Topographic Coupled Oscillator Networks Learn Transformations as Traveling Waves

T. Anderson Keller
UvA-Bosch Delta Lab
University of Amsterdam
Amsterdam, NL 1098XH

Max Welling
UvA-Bosch Delta Lab
University of Amsterdam
Amsterdam, NL 1098XH

Abstract

Structured representations, such as those of convolutional and other equivariant neural networks, have shown increased generalization performance and data efficiency when the integrated structure accurately represents the symmetry transformations present in the data. However, in order to impose such structure, most methods require explicit knowledge of the underlying transformation group which is infeasible for many real-world transformations. In this work, we suggest an extremely general inductive bias – that of traveling waves across features space – may be capable of inducing approximately equivariant structure for arbitrary observed transformations in an unsupervised manner. To demonstrate this, we leverage a biologically relevant dynamical system known to exhibit traveling waves, specifically a network of topographically coupled oscillators, and show that when integrated into a modern deep neural network architecture, such a system does indeed learn to represent observed symmetry transformations as traveling waves in the latent space. The approximate equivariance of the model is verified by artificially inducing waves in the latent space and subsequently decoding to visualize transforming test sequences, implying commutativity of the feature extractor and the transformations in the input and feature spaces. We further demonstrate that our model yields performance competitive with state of the art on a suite of sequence classification and forecasting benchmarks while simultaneously converging more consistently and requiring significantly fewer parameters than its globally coupled counterpart.

1 Motivation

Structured representations are an important component of the recent success of otherwise exceptionally flexible artificial neural networks. A prominent class of models with integrated structure, known as equivariant neural networks, guarantee that the outputs of a model obey known transformation laws for given transformations of the input. When appropriately leveraged, such imposed structure ensures that the model's performance remains consistent between transformations of unseen future examples. Theoretically, such models are known to increase generalization performance and data efficiency when the integrated structure accurately represents the symmetry transformations present in the data. However, in practice, such performance benefits have only been maximally realized in settings where the symmetries are exact and known mathematically, such as with coordinate symmetries for 3-dimensional inputs [50, 33], or rotational and translation symmetries in image tasks [14, 54, 55, 4]. Nevertheless, in these particular settings, the performance benefits have been highly significant, leading to the near universality of convolutional neural networks for image processing, and similarly leading to the adoption of equivarant structure in many of the foremost empirical successes of deep learning models to date [33]. Such a profoundly successful inductive bias naturally raises the question: How can the benefits of structured representations be realized for the diverse set of real-world transformations which are not as easily encoded a priori?

Preprint. Under review.

In searching for an answer, we consider the general properties of structured representations which we may be able to impose as inductive 'structure' biases. Due to their relative simplicity and empirical success, we turn to convolutional and group equivariant neural networks for inspiration. We see that at a high level, such networks satisfy a key criterion when their input is transformed: their output activations perform a predictable magnitude preserving shift across feature dimensions. For example, when an input image is translated, the activations of a convolutional neural network shift in unison. Similarly, when an input to a rotation equivariant group convolutional network is spatially rotated, activations are cyclically shifted through the equivariant channel dimension of the feature space.

In order to maintain maximum flexibility with respect to the set of learnable transformations, we wish our inductive biases to enforce such a shift in as flexible of a computational model as possible. One of the simplest mechanisms which satisfies this property is the traveling wave. Mathematically, a traveling wave can be written as a function whose response at a given location x over time t is equivalent to it's response at another location, but shifted in time: $y(x, t) = f(x - vt)$. In dynamical systems, wave equations are solutions to one of the most general forms of a local conservation law – the continuity equation: ($\frac{\partial \rho}{\partial t} = -\nabla \cdot \mathbf{j}$) where $\mathbf{j}$ is the flux of our quantity of interest ρ. Such a relationship gives credence to the idea that if a model were biased towards generating traveling waves, such waves may serve as latent space operators to satisfy our desideratum described above – they naturally shift and conserve activation between features over time for observed input transformations.

One system which is known to exhibit traveling waves is a network of locally coupled oscillators. In fact, when the number of oscillators is taken to the continuum limit, the wave equation emerges naturally as a description of the dynamics of the system [51]. Interestingly, such networks are additionally among the most commonly used to model the activity of biological assemblies of neurons, and the traveling waves they exhibit have recently been demonstrated to exhibit many non-trivial similarities with traveling waves observed to propagate across the surface of the cortex [17].

Recent work [45] introduced an abstract version of a coupled oscillator recurrent neural network shown to be capable of learning very long time dependencies. In this work, we adopt this model and adapt it to our goal of learning structured representations by introducing topographic constraints on the recurrent connections, biasing it towards exhibiting traveling waves, and further extending it to the unsupervised autoregressive setting to allow for sequence generation. Empirically, we demonstrate that this model indeed learns to generate traveling waves, and further that such waves can be seen as approximately equivariant latent operators on the hidden state of the network in direct correspondence with the observed input transformation. We further show preliminary results indicating that our topographically constrained model converges more consistently than its globally coupled counterpart, and that this effect is exacerbated with increased dimension. Taken together with the dramatically reduced parameter count, these findings suggest topographic coupling may be an important option for scalability of such systems. Finally, we demonstrate that such a topographically coupled oscillator network is additionally a flexible and powerful computational mechanism, yielding competitive performance with state of the art models on a suite of sequence classification and dynamics modeling benchmarks.

2 Background

Traveling Waves Traveling waves are known to exist at a diversity of regions and scales across the biological cortex [39]. Although such waves were originally only measured in anesthetized subjects, improved multi-channel recording and analysis techniques have recently demonstrated propagating wave activity in conscious subjects as well, originating from both external stimuli and internal 'spontaneous' recurrent connections [49, 41, 39]. It has been suggested that they form the basis of alpha and theta oscillations [59] and may serve to integrate information across space and time [49]. However, their exact computational role is still debated, and many hypotheses have been put forth including that they may encode motion [32], modulate information transfer [7], facilitate predictive coding [24, 1], lower the threshold for detection of weak stimuli [18], serve as a short term memory [36, 39], and as a mechanism for the formation of long-term memories during sleep [40].

A well known model of traveling waves from the neuroscience literature is based upon a dynamical system of topographically coupled oscillators [20, 22, 38, 21, 17, 52]. Such models have been demonstrated to yield traveling waves which appear to uniquely agree with human cortical traveling waves in a variety of dimensions, including numerous models which implicate lateral topographic connections as the medium by which they propagate [49, 17]. However, to the best of our knowledge, such models have not yet explicitly been tested for their computational ability when integrated with deep neural networks, nor explicitly for their ability to learn symmetry transformations. From a computational point of view, the study of waves as computational elements has uncovered multiple mechanisms by which traveling and colliding waves can be used to perform boolean computations and ultimately form the basic components necessary to build a universal computer [31, 25].

Structured Symmetry Representations The idea of symmetries, invariance, and equivariance are prevalent throughout physics, neuroscience, and machine learning. At a high level, a symmetry transformation is a transformation which leaves some underlying quality unchanged. For example, convolutional neural networks rely on the assumed translation symmetry of images – translating an object in an image may alter the individual pixel values substantially, but the underlying classification of the image should remain the same.

In neuroscience, it has long been suggested that as one progresses deeper in the visual hierarchy, neurons get progressively better at object detection and become more invariant to arbitrary nuisance transformations [48]. Indeed, models trained with rotation invariance have recently achieved state of the art similarity scores with area V4 in the brainscore competition [6], suggesting that correctly handling symmetry transformations may also be an important inductive bias in the brain. Further, recent studies have demonstrated remarkable similarities between a type of structured representation (namely a disentangled representation) and activations of individual neurons in the brain [28, 27].

In machine learning, one way symmetries have been incorporated into neural networks is through equivariant architectures such as group equivariant convolutional neural networks [14]. Formally, an equivariant map f is defined as one which commutes with the transformation: $f(\tau_\rho[\mathbf{u}]) = \Gamma_\rho[f(\mathbf{u})]$, where τ and Γ are the representations of the action of the group element ρ on the input and output spaces respectively. In the simplest setting, as mentioned in the introduction, the activations of a rotation equivariant neural network will be seen to cyclically shift within the equivariant feature dimension when the input and feature maps are acted upon by the appropriate transformation τ (e.g. spatial rotation). Interestingly, Γ can thus equivalently be viewed as a magnitude preserving shift of activation over a time sequence, satisfying the definition of a traveling wave between feature maps. Some recent work leveraging this connection to learn equivariant maps includes the Topographic VAE [35], where the prior over latent variable is explicitly designed to encourage a 'roll' of activations through feature space for observed sequences. Recent theoretical work has further demonstrated that a similar carefully chosen latent operator (called the shift operator) is linearly isomorphic to a surprisingly large set of image transformations including pixel translations and rotations [10]. The authors use this theory to prove that by enforcing their shift operator in latent space, they can learn equivariance with respect to any finite cyclic group of affine transformations using a simple linear autoencoder.

