SPONTANEOUS SYMMETRY BREAKING AND GOLDSTONE MODES FOR DEEP INFORMATION PROPAGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In physical systems, whenever a continuous symmetry is spontaneously broken, the system possesses excitations called Goldstone modes, which allow coherent information propagation over long distances and times. In this work, we study deep neural networks whose internal layers are equivariant under a continuous symmetry and may therefore support analogous Goldstone-like degrees of freedom. We demonstrate, both analytically and empirically, that these degrees of freedom enable coherent signal propagation across depth and recurrent iterations, providing a mechanism for stable information flow without relying on architectural stabilizers such as residual connections or normalization. In feedforward networks, this results in improved trainability and representational diversity across layers. In recurrent settings, we demonstrate the same mechanism is valuable for long-term memory by propagating information over recurrent iterations, thereby improving performance of RNNs and GRUs on long-sequence modeling tasks.

1 MOTIVATION

The flow of information through the layers of a neural network is central to deep learning and has been the topic of intense study. It is interesting to consider the layer direction to be analogous to "time" in a physical system. In that case, the problem of information flow through the network is mapped to the problem of creating stable structures in nature in a potentially turbulent environment. One can ask how this problem is solved in physical systems.

In fact, there is a universal physical principle which allows information to propagate coherently across long distances in space and time, that of *spontaneously broken symmetry*. Whenever the dynamics of a system are invariant under a symmetry operation, but the equilibrium configuration of the system is *not* invariant under the same symmetry, as shown in Figure 1, we say that the symmetry is *spontaneously broken*. As we briefly review in Appendix A, it is well-known that in situations of broken continuous symmetry the system generally possesses long-lived excitations called *Goldstone modes*, corresponding to coherent modulations of the equilibrium state by the symmetry operation (Goldstone et al., 1962).

Such Goldstone modes play a central role in physics, and there is a sense in which much of the long-distance information propagation in nature (e.g. sound (Leutwyler, 1997), spin waves in magnets (Sachdev, 1999), the pion exchange which mediates nuclear structure (Weinberg, 1979), even a modern understanding of light (Gaiotto et al., 2015; Lake, 2018; Hofman & Iqbal, 2018)) can be related to a spontaneously broken symmetry. In this work we consider a deep-learning analogue of this phenomenon. In particular, we demonstrate that in neural networks where *internal layers* are equivariant under a symmetry, degrees of freedom which are somewhat analogous to Goldstone modes propagate through the network, carrying information. These modes are outside the conventional "edge of chaos" paradigm for deep neural networks, which assumes no extra structure. We show that they lead to concrete advantages in neural network performance, allowing very deep networks to be trained with no extra mitigations (e.g. LayerNorm, skip connections, etc.) and providing benefits in the performance of recurrent neural networks on long-sequence tasks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

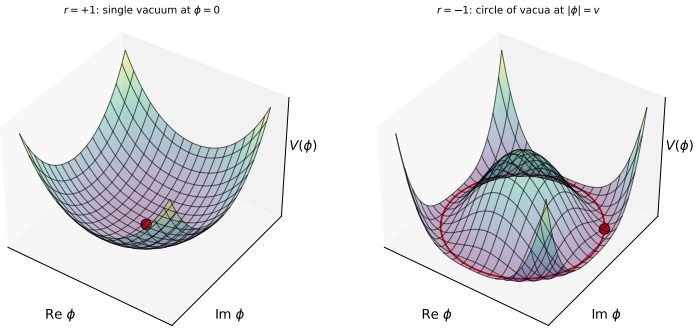


Figure 1: Equilibrium configurations are minima of the potential energy $V(\phi)$. If $r > 0$ we have a single isolated minimum at $\phi = 0$ and the symmetry is **unbroken**. If $r < 0$ we find a ring of minima with constant $|\phi|$, of which the system must choose one; the symmetry is **spontaneously broken**.

1.1 Goldstone modes in physics

We provide some very brief motivational remarks about symmetries and their spontaneous breaking in the form of an example. Consider a system described by a complex variable ϕ, whose potential energy is given by a function $V(|\phi|) = r|\phi|^2 + \lambda|\phi|^4$. Note this potential energy is invariant under the phase rotation $\phi \to e^{i\alpha}\phi$. The equilibrium configurations of this system are found by minimizing the potential energy. The structure of these low-energy configurations qualitatively differs depending on the sign of r. If $r > 0$, there is only one unique solution $\phi = 0$ which minimizes the energy. This configuration is invariant under the symmetry operation, and we say that the symmetry is **unbroken**. However if $r < 0$ then we find a *ring* of potential solutions with $|\phi| = \sqrt{\frac{-r}{2\lambda}}$. The solution must pick one point along this ring, and it is no longer invariant under the symmetry operation, which now takes us between solutions. In this situation we say that the symmetry is **spontaneously broken**.

