datasets to 1 sec in larger experiments and 300 epochs sufficed for all the experiments.

E Impact

The goal of this paper is to design a deep structured architecture for generative modeling of dynamic data by adopting the formalism of group theory. We do not expect that the developed model has an immediate societal impact or poses any direct risks. However, because the problem of deep generative modeling concerns designing a probabilistic model that can generate realistic data, it may be misused in producing fake realistic data.