In this work, we leverage this intuition to propose that a neural network architecture prone to exhibiting traveling waves may have the appropriate bias towards spatio-temporal structure necessary to facilitate learning flexible structured representations for diverse real world transformations.

3 Topographic Coupled Oscillator Networks

In the following section we introduce a deep neural network architecture which facilitates traveling waves while simultaneously retaining the ability to perform flexible differentiable computation. To achieve this we leverage ideas from two recent distinct lines of work from the computational neuroscience and machine learning communities. Specifically, we consider the Coupled Oscillatory Recurrent Neural Network (coRNN) recently introduced by [45] as a powerful sequence processing model, and propose that the constraints necessary for traveling waves outlined in [17] may abstractly be incorporated into such a model with trainable parameters to allow for diverse hidden state dynamics.

Coupled Oscillatory Recurrent Neural Networks In [45] the authors propose to solve the Exploding and Vanishing Gradient Problem (EVGP) in recurrent neural networks by defining a recurrent neural network with hidden state dynamics given by the parameterized equations of a system of coupled, damped, and driven oscillators. Explicitly, the hidden state of the recurrent neural network $\mathbf{x}$ is updated by solving the following second order partial differential equation:

$$\ddot{\mathbf{x}} = \sigma\left(\mathbf{W}_x\mathbf{x} + \mathbf{W}_{\dot{x}}\dot{\mathbf{x}} + \mathbf{V}\mathbf{u} + \mathbf{b}\right) - \gamma\mathbf{x} - \alpha\dot{\mathbf{x}} \tag{1}$$

Where $\frac{\partial \mathbf{x}}{\partial t} = \dot{\mathbf{x}}$, $\frac{\partial^2 \mathbf{x}}{\partial t^2} = \ddot{\mathbf{x}}$ are the first and second derivatives of the hidden state with respect to time, and $\mathbf{u}$ denotes the (potentially externally embedded) input at each time step. The terms $\mathbf{W}_x\mathbf{x}$, $\mathbf{W}_{\dot{x}}\dot{\mathbf{x}}$, and $\mathbf{Vu}$ can then be interpreted as the coupling, damping, and driving terms respectively. Finally, σ is a nonlinear activation function such as the hyperbolic tangent, and γ & α are scalar variables which can be fixed or learned in combination with the above matrices.

In practice, the above differential equation can discretized and integrated numerically using an IMEX (implicit-explicit) discretization scheme shown to preserve the desirable bounds of the continuous system. Such a discretization can be achieved by first introducing a 'velocity' variable $\mathbf{v} = \dot{\mathbf{x}}$, turning the second order system into a set of two coupled first order equations:

$$\dot{\mathbf{x}} = \mathbf{v}, \quad \dot{\mathbf{v}} = \sigma\left(\mathbf{W}_x\mathbf{x} + \mathbf{W}_{\dot{x}}\mathbf{v} + \mathbf{Vu} + \mathbf{b}\right) - \gamma\mathbf{x} - \alpha\mathbf{v} \tag{2}$$

Then, for a fixed time step $0 < \Delta t < 1$, the hidden state of the RNN at time $t + 1$ can be defined as:

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \Delta t\big(\sigma\left(\mathbf{W}_x\mathbf{x}_t + \mathbf{W}_{\dot{x}}\mathbf{v}_t + \mathbf{V}\mathbf{u}_{t+1} + \mathbf{b}\right) - \gamma\mathbf{x}_t - \alpha\mathbf{v}_t\big) \tag{3}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta t\left(\mathbf{v}_{t+1}\right) \tag{4}$$

This model was theoretically demonstrated to have a bounded gradient and hidden state magnitude under assumptions on the time-step Δt and the infinity norm of the coupling parameters. Empirically, such stable gradient dynamics were shown to yield better performance than existing recurrent neural networks on tasks with very long time-dependencies. Importantly, in relation to this work, the hidden state $\mathbf{x}$ was not endowed with any notion of spatial layout, and thus the recurrent matrices $\mathbf{W}_x$ and $\mathbf{W}_{\dot{x}}$ were both fully connected, implying a global coupling between neurons. Throughout this paper we will refer to this model as the globally coupled oscillator network. Although fully connected coupling matrices are undeniably flexible, we argue that, similar to the distinction between fully-connected and convolutional layers for image processing, reducing the total parameter count can be beneficial for both generalization performance and computational efficiency as scale increases.

Topographic Connectivity In [17], the authors study a large scale spiking neural network model, quantifying the emergence of traveling waves, and comparing them with waves observed in the human cortex. At a high level, as it is relevant to this work, the study concludes that topographically organized connectivity and distance dependant conduction delays are both necessary and sufficient to produce traveling waves. Further they observe that such waves are fairly robust to the synaptic strengths of their model when given a sufficiently large number of neurons. Given these findings, we hypothesize that the Coupled Oscillitory Recurrent Neural Network may yield traveling waves if similarly constrained.

To impose such constraints we begin by defining an arbitrary topographic layout for the N-dimensional hidden state $\mathbf{x}$ in the model. For computational simplicity, we propose to use a regular 1 or 2 dimensional grid, $\mathbf{x}_{1D} \in \mathbb{R}^{C_h \times N}$ or $\mathbf{x}_{2D} \in \mathbb{R}^{C_h \times \sqrt{N} \times \sqrt{N}}$ respectively, where C_h is the number of simultaneous 'channels' in our hidden state. We then see that specifically, if the recurrent connections $\mathbf{W}_x$ and $\mathbf{W}_{\dot{x}}$ are made local over our spatial dimensions rather than global, and a distance-dependant time-delay introduced, the aforementioned constraints will be satisfied and the remainder of the properties such as synaptic strength and the precise local distribution of connections will be left up to the model to learn. In practice, we simplify the model by restricting the topographic connectivity of each neuron to its immediately adjacent neighbors in the grid, and define all distances to these neurons to be equal to 1. Such a simplification allows us to efficiently implement the local connections with a simple size 3 or 3×3 convolutional kernel for 1 and 2 dimensional grids respectively. Importantly, we see that such a constraint does not immediately invalidate any of the assumptions required for the theorems about mitigating the Exploding and Vanishing Gradient Problem (EVGP) since the infinity norm of the weights is unlikely to significantly increase when simply switching from fully to locally-connected matrices. In summary, our model is then given identically as in Equations 3 & 4 but with convolutional layers in place of the dense recurrent matrices. Explicitly, in the 2-dimensional setting, for convolutional kernels $\mathbf{w}_x, \mathbf{w}_{\dot{x}} \in \mathbb{R}^{C_h \times C_h \times 3 \times 3}$, we get:

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \Delta t\big(\sigma\left(\mathbf{w}_x \star \mathbf{x}_t + \mathbf{w}_{\dot{x}} \star \mathbf{v}_t + \mathbf{V}\mathbf{u}_{t+1} + \mathbf{b}\right) - \gamma\mathbf{x}_t - \alpha\mathbf{v}_t\big) \tag{5}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta t\left(\mathbf{v}_{t+1}\right) \tag{6}$$

We see that since $\mathbf{Vu}$ does not inherently have spatial dimensions, it must simply be reshaped to match the layout of the hidden state. In all following experiments for simplicity, we define our hidden state to have only have a single channel (C_h = 1), implying our convolutional kernels similarly have a single input and output channel each. We denote this model the Topographic Coupled Oscillator Recurrent Neural Network (TcoRNN).

Convergence Speed In line with the empirical analysis performed by [45], we first test our proposed topographic network on the long-sequence addition task using the hyperparameters found through grid search in [45] to be optimal for the coRNN. Importantly, we notice that the number of iterations to convergence for the globally coupled coRNN model is heavily dependant on initialization, resulting in many models failing to converge at all in the maximum allotted iterations. Specifically, we found these convergence difficulties most apparent on the extremely long sequences (length 2000 and 5000) where we were unable to get any models to converge in 60,000 iterations.[1] To quantify this effect, we trained 40 coRNN models and 40 TcoRNN on sequences of length 500 for hidden states of size 128 and 256. For the TcoRNN we define the hidden state topology to be a 2-dimensional grid of size 11×11 and 16×16 in each setting respectively. We then defined 'convergence' as the first evaluation iteration with MSE below 0.05. We note that this threshold approach to measuring convergence is believed to be valid in this setting since the addition task has a simple solution which results in training curves with very dramatic decreases in error from the random baseline (≈ 0.15) to < 0.05 in less than 50 iterations once the model 'solves' the task (see curves in [45]). Ultimately, in Table 1 we show the total number of runs which reach this convergence threshold in the total allotted iterations (30,000), and additionally the mean and median number of iterations required to reach convergence for the models which did converge. These results agree with our observation, showing that the TcoRNN converges nearly universally in a consistent number of iterations while it's globally coupled counterpart does not, and furthermore appears to get less consistent with increased dimensionality. Taken together with the dramatically reduced parameter count, these results suggest that topographic constraints may be a promising path forward to enhancing scalability of coupled oscillator networks.