The continuous family of low-energy solutions often results in dramatic changes to the long-time dynamics. In physics one often considers a situation where $\phi = \phi(t, x)$ is a *field* which depends on space and time, and then in the spontaneously broken phase one finds waves called **Goldstone modes** which propagate coherently for long distances due to their intimate relation with the symmetry. We provide a brief review in Appendix A. In this work we find analogous phenomena in the context of deep learning.

1.2 Prior work

Information propagation through the layers of a deep neural network, or over the timesteps of a recurrent network, is a very well-studied problem; see e.g. Poole et al. (2016); Schoenholz et al. (2017a); Yang & Schoenholz (2017); Xiao et al. (2018); Chen et al. (2018). We will extend this formalism to include equivariance, uncovering new dynamics. Our neural networks fit into a larger body of literature (Miyato et al., 2024; Liboni et al., 2023; Keller et al., 2023) that is inspired by physical systems and recent observations in neuroscience (Bhattacharya et al., 2022; Muller et al., 2018; Das et al., 2026). Group equivariance in neural networks (Cohen & Welling, 2016) is an extremely mature field (see e.g. the textbooks Weiler et al. (2023); Bronstein et al. (2021)), though generally it is related to a symmetry of the *task*: here our tasks have no symmetry, and we consider equivariance as a purely architectural tool for information propagation. See Kaba & Ravanbakhsh (2023); Smidt et al. (2021) for applications of symmetry breaking in the context of a given task in equivariant networks. In Löwe et al. (2022; 2024) equivariances similar to ours were used as an architectural tool, rather than as a symmetry of the data distribution.

Related work in the context of a *discrete* symmetry rather than a continuous one includes Iqbal & Welling (2025), where it was seen that topological defects play a role that is somewhat analogous to the Goldstone modes in our study. The advantages of a non-unique late-time attractor – similar to the symmetry orbits that we find – in the context of graph neural network oversmoothing were discussed in Turan et al. (2026). See also the recent work (Mo, 2026), which also discusses the influence of symmetry on dynamical stability.

2 MODEL AND THEORETICAL DESCRIPTION

We now describe our general framework for building deep neural networks which exhibit spontaneously broken symmetries, and provide preliminary analytical and empirical results that support the role of Goldstone modes in deep information propagation.

2.1 EQUIVARIANT INTERNAL LAYERS

We will consider deep neural networks with $L+2$ layers, of which the *bulk* of the network preserves a given symmetry group G. We denote the internal representation at layer l by $\mathbf{x}^l \in \mathbb{R}^N$. We will sometimes need to discuss the individual components of $\mathbf{x}^l$, which we denote as x_i^l with $i \in \{1, \cdots N\}$. Each internal state is acted on by a layer f^l as

$$\mathbf{x}^{l+1} = f^l(\mathbf{x}^l) \qquad f^l : \mathbb{R}^N \to \mathbb{R}^N \,. \tag{1}$$

We assume that $\mathbf{x}^l$ transforms under some representation of G. We construct the internal layers f^l to be equivariant under this transformation, i.e. for every $g \in G$ we have a representation matrix ρ_g such that

$$\rho_g f^l(\mathbf{x}^l) = f^l(\rho_g \mathbf{x}^l) \qquad l \in \{0, \cdots, L-1\}. \tag{2}$$

This is straightforwardly done by appropriately restricting f.

We now consider the input to the network $\mathbf{x}^{\text{in}}$ and the output $\mathbf{x}^{\text{out}}$. In most applications of group-equivariant neural networks (Cohen & Welling, 2016), G corresponds to a symmetry of the *task*, and thus both the input and output have a well-defined transformation under G. In our framework, on the other hand, we use G-equivariance only as a tool for propagating and processing information. It has nothing to do with the tasks that we will study. Thus the first and final layers of the network f^{in} and f^{out} are fully general and break equivariance. We will see that the equivariance will still be useful for propagating information through layers. Some examples of G that we will study include:

- $U(1)$: in this case it is convenient to consider the representation space to be $\mathbb{C}^N$, where the symmetry acts as
$$z_i \to e^{i\alpha} z_i \tag{3}$$
i.e. by a constant phase rotation on all the components of $\mathbf{z}$.
- $O(k)$: Here we consider a k which divides N and we consider $\mathbb{R}^N$ as a direct sum of N/k blocks of k-dimensions each, each of which is acted on by the $O(k)$ matrix. See App. §B.