Table 1: Number of converged models (MSE < 0.05) and mean (median) ± std. iterations to convergence for those models on length 500 sequential adding task.

	# Hidden	# Converged ↑	# Iterations ↓	# Param.
coRNN	128	34/40	7508 (**2900**) ± 8346	33,000
TcoRNN 2D		**39**/40	**4682** (4200) ± **1496**	381
coRNN	256	22/40	8887 (6450) ± 7256	131,000
TcoRNN 2D		**40**/40	**6345 (4000)** ± **4562**	786

Experiments on Sequence Classification To evaluate the impact of the structure imposed on the model, we compare the TcoRNN with its globally-coupled counterpart (coRNN) on the set of sequence classification tasks presented by [45] to specifically test the model's theoretical ability to handle long time-dependencies. In Table 2 we compare the accuracy (Acc.) and number of parameters (#θ measured in thousands) of the coRNN with both one and two dimensional TcoRNNs. We additionally include an entirely uncoupled baseline which corresponds to fixing the recurrent connections to be proportional to identity matrices, entirely eliminating topographic 'horizontal' connectivity. From the table we see that the topographic constraint incurs only a minor decrease in performance across most tasks, while simultaneously dramatically reducing the number of trainable parameters. Importantly, however, compared to completely removing all recurrent connections, the topographic model performs incomparably better, indicating that the such models are indeed learning to leveraging their limited degrees of coupling to perform complex processing operations – potentially through propagation of activity between neurons over extended lengths of time.

Table 2: Accuracy on supervised sequence benchmarks from [45]. All results are mean ± std. over 3 random initalizations. Additional experiment details can be found in the appendix.

	sMNIST		psMNIST		CIFAR10		IMDB	
	Acc.	#θ	Acc.	#θ	Acc.	#θ	Acc.	#θ
coRNN	99.1 ± 0.1	134	95.0 ± 2.4	134	58.3 ± 0.3	46	86.4 ± 0.2	46
TcoRNN 2D	92.2 ± 3.7	2.8	88.0 ± 2.7	2.8	47.8 ± 2.1	14	86.1 ± 0.3	13
TcoRNN 1D	91.8 ± 1.5	2.8	86.4 ± 1.8	2.8	48.4 ± 0.5	14	85.2 ± 0.4	13
Uncoupled	35.6 ± 0.1	2.8	37.0 ± 0.7	2.8	30.9 ± 0.4	14	50.4 ± 0.1	13

[1] We suspect that the original authors may have missed due to the grid search performing a simultaneous optimization of random initialization and hyperparameters, yielding a survivorship bias.

4 Learning Transformations as Traveling Waves

To investigate the main research question of this paper, whether traveling waves may be leveraged as an inductive bias towards learning structured representations, we propose to train the above model on sequences of transforming images and explore the simultaneous relationship between transformations in the hidden state and those of the input. In such a setting, a natural extension to our model which facilitates this exploration, and which similarly improves its flexibility, is to extend the model in Equations 5 & 6 to the autoregressive setting. In this section, we detail this extension and demonstrate that such a model indeed learns traveling waves which correspond directly to observed input transformations in an approximately equivariant manner.

Autoregressive Training Autoregressive models such as language models have lead to some of the most flexible and capable general purpose learning algorithms to date [11]. To extend the model from Section 3 to the autoregressive setting, we propose to add a learned decoder from the hidden state $\mathbf{x}_t$ back to the input at the next timestep $\mathbf{u}_{t+1}$. Explicitly, this results in the following equations:

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \Delta t\big(\sigma\left(\mathbf{w}_x \star \mathbf{x}_t + \mathbf{w}_{\dot{x}} \star \mathbf{v}_t + \mathbf{V} f_\theta(\mathbf{u}_{t+1}) + \mathbf{b}\right) - \gamma \mathbf{x}_t - \alpha \mathbf{v}_t\big) \tag{7}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta t \left(\mathbf{v}_{t+1}\right), \qquad \hat{\mathbf{u}}_{t+2} = g_\theta(\mathbf{x}_{t+1}) \tag{8}$$

Where f_θ and g_θ are the encoder and decoder networks respectively. We thus see that such a modification allows the model to be trained unsupervised through a mean-squared error reconstruction loss while simultaneously allowing for reliable sequence forecasting by training the model via standard 'teacher forcing'. As a second minor addition which we observe improves performance on long-term trajectory modeling tasks, we introduce an additional encoder network which learns to predict the initial conditions $\mathbf{x}_0$ and $\mathbf{v}_0$ of the network given a partial 'inference' sequence. Explicitly, we can write this as: $\mathbf{x}_0, \mathbf{v}_0 = f_\theta^{IC}(\{\mathbf{u_t}\}_{t=0}^{T_{inf}})$. Such an initial-condition network is common in the Neural-ODE literature [12], and in this setting helps to stabilize the latent dynamics which would otherwise take a significant number of iterations to reach their final magnitude.

Visualizing & Measuring Traveling Waves To verify that the proposed model indeed exhibits traveling waves, we begin by training a small 2D-TcoRNN with a linear encoder and decoder on a simple rotating MNIST benchmark [35] and visualize spatially propagating hidden state activations over time. To support this, we borrow a signal analysis technique frequently used in the neuroscience literature to directly compute the instantaneous phase and velocity of the putative waves from noisy real-valued signals. Specifically, we follow the work of [18] and compute the 'generalized phase' of a real valued signal $\mathbf{x}(t)$ by first transforming the signal to a complex-valued analytic signal $\mathbf{x}_a(t)$ through the Hilbert transform $\mathcal{H}$ and then taking the complex argument of this signal as the phase $\boldsymbol{\phi}(t)$ at each point in space and time. Formally: $\mathbf{x}_a(t) = \mathbf{x}(t) + i\mathcal{H}[\mathbf{x}(t)]$, and $\boldsymbol{\phi}(t) = Arg[\mathbf{x}_a(t)]$. Finally, wave velocities can then straightforwardly be computed using the spatial gradient of this phase: $\boldsymbol{\nu} = -\nabla\boldsymbol{\phi}$.

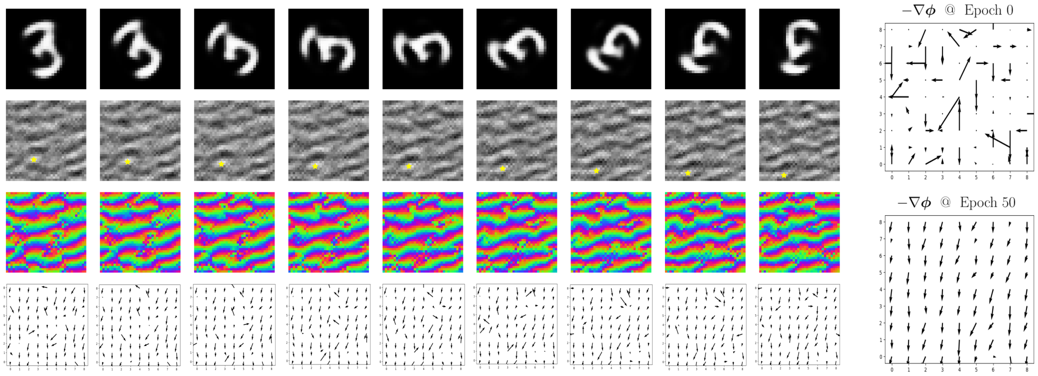

Figure 1: (Left) Plot of reconstructions $\hat{\mathbf{u}}$ (top), hidden state $\mathbf{x}$ (2^{nd}), generalized phase $\boldsymbol{\phi}$ (3^{rd}), and estimated wave velocity $-\nabla\boldsymbol{\phi}$ (bottom) over the course of a transformation sequence. A small gold star is added to help track the approximate peak of a traveling wave in the hidden state. (Right) Plot of estimated wave velocity before and after training. Further details in the appendix.