One can imagine that the set of values acted on by G (i.e. the complex entries in the $U(1)$ case, or the k-vectors in the $O(k)$ case), form a *capsule* (Sabour et al., 2017), where in our case the capsule is defined in terms of its symmetry transformation.

2.2 PHASES OF SYMMETRY AND INFORMATION PROPAGATION

In this section we describe how symmetries affect information propagation. We use the $U(1)$ case for concreteness, but most of the results apply to any continuous symmetry group with the appropriate change of notation. In this section we disregard the non-equivariant input and output layers, and consider a $U(1)$-equivariant network whose updates take the form:

$$z_i^{l+1} = \sum_j W_{ij}^l \phi(z_j^l) \tag{4}$$

where we use a complex notation where at each layer l, $z_i^l \in \mathbb{C}$ and W_{ij}^l is a complex matrix. The non-linearity is picked to be equivariant under a phase rotation, e.g.

$$\phi(z) = \frac{\tanh(|z|)}{|z|} z \tag{5}$$

which maps the magnitude of the complex number $|z|$ to $\tanh(|z|)$ while preserving its phase. This network is equivariant under a $U(1)$ symmetry acting on the inputs as equation 3.[1]

[1]The theory that follows holds for any smooth and bounded activation function, though details such as the precise location of the phase transition line, the functional form of $c^\star$ etc. depend on the choice of activation.

In this work, we interpret information propagation through three complementary perspectives: (i) preservation of input-dependent degrees of freedom, (ii) non-vanishing components of the input–output Jacobian, and (iii) sustained correlations across layers. We will relate these features to the possible phases of deep information propagation in the presence of symmetry. We do this by studying the behavior of the network at initialization using the tools developed in Schoenholz et al. (2017a); Yang & Schoenholz (2017); Xiao et al. (2018); Chen et al. (2018).

We take each entry of W_{ij} at initialization to be an i.i.d. complex number where the real and imaginary parts have variance $\frac{\sigma_W^2}{N}$. We briefly recall the results of Schoenholz et al. (2017a) for a conventional feedforward network without any equivariance properties. The behavior of the network at initialization depends on the value of σ_W: if this is small then the network will tend to collapse all inputs to a single attractor, thereby removing all information of the inputs. On the other hand, if it is very large then it will result in chaotic behavior which *effectively* decorrelates inputs from outputs. Both of these behaviors result in poor training, essentially not allowing information to propagate further than a certain finite (calculable) depth scale ξ. It is argued in Schoenholz et al. (2017a) that for efficient training one should instead tune σ_W to the precise boundary between these two behaviors, i.e. to the *edge of chaos*, at which ξ diverges.

However the situation in the case of equivariance is fundamentally different. Consider performing a $U(1)$ rotation on an input $z_i^0 \to e^{i\alpha} z_i^0$ to an L-layer equivariant network as in equation 3. Due to the imposed equivariance, the action of this rotation must be faithfully transmitted all the way through the layers of the network, rotating the output $z_i^L \to e^{i\alpha} z_i^L$.

There are now two broad possibilities for what happens for very deep networks:

1. **Symmetry unbroken phase:** here we have $z_i^L \approx 0$, i.e. the network has forced all the inputs to zero. In this case the symmetry operation has no effect on the output.
2. **Symmetry spontaneously broken phase:** here we have $z_i^L \neq 0$, i.e. the final output has nonzero magnitude. In this case the symmetry necessarily acts non-trivially on the output.

The order parameter which distinguishes these two phases is:

$$c^l \equiv \frac{1}{N} \mathbb{E} \left(\sum_j \phi(z_j^l) \phi^\dagger(z_j^l) \right) \tag{6}$$

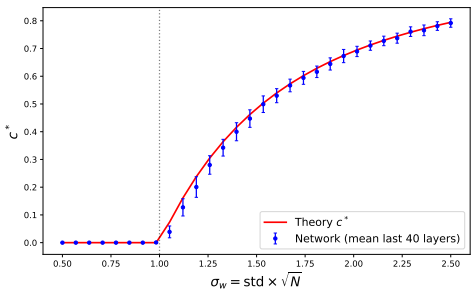


Figure 2: Comparison of the mean-field result for $c^\star$ from numerically solving equation 8 with an empirical result measured over an ensemble of $L = 100$, $N = 16$ neural networks at initialization. Note the clear phase transition at $\sigma_W = 1$.

i.e. the $U(1)$-invariant magnitude of each activation, averaged over the whole set, where the expectation value is measured over randomly sampled weights and some appropriate input distribution at initialization. If this $c^l \to 0$ as $l \to \infty$ we have the symmetry unbroken phase, and if it is finite then we have the Spontaneous Symmetry Broken (SSB) phase.