In Figure 1 on the left we visualize the hidden state of our network $\mathbf{x}$, the generalized phase ϕ, and the wave velocities $-\nabla\phi$ at a series of timesteps. We see that the hidden state indeed contains periodic spatially localized regions of high magnitude which translate over time. Further, these movements correspond directly with the computed putative wave velocity as desired. Interestingly, in the two plots on the right of Figure 1, we plot the average instantaneous velocity of each neuron before (top) and after training (bottom). We see that these structured waves do not exist early in training, and only become visually apparent when the training loss begins to decrease significantly, implying their function is indeed learned from a general inductive bias.

Inducing Traveling Waves To demonstrate that the observed waves in feature space approximately commute with the input transformation, and thus can be seen as an approximately equivariant latent operator, we show that by artificially inducing traveling waves in the hidden state, we can reconstruct appropriately transforming images at each time step. Given the high degree of flexibility of the potentially emergent wave dynamics of the 2-D system presented in Figure 1, we concede that two restrictions must be placed on the model in order for us to be able to accurately induce waves which match those the model has learned. Explicitly, we first define the latent space to be a set of disjoint 1-dimensional tori (circles) such that learned wave propagation will be restricted to a single axis. Secondly, we restrict our topographic coupling to be 1-directional by masking out all weights except for one (non-central weight) in our convolutional kernel which is shared over all tori. In combination, these restrictions ensure that *if* traveling waves are learned by the model, they will likely be able to be approximately modeled by solutions to the 1-dimensional 1-way wave equation: $y(x, t) = f(x - vt)$.

In Figure 2 we depict the results of this experiment. In detail, we train the 1D TcoRNN described above on a dataset of length $T = 18$ sequences of rotating MNIST digits. At test time, we encode a full sequence and take the final hidden state $\mathbf{x}_T$ as the initial state for our system. We then sequentially shift (or linearly interpolate) activations across the spatial dimension of each circular subspace according to our assumed velocity, decoding back to the image space at each step. The result in Figure 2 shows that indeed by inducing such traveling wave activity into the hidden state we observe an approximate reconstruction of the learned transformation. In this case we assume $v = 1$ and observe that the induced transformation (shown on the bottom) is slightly faster than the ground truth transformation, yet the reconstructions remain consistent even past the first period.

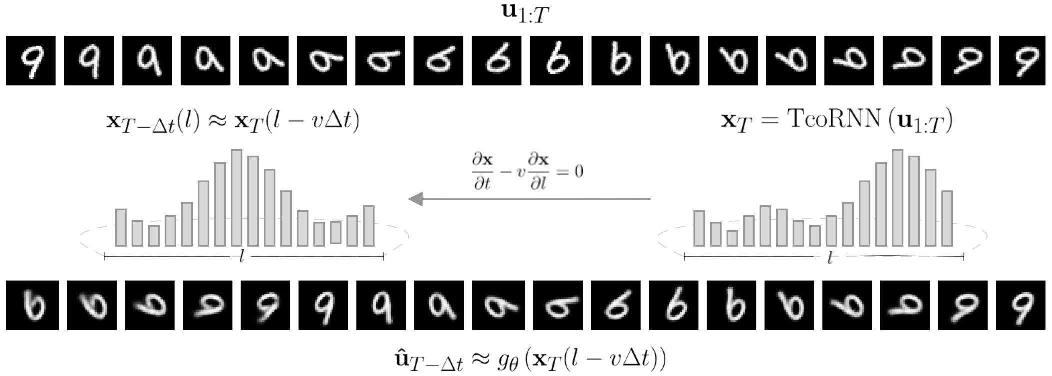
$\mathbf{u}_{1:T}$
$\mathbf{x}_{T-\Delta t}(l) \approx \mathbf{x}_T(l - v\Delta t)$
$\mathbf{x}_T = \text{TcoRNN}(\mathbf{u}_{1:T})$
$\frac{\partial \mathbf{x}}{\partial t} - v\frac{\partial \mathbf{x}}{\partial l} = 0$
l
l
$\hat{\mathbf{u}}_{T-\Delta t} \approx g_\theta(\mathbf{x}_T(l - v\Delta t))$

Figure 2: Visualization of an induced traveling wave in the one-dimensional hidden state, and the resulting reconstructed image sequence (bottom). We see that the model accurately rotates the image backwards as waves are propagated backwards in the latent space.

5 Hamiltonian Dynamics Suite

To test the ability of our model to leverage the imposed topographic structure in more complex settings, we evaluate the full autoregressive model described in Equations 8 & 7 on the hamiltonian dynamics benchmark outlined by [9]. At a high level the benchmark consists of a diversity of tasks governed by known equations of motion, designed to test the ability of models to learn different kinds of dynamics directly from high dimensional pixel observations. The tasks include modeling toy phsyics examples such as idealized springs and pendulums, as well as molecular dynamics, cyclic games, and camera motion in three dimensional space. Models are evaluated based on the number of time steps into the future they are able to accurately predict the dynamics of the system with reconstruction error under a predefined threshold (denoted valid precition time (VPT)), as well as based on their raw reconstruction error over these extrapolated trajectories. In Table 3 we compare performance over 13 distinct tasks for our model (1D & 2D), the globally coupled coRNN counterpart, and three state of the art models (HGN++ [29], standard autoregressive models (AR) [30], and Neural ODEs [12] trained both forwards and backwards in time (ODE [TR])). For the three baseline models, we use the hyperparameters given in [9, 29] to be optimal, and a hidden dimensionality of 32. For the topographic models, we increase the hidden state dimensionality to 23×23 to allow for sufficient space for the emergence of spatial structure in the hidden state. Although this hidden state is therefore significantly larger, we do not intend this to be a direct comparison of model performance, but rather as an investigation of the capabilities of topographic coupled oscillator networks of sufficiently large spatial dimensions. Finally, for the globally connected coRNN we additionally set the hidden dimensionality to $529 = 23 \times 23$ to serve as a equal dimensionality comparison and study the direct impact of topographic connections on model performance.

Table 3: Valid Prediction Time 'VPT' ($\pm$ std.) on the Hamiltonian Dynamics Benchmark.

	HGN++	AR	ODE [TR]	coRNN	TcoRNN 2D	TcoRNN 1D
Spring	447 (0)	302 (63)	430 (26)	375 (14)	311.8 (27)	431 (24)
Pendulum	105 (21)	3 (4)	212 (65)	179 (91)	155.1 (24)	174 (65)
Double Pendulum	11 (5)	0 (0)	22 (7)	3 (1)	9 (9)	10 (8)
Two Body	444 (3)	263 (92)	439 (11)	431 (40)	413 (53)	420 (27)
RPS	141 (23)	77 (15)	124 (23)	109 (36)	133 (18)	110 (12)
Pennies	79 (6)	118 (25)	164 (14)	165 (23)	141 (37)	163 (9)
Mujoco Circle	3 (2)	5 (3)	47 (8)	13 (10)	3 (2)	5 (4)
Molecular Dynamics 4	0 (0)	0 (0)	13 (2)	0 (0)	0 (0)	0 (0)
Molecular Dynamics 16	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Spring + C	283 (27)	2 (2)	374 (36)	42 (18)	17 (16)	22 (19)
Pendulum + C	21 (8)	0 (0)	47 (26)	1 (1)	6 (3)	1 (1)
Double Pendulum + C	16 (6)	0 (0)	22 (7)	0 (0)	1 (1)	10(8)
Two Body + C	46 (5)	24.3 (9)	73 (29)	28 (6)	42 (9)	32 (14)

Ultimately, we see that the TcoRNN compares competitively with the state of the art models, and even surpasses the performance of the globally coupled counterpart on simple tasks such as the spring. Additionally, as visualized in the appendix, we see that the TcoRNN again displays complex spatial dynamics in its hidden state in correspondence with the input transformation, although the dynamics are significantly more complex than the simple one-way waves leveraged in Figure 2. Anecdotally, we see that many solutions appear to incorporate dynamics qualitatively similar to standing waves to model the periodic nature of the input sequences. Finally, we note that the model performs poorly on tasks denoted $+C$ (implying 'plus color') in Table 3. These tasks involve the same underlying system, but with fundamental properties of the system shifting between training examples. For example, in the spring task, the mass, color and location of the spring in the image change for each example. We see that the globally coupled oscillator network and the standard autoregressive network do not perform overwhelmingly better on such tasks, suggesting that perhaps it is a limitation in the autoregressive design as a whole.

In summary, we think these experiments show that modern deep neural networks are able to positively leverage the computational complexity of coupled oscillatory systems for temporal information processing, potentially shedding light on the origins and computational purpose of traveling waves observed in the biological cortex.

6 Related Work

In addition to the models which have similarity to our model in functional form, such as Neural ODEs [12], coRNNs [45], and autoregressive models [11], this work is related to a vast amount of work which attempts to incorporate symmetries, conservation laws, and physical priors into neural networks. One prominent recent example is Physics Informed Neural Networks (PINNs) [43] which allow for the direct inclusion of partial differential equation constraints into the optimization procedure of a neural network. Unlike our model, these models are defined with continuous spatial dimensions as inputs allowing for gradients with respect to these dimensions to be computed exactly with automatic differentiation. In contrast, our model has an explicit discretized spatial layout to its hidden state over which structure is imposed. Similarly, Hamiltonian and Lagrangian Neural Networks [26, 53, 29, 16] propose to directly define dynamics of the hidden state through the analytic gradients of a learned hamiltonian or lagrangian function. Abstractly, our work can be seen as similar to these but with very restricted set of functions learnable for the hamiltonian. Finally, a recent follow up work to the coRNN extends it to the graph neural network (GNN) setting [44] for the purpose of solving the 'oversmoothing' in deep GNNs. Although for an appropriately defined graph such a model could be interpreted as similarly having a 'topographic' organization, we note that the motivations and directions of both papers appear largely distinct.

Equivariant neural networks [14, 13, 58, 57, 23, 56] can additionally be seen as an exact analytic group theoretic approach to the inclusion of symmetry information in neural networks. In line with the motivation of this work, some recent works have attempted to learn more flexible structured symmetry representations directly from the data through supervision of some form [5, 15, 19]. Other models have approached the problem of learning symmetries with meta-learning, aiming to either meta-learn the weight-sharing scheme defining the symmetry explicitly [60], or by meta-learning conservation laws which improve model performance at test time [2].

Most related to this work in both motivation and form is the Topographic VAE (TVAE) [35, 34]. Our model can be seen as an extension of the TVAE which learns a much more flexible dynamic roll operator in the latent space rather than having this enforced a priori. Some of the immediately apparent benefits of this additional flexibility are the fact that our model naturally handles higher dimensional 'capsules', as well as capsules with arbitrary boundary conditions. Further we see that our model can functionally learn 'roll' operators of various speeds simultaneously while the TVAE must have these predefined.

7 Conclusion

In this work we introduce the hypothesis that a system biased towards exhibiting traveling waves may be capable of learning structured representations of symmetry transformations without supervision. To test this, we introduce the biologically relevant Topographic Coupled Oscillator Network capable of exhibiting such traveling waves in a flexible computational framework. We observe that once trained, traveling waves can be seen to propagate through the latent state synchronously with input transformations, and reciprocally by propagating initial latent state activity with simple wave equations we can generate sequence transformations in the input space – implying approximate commutativity between the traveling waves, the feature extractor, and the input transformation. Finally we demonstrate that such a topographically constrained model performs competitively with state of the art on complex dynamics modeling and sequence classification benchmarks, while additionally showing signs of improved scalability and robustness on toy long-sequence tasks.

Limitations Given the flexibility of hidden state dynamics of our model, the imposed structure bias is not as strong as the exactly structured representations found in group equivariant neural networks. Because of this, to achieve many of the performance benefits expected from equivariant neural networks, further tuning of the proposed model architecture and hyper-parameters may be required. For example, it remains unclear how to structure to topographic organization to best encourage structure while simultaneously allowing flexible computation, although our initial results seem to imply this is already feasible without significant tuning. This work therefore serves as evidence of the possibility of flexible learned structure emerging in the form of traveling waves, rather than as an evaluation of the benefits of such structure. In future work we intend to explore modifications such as longer distance connections with increased time-delays which we believe may indeed yield a stronger structure bias and measurable performance improvements. Additionally, we intend to explore the integration of topographic constraints into recent extensions of the of the coRNN such as the UNIcoRNN [46] and LEM [47] models which may confer further flexibility and computational efficiency.

Broader Impact As generative models become more capable of learning generalizable transformations, their ability to generate more convincing artificial data similarly increases. We thus refer to the broader impact of advances in generative models more generally as they identically apply here.

Acknowledgments and Disclosure of Funding

We would like to thank the creators of Weight & Biases [8] and PyTorch [42]. Without these tools our work would not have been possible. We thank the Bosch Center for Artificial Intelligence for funding.

References

[1] Andrea Alamia and Rufin VanRullen. Alpha oscillations and traveling waves: Signatures of predictive coding? *PLOS Biology*, 17(10):e3000487, October 2019.

[2] Ferran Alet, Dylan Doblar, Allan Zhou, Joshua Tenenbaum, Kenji Kawaguchi, and Chelsea Finn. Noether networks: Meta-learning useful conserved quantities, 2021.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[4] Erik J Bekkers, Maxime W Lafarge, Mitko Veta, Koen AJ Eppenhof, Josien PW Pluim, and Remco Duits. Roto-translation covariant convolutional networks for medical image analysis, 2018.

[5] Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Learning invariances in neural networks. *Advances in Neural Information Processing Systems*, December, 2020.

[6] William Berrios and Arturo Deza. Joint rotational invariance and adversarial training of a dual-stream transformer yields state of the art brain-score for area v4. In *Brain-Score Workshop*, 2022.

[7] Michel Besserve, Nikos Logothetis, and Bernhard Schölkopf. Shifts of gamma phase across primary visual cortical sites reflect dynamic stimulus-modulated information transfer. 01 2015.

[8] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[9] Aleksandar Botev, Andrew Jaegle, Peter Wirnsberger, Daniel Hennes, and Irina Higgins. Which priors matter? benchmarking models for learning latent dynamics, 2021.

[10] Diane Bouchacourt, Mark Ibrahim, and Stéphane Deny. Addressing the topological defects of disentanglement via distributed operators. *ArXiv*, abs/2102.05623, 2021.

[11] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[12] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2018.

[13] Taco Cohen and M. Welling. Steerable cnns. *ArXiv*, abs/1612.08498, 2017.

[14] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016.

[15] Marissa Connor, Gregory Canal, and Christopher Rozell. Variational autoencoder with learned latent structure. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2359–2367. PMLR, 13–15 Apr 2021.

[16] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks, 2020.

[17] Zachary W. Davis, Gabriel B. Benigno, Charlee Fletterman, Theo Desbordes, Christopher Steward, Terrence J. Sejnowski, John H. Reynolds, and Lyle Muller. Spontaneous traveling waves naturally emerge from horizontal fiber time delays and travel through locally asynchronous-irregular states. *Nature Communications*, 12(1), October 2021.

[18] Zachary W. Davis, Lyle Muller, Julio Martinez-Trujillo, Terrence Sejnowski, and John H. Reynolds. Spontaneous travelling cortical waves gate perception in behaving primates. *Nature*, 587(7834):432–436, October 2020.

[19] Nichita Diaconu and Daniel Worrall. Learning to convolve: A generalized weight-tying approach. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1586–1595. PMLR, 09–15 Jun 2019.

[20] NE Diamant and A Bortoff. Nature of the intestinal low-wave frequency gradient. *American Journal of Physiology-Legacy Content*, 216(2):301–307, February 1969.

[21] G Bard Ermentrout and David Kleinfeld. Traveling electrical waves in cortex: insights from phase dynamics and speculation on a computational role. *Neuron*, 29(1):33–44, 2001.

[22] George Bard Ermentrout and Nancy Kopell. Frequency plateaus in a chain of weakly coupled oscillators, i. *SIAM journal on Mathematical Analysis*, 15(2):215–237, 1984.

[23] Marc Finzi, Max Welling, and Andrew Gordon Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3318–3328. PMLR, 18–24 Jul 2021.

[24] Karl J. Friston. Waves of prediction. *PLOS Biology*, 17(10):e3000426, October 2019.

[25] Pulin Gong and Cees Van Leeuwen. Distributed dynamical computation in neural circuits with propagating coherent activity patterns. *PLoS Computational Biology*, 5(12):e1000611, 2009.

[26] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[27] Irina Higgins, Le Chang, Victoria Langston, Demis Hassabis, Christopher Summerfield, Doris Tsao, and Matthew Botvinick. Unsupervised deep learning identifies semantic disentanglement in single inferotemporal face patch neurons. *Nature Communications*, 12(1), November 2021.

[28] Irina Higgins, Sébastien Racanière, and Danilo Rezende. Symmetry-based representations for artificial and biological general intelligence. *Frontiers in Computational Neuroscience*, 16, April 2022.

[29] Irina Higgins, Peter Wirnsberger, Andrew Jaegle, and Aleksandar Botev. Symetric: Measuring the quality of learnt hamiltonian dynamics inferred from vision. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25591–25605. Curran Associates, Inc., 2021.

[30] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[31] Eugene M Izhikevich and Frank C. Hoppensteadt. Polychronous wavefront computations. *International Journal of Bifurcation and Chaos*, 19(5):1733–1739, 01 2008.

[32] Dirk Jancke, Frédéric Chavane, Shmuel Naaman, and Amiram Grinvald. Imaging cortical correlates of illusion in early visual cortex. *Nature*, 428(6981):423–426, 2004.

[33] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen,

David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, July 2021.

[34] T. Anderson Keller and Max Welling. Predictive coding with topographic variational autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1086–1091, October 2021.

[35] T. Anderson Keller and Max Welling. Topographic vaes learn equivariant capsules, 2021.

[36] Jean-Rémi King and Valentin Wyart. The human brain encodes a chronicle of visual events at each instant of time through the multiplexing of traveling waves. *The Journal of Neuroscience*, 41(34):7224–7233, April 2021.

[37] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[38] Yoshiki Kuramoto. Rhythms and turbulence in populations of chemical oscillators. *Physica A: Statistical Mechanics and its Applications*, 106(1-2):128–143, 1981.

[39] Lyle Muller, Frédéric Chavane, John Reynolds, and Terrence J. Sejnowski. Cortical travelling waves: mechanisms and computational principles. *Nature Reviews Neuroscience*, 19(5):255–268, March 2018.

[40] Lyle Muller, Giovanni Piantoni, Dominik Koller, Sydney S Cash, Eric Halgren, and Terrence J Sejnowski. Rotating waves during human sleep spindles organize global patterns of activity that repeat precisely through the night. *eLife*, 5, November 2016.

[41] Lyle Muller, Alexandre Reynaud, Frédéric Chavane, and Alain Destexhe. The stimulus-evoked population response in visual cortex of awake monkey is a propagating wave. *Nature Communications*, 5(1), April 2014.

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.

[43] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[44] T. Konstantin Rusch, Benjamin P. Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael M. Bronstein. Graph-coupled oscillator networks, 2022.

[45] T. Konstantin Rusch and Siddhartha Mishra. Coupled oscillatory recurrent neural network (cornn): An accurate and (gradient) stable architecture for learning long time dependencies. In *International Conference on Learning Representations*, 2021.

[46] T. Konstantin Rusch and Siddhartha Mishra. Unicornn: A recurrent model for learning very long time dependencies, 2021.

[47] T Konstantin Rusch, Siddhartha Mishra, N Benjamin Erichson, and Michael W Mahoney. Long expressive memory for sequence modeling. In *International Conference on Learning Representations*, 2022.

[48] Nicole C. Rust and James J. DiCarlo. Selectivity and tolerance (“invariance”) both increase as visual information propagates from cortical area v4 to it. *Journal of Neuroscience*, 30(39):12978–12995, 2010.

[49] Tatsuo K. Sato, Ian Nauhaus, and Matteo Carandini. Traveling waves in visual cortex. *Neuron*, 75(2):218–229, July 2012.

[50] Victor Garcia Satorras, Emiel Hoogeboom, Fabian B. Fuchs, Ingmar Posner, and Max Welling. E(n) equivariant normalizing flows, 2021.

[51] Matthew Schwartz. Harvard physics 15c lecture notes, lecture 4, Spring 2016.

[52] Klaus M. Stiefel and G. Bard Ermentrout. Neurons as oscillators. *Journal of Neurophysiology*, 116(6):2950–2960, December 2016.

[53] Peter Toth, Danilo Jimenez Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. *arXiv preprint arXiv:1909.13789*, 2019.

[54] Elise van der Pol, Daniel E. Worrall, Herke van Hoof, Frans A. Oliehoek, and Max Welling. MDP homomorphic networks: Group symmetries in reinforcement learning. *CoRR*, abs/2006.16908, 2020.

[55] Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. *CoRR*, abs/1806.03962, 2018.

[56] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 10402–10413, Red Hook, NY, USA, 2018. Curran Associates Inc.

[57] Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[58] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7168–7177, 2017.

[59] Honghui Zhang, Andrew J. Watrous, Ansh Patel, and Joshua Jacobs. Theta and alpha oscillations are traveling waves in the human neocortex. *Neuron*, 98(6):1269–1281.e4, June 2018.

[60] Allan Zhou, Tom Knowles, and Chelsea Finn. Meta-learning symmetries by reparameterization, 2020.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Conclusion
   (b) Did you describe the limitations of your work? [Yes] See Conclusion

(c) Did you discuss any potential negative societal impacts of your work? [Yes] See Conclusion

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [N/A]

   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Supplementary Material

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Supplementary Material

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See ± std on tables.

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Supplementary Material

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes] See Supplementary Material

   (b) Did you mention the license of the assets? [Yes] See Supplementary Material

   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix A Experiment Details

The code is built as extensions of three existing public repositories, allowing us to reproduce all baseline results from the original authors' code. Specifically, we make use: (I) The coRNN repository `<https://github.com/tk-rusch/coRNN>` for the supervised sequence experiments, (II) The Topographic VAE repository `<https://github.com/akandykeller/TopographicVAE/>` for the rotating MNIST experiments, and (III) The DeepMind Physics Inspired Models repository `<https://github.com/deepmind/deepmind-research/tree/master/physics_inspired_models>` for the Hamiltonian Dynamics Suite Experiments.

A.1 Convergence Test and Sequence Classification

The experiments from section 3 were performed by modifying the published code for the original coRNN [45] to incorporate the topographic connectivity constraints outlined in the main text. All hyperparameters were thus set to the defaults in the published code which matched the optimal hyperparameters stated by the authors to be found from a grid search on each dataset independently. The baseline coRNN values in Tables 1 & 2 are thus simply from re-running the original authors code, and we observe similar values to those published in [45]. We acknowledge that running a separate grid search for the TcoRNN models may be beneficial to their performance but we were unable to do so due to time and computational constraints and thus leave this to future work. In practice we found the original coRNN parameters worked well enough to give an initial intuition for the relative performance.

For the 2D TcoRNN, the topology of the hidden state was defined to be a regular square 2D grid with side lengths equal to square root of the default hidden state size (or the integer floor of the square root for non-perfect-square values). Each neuron was defined to be connected to its immediate surrounding 8 cells in the grid, in addition to a self-connection. The boundary conditions of the topology were defined to be periodic (implemented through circular padding) such that the global topology was that of a 2-dimensional torus. The recurrent topographic coupling parameters were shared over all spatial locations of the grid, allowing the above topographic connectivity to be implemented as a periodic convolution with a kernel of size 3×3.

Similarly, for the 1D TcoRNN, the topology of the hidden state was defined to be a set of disjoint 1-D tori (circles) with length again equal to the square root of the default hidden size. To keep dimensionality equal, the number of total disjoint circles was then also set to this same square root (or floor) value. Each neurons was defined to be connected to its immediate surrounding 2 cells in the circle, in addition to a self-connection. The recurrent topographic coupling parameters were shared over all locations and over all circles, allowing the above topographic connectivity to be implemented as a periodic convolution with a kernel of size 1×3.

A.2 Learning Transformations as Traveling Waves

The experiments in Section 4 were performed by modifying the published code for the Topographic VAE [35] to introduce our proposed TcoRNN in place of the 'shifting temporal coherence' construction of the topographic Student's-T variable in the original paper. To achieve this, the encoder and decoder (f_θ & g_θ) of Equations 7 & 8 were implemented as a variational autoencoder [37] with a standard Gaussian prior and Bernoulli distribution for the likelihood of the data. Practically, this was achieved by setting the output dimensionality of the encoder f_θ to twice the hidden state dimensionality, defining half of the outputs as the posterior mean μ_θ, and the second half as the log of the posterior variance σ_θ. We additionally found that applying Layer Normalization [3] (denoted LN) to the output of the encoder helped increase convergence speed. Explicitly, the model can thus be described as:

$$\mathbf{z}_{t+1} \sim q_\theta(\mathbf{z}_{t+1}|\mathbf{u}_{t+1}) = \mathcal{N}\big(\mathbf{z}_{t+1}; \mu_\theta(\mathbf{u}_{t+1}), \sigma_\theta(\mathbf{u}_{t+1})\mathbf{I}\big), \qquad \bar{\mathbf{z}}_{t+1} = \mathrm{LN}(\mathbf{z}_{t+1}) \tag{9}$$

$$\mathbf{v}_{t+1} = \mathbf{v}_t + \Delta t\big(\sigma\left(\mathbf{w}_x \star \mathbf{x}_t + \mathbf{w}_{\dot{x}} \star \mathbf{v}_t + \mathbf{V}\bar{\mathbf{z}}_{t+1} + \mathbf{b}\right) - \gamma \mathbf{x}_t - \alpha \mathbf{v}_t\big) \tag{10}$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \Delta t\left(\mathbf{v}_{t+1}\right) \tag{11}$$

$$p_\theta(\mathbf{u}_{t+2}|g_\theta(\mathbf{x}_{t+1})) = \mathrm{Bernoilli}(\mathbf{u}_{t+2}; g_\theta(\mathbf{x}_{t+1})) \tag{12}$$

Where the objective is then computed by averaging the evidence lower bound (ELBO) over the length of the sequence:

$$\mathcal{L}(\mathbf{u}_{1:T};\boldsymbol{\theta}) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{\mathbf{z}_t \sim q_\theta(\mathbf{z}_t|\mathbf{u}_t)}\big(\log p_\theta(\mathbf{u}_{t+1}|g_\theta(\mathbf{x}_t)) - D_{KL}[q_\theta(\mathbf{z}_t|\mathbf{u}_t)||p_{\mathbf{Z}}(\mathbf{z}_t)]\big) \tag{13}$$

The initial conditions for the TcoRNN were then given by simply setting the initial position equal to the first encoder output, and the initial velocity to zero, i.e. $\mathbf{x}_0 = \bar{\mathbf{z}}_0$ & $\mathbf{v}_0 = \mathbf{0}$. Although we did not test the experiments in Section 4 with a deterministic autoencoder, we note that traveling waves can also clearly be seen in the hidden states of the deterministic models presented in Sections 3 and 5 (as visualized in the github repository), implying that the variational formulation is not necessary for the emergence of traveling waves.

For the experiment depicted in Figure 1 of Section 4, we used a simple linear encoder and decoder, and a hidden state dimensionality of 1296 reshaped into a 2D grid of shape 36×36. As in the rest of the paper, our topographic connectivity was implemented using a convolutional kernel of shape 3×3 shared over all elements of the grid, with circular padding to enforce periodic boundary conditions on the grid. For training, we presented the model with length 18 sequences of MNIST digits rotating at 20 degrees per step (thus completing a full period per training sequence). At test time, to create the visualization in Figure 1, we increased the sequence length to 72 elements (or four periods) and visualize a portion of the final period, allowing the system to reach a steady state of wave activity for better visualization. We see that despite not being trained on such long sequences, the TcoRNN is able to generalize and maintain wave activity. For computing the generalized phase, we set use a 4-th order butterworth bandpass filter with bounds set at 0.2 and 0.4 of the Nyquist frequency. As hyperparamters for training, we used standard SGD with momentum of 0.9, a learning rate of 2.5×10^{-4}, and a batch size of 128 for 50 epochs. Following the suggestion outlined in [45], we allowed the parameters γ, α, & Δt to be learned during training by initializing them to $\Delta t = \sigma^{-1}(0.125) = -1.95$, $\gamma = 1.0$, & $\alpha = 0.5$ and then applying appropriate activation functions to keep them within the desired bounds (e.g. sigmoid, ReLU, & ReLU respectively). These hyperparameters and initalization values were determined by implementing a simple toy version of the model with random data and random weights and manually altering parameters to determine the ranges for which coherent wave dynamics were likely to emerge. We note that the properties of the emergent waves appear qualitatively different for different random initalizations of the model. Specifically the wavelength and velocity of the waves appears to vary greatly from run-to-run. We show a few of these different learned dynamics in the additional results section below.

For the experiment depicted in Figure 2 of Section 4, we used a 3-layer Multi-Layer Perceptron (MLP) for the both encoder and decoder, and a hidden state of dimensionality 1296 reshaped into a set of 24 disjoint 1-D tori (circles) each composed of 54 neurons. We implemented topographic coupling between the immediate neighbors on each circle via a 1-dimensional convolutional kernel of size 3 with circular padding. We then implemented the uni-directionality constraint outlined in the main text be masking the first two elements of the kernel to 0, yielding a kernel with a single trainable parameter explicitly connecting each neuron with its neighbor directly to one side. For training, the dataset and hyperparameters all remained the same as in Figure 1 described above, however the batch size was reduced to 8 for quicker evaluation. We found that additionally adding another layer normalization layer between recurrent steps improved the consistency of the learned waves and thus allowed us to simulate them more accurately at test time. Explicitly this amounted to modifying Equation 11 to: $\mathbf{x}_{t+1} = \mathrm{LN}\big(\mathbf{x}_t + \Delta t\,(\mathbf{v}_{t+1})\big)$. Furthermore, to ensure consistency of waves across each circular subspace separately, we shared the bias vector $\mathbf{b}$ across each subspace. To induce a traveling wave in the hidden state of the network and thereby generate the transformation sequence shown in the bottom row of the figure, we first encode the input sequence (shown in the top row), using the equations outlined in this section. We take the final hidden state of the network ($\mathbf{x}_T$) as the initial state from which we begin the wave propagation. Then, across each 1-D circular subspace of the hidden state, we update the values of the hidden state based on the 1-D 1-way wave equation $y(x,t) = f(x - vt)$ for a velocity $v = 1$ for time $t = 1$ to 18. Written in terms of the hidden state $\mathbf{x}_t$, we can effectively propagate waves backwards through the hidden state by moving activation from one spatial location l to a location shifted by $v\Delta t$: $\mathbf{x}_T(l) \rightarrow \mathbf{x}_T(l - v\Delta t)$. Practically, this amounts to sequentially circularly shifting the hidden state activation across each circular subspace as depicted in the middle of Figure 2.

A.3 Hamiltonian Dynamics Suite

The experiments in Section 5 were performed using the DeepMind Physics Inspired Models and Hamiltonian Dynamics Suite, implemented in JAX, as a starting point. All values reported for the baselines (HGN++, AR, and ODE [TR]) were thus obtained by re-running the original code with the hyperparameters stated in [9]. Specifically, for the HGN++, we trained the model both forwards and backwards in time, including over the inference steps, with a final beta value of 0.1 in the ELBO. For the AR model, we used an LSTM with all other paramters default. For the ODE, we used the default parameters with forwards and backwards training, again including inference steps. The only change to the default hyperparamters for all three models was to reduce the batch size to 8 per GPU (thus 32 total per iteration) to fit on our GPUs.

The coRNN and TcoRNN architectures were added as extensions to the auto-regressive model already implemented in library. They thus made use of all the same default hyperparameters, with the only changed values being the aforementioned reduced batch size, an increased number of inference steps (31), an increased number of target steps (60), and an increased hidden state size (23×23). The increased number of inference and target steps was found useful to improve performance on more chaotic tasks such as the pendulum where the the accuracy of the initial state is hugely important to the model forecasting performance. Additionally, we note that these values are within the values searched by the grid search of the authors in [9] making their use here for comparison relatively fair. The size of the hidden state was picked as the largest which fit in our GPU memory across all devices. The values of α, γ, and Δt were initialized to the same values as the MNIST experiments described above, and were again allowed to be updated during training simultaneously with the other model parameters. For the 2D TcoRNN, the hidden state topology was again defined to be a 2D torus of size (23×23) implemented through periodic convolution with a 3×3 kernel. The 1D TcoRNN topology was similarly composed of 23 disjoint 1D circles each with 23 neurons, again implemented with periodic convolution with a 1×3 kernel. The coRNN and TcoRNN models additionally used a separate initial condition network to initialize $\mathbf{x}_0$ and $\mathbf{v}_0$. This network was implemented as a GRU with a hidden state of size $2 \times 23 \times 23$ which ran backwards over the inference sequence (length 31) first embedded with the model encoder f_θ. The final hidden state of the model was then split in half and taken to initialize the inital positions and velocities of the coRNN & TcoRNNs.

All models make use of the same deep convolutional encoder with ReLU activations and a similarly deep convolutional spatial broadcast decoder as in the original work. They were similarly all trained for 500,000 iterations to match the original work. Given the high variance of the VPT value from batch-to-batch, the values reported in Table 3 were computed as the mean and standard deviation of the VPT over the final 5 evaluation iterations. We see that the values roughly agree with those reported in [9], however certain discrepancies may still appear due to the fact that the authors of [9] only report the range of the grid search they performed but not the actual hyperparameter values of their best performing models.

A.4 Hardware Details

All models were run on a cluster across roughly 8 NVIDIA GeForce 1080Ti GPUs, 8 NVIDIA GeForce 980Ti GPUs, and 8 NVIDIA Titan X Gpus. Each model in Table 3 thus required roughly 6-8 GPU days to train to the final number of iterations.

Appendix B Extended Results

B.1 Visualizing Traveling Waves on MNIST

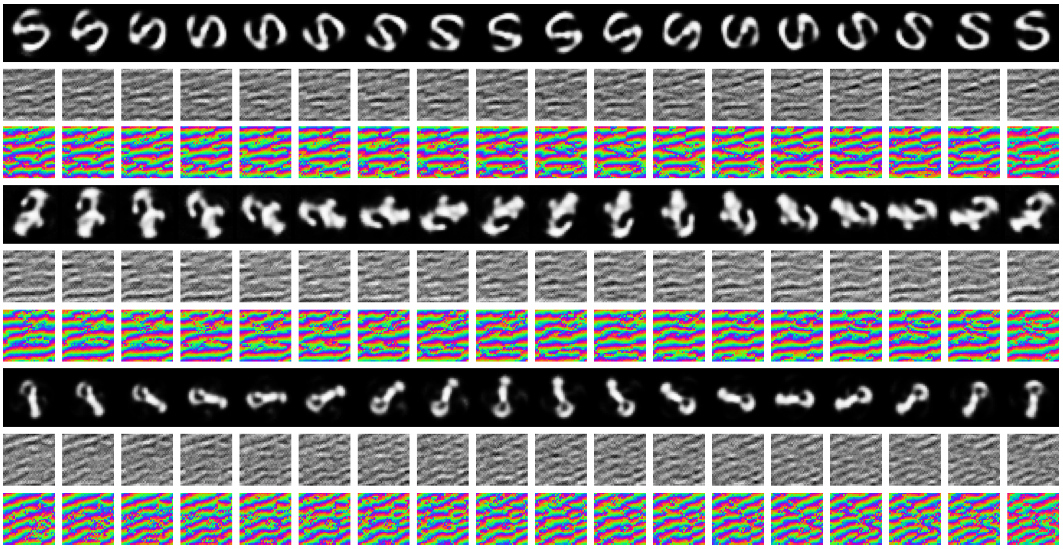

Figure 3: Additional hidden state visualizations for the model in Figure 1. Reconstructions (Top), Hidden state (middle) and generalized phase (bottom), for the final 18 timesteps of the test sequence.

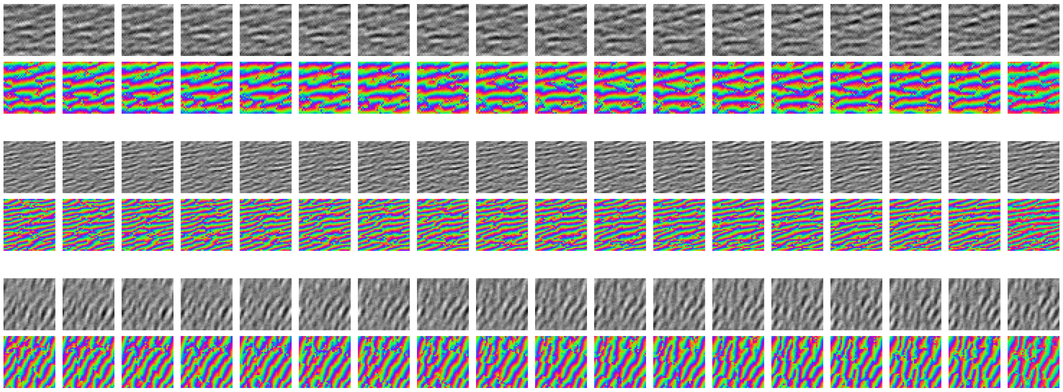

Figure 4: Visualization of the hidden state and phase for three models identical to those in Figure 1, but with different random initalizations. We see that the models learn different wavelengths and velocities depending on their initialization.

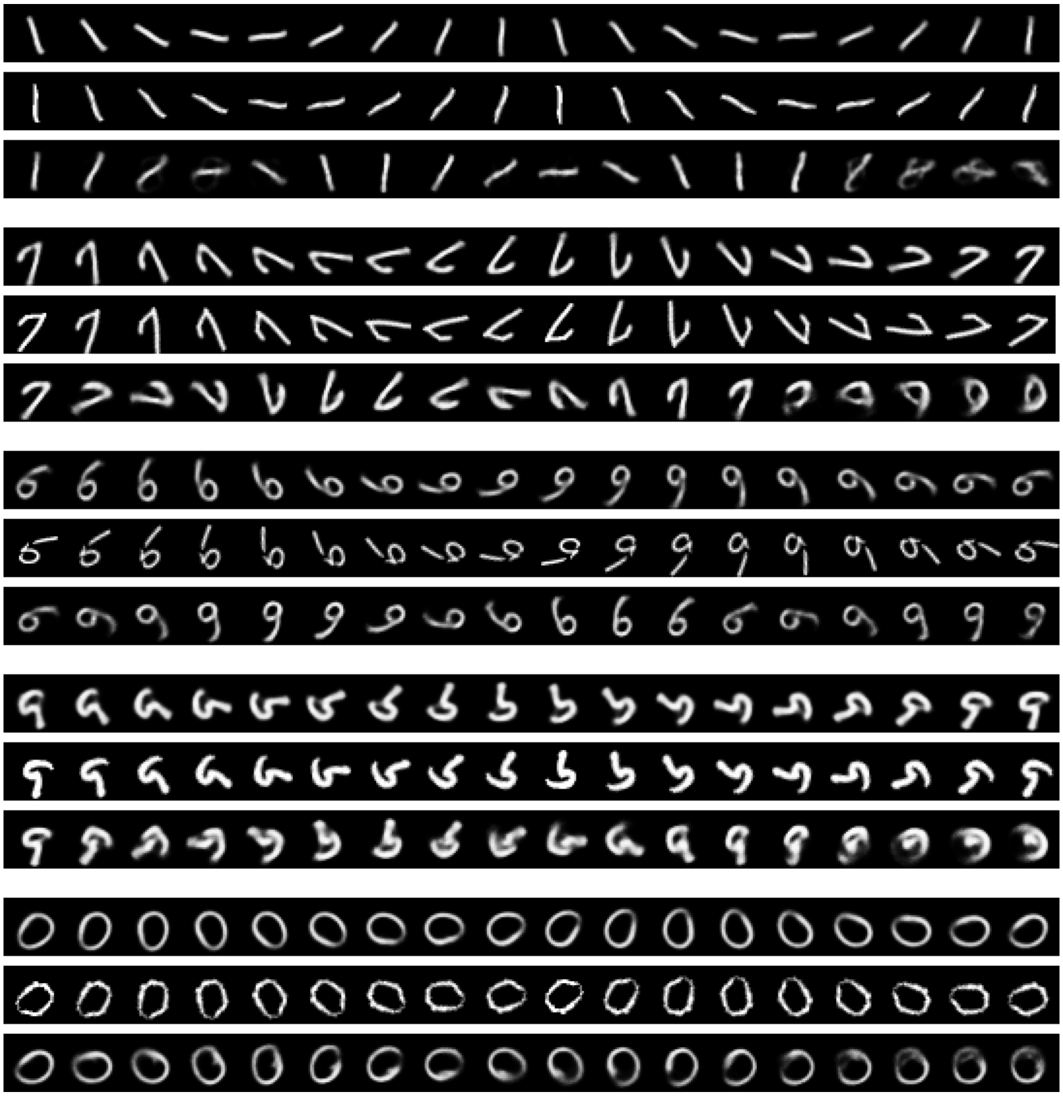

Figure 5: Additional visualizations of reconstructions from induced wave activity in the hidden state of the 1D TcoRNN as depicted in Figure 2. We show a set of random input sequences (top), the original model reconstruction (middle), and images generated by sequentially propagating the initial state backwards by an induced wave and decoding at each step (bottom). We see that, as in the main text, the assumed wave velocity of $v = 1$ is slightly faster than the actual velocity, and thus the reconstructed transformations are slightly faster than the input transformations. Because of this, we also observe that for certain examples, the induced wave reconstructions lose consistency with the input after the first period. This appears to imply that both the initial location of the wave activity matters in addition to its wave properties, and thus our model has learned to only propagate waves over parts of the feature space to optimize the capacity of the hidden state for this dataset. Finally, we observe that the induced transformations occur in reverse order due to the fact that our induced waves propagate in the reverse direction to those naturally exhibited for training examples, effectively propagating backwards in time.