These two phases may appear to be in rough correspondence with the order and chaos phases in the usual discussion of deep information propagation Schoenholz et al. (2017a). A difference is that the attractor in the "order" phase is necessarily the only $U(1)$-invariant point (i.e. 0). Conversely, in the "chaos" phase the radius of each neuron (in expectation) approaches the calculable constant $\sqrt{c^l} > 0$, but the overall phase of the whole vector $\mathbf{z}$ is not set by the propagation over the layers, but rather by the $U(1)$ phase of the input. This is analogous to the choice of vacuum in the potential shown in the right panel of Figure 1. As a symmetry operation is faithfully transmitted through the network, the presence of equivariance then provides a stable path for some information to flow, stored in this overall phase. The presence of this channel may loosely be understood as a Goldstone mode.

To formalize the idea of the flow of information, in Appendix C.3 we show that a certain component of the Jacobian of the input-output map $\frac{\partial z^L}{\partial z^0}$ can be related to this order-parameter,

$$d^L = \frac{1}{N}\sum_i \mathbb{E}\left(\left|\sum_j \left(\frac{\partial z_i^L}{\partial z_j^0} z_j^0 - \frac{\partial z_i^L}{\partial z_j^{0\dagger}} z_j^{0\dagger}\right)\right|^2\right) \tag{7}$$

where d^l is the pre-activation analogue of c^l. This formula arises from considering how an infinitesimal symmetry transformation is propagated through the network. As we derive explicitly below, the left-hand side of this expression is $O(1)$ everywhere in the SSB phase: this means that a certain component of the Jacobian always remains $O(1)$, no matter how deep the network. This is non-trivial, as for vanilla networks these Jacobians tend to either collapse or explode. We will show below that this regularity seems to confer a concrete advantage in training very deep networks. See App. §C.4 for a discussion on how this relates to symmetry breaking in physics.

2.3 Large N analysis

We now demonstrate that some aspects of the dynamics can be understood theoretically at initialization. Consider supplying an input z_i^0 at the start of the network. We study how the distribution of these inputs changes due to the influence of the randomly chosen W_{ij} as we move through the network. In the large-N limit it is well-known that this evolution is given by a Gaussian process (Schoenholz et al., 2017a; Lee et al., 2017; Klambauer et al., 2017) We extend this calculation to our equivariant model in Appendix C, performing a self-contained derivation using the tools of stochastic path integrals. Here we provide only results.

Order parameter dynamics. In the large-N limit the distribution of activations is a Gaussian whose covariance c^l defined in equation 6 satisfies a well-defined evolution equation. The evolution of c^l through layers is given by equation 36 in the Appendix:

$$c^{l+1} = 2\int_0^\infty du\, ue^{-u^2}\left(\phi(u\sigma_W\sqrt{c^l})^2\right) \tag{8}$$

From now on for concreteness we will specialize to the choice of non-linearity $\phi = \tanh$. Then there is a clear solution at $c^l = 0$, corresponding to all activations collapsed at zero. To see if this solution is stable, it is instructive to expand out the right-hand side of equation 8 and perform the integral at small c^l to find

$$c^{l+1} = \sigma_W^2 c^l + \mathbb{O}((c^l)^2) \tag{9}$$

and thus we see that for $\sigma_W^2 < 1$, $c^l \to 0$ as we take l large. Here all activations are eventually zero and thus invariant under the symmetry: as described above this is the **unbroken symmetry phase**.

In contrast, for $\sigma_W^2 > 1$, the solution at $c^l = 0$ is unstable and the system heads to a non-trivial fixed point. It is possible to numerically solve equation 8 with $c^l = c^{l+1} = c^\star$ to find this fixed point, which we do in Figure 2, comparing it with the value of $c^\star$ obtained from many instantiations of the network at initialization, finding excellent agreement. This is the **symmetry spontaneously broken phase**. The phase transition is at $\sigma_W = 1$.

Two-input correlation. We now consider the covariance of two inputs $z_{a,b}^l$ where $a, b \in \{1, 2\}$. We now have two $U(1)$ symmetries: $U(1)_a \times U(1)_b$, and can construct the covariance matrix:

$$C_{ab}^l \equiv \frac{1}{N}\mathbb{E}\left(\sum_j \phi(z_{j,a}^l)\phi^\dagger(z_{j,b}^l)\right) = \begin{pmatrix} c^l & \Delta^l e^{i\varphi^l} \\ \Delta^l e^{-i\varphi^l} & c \end{pmatrix} \tag{10}$$

where the diagonal entries are the covariance c^l that has previously been studied[2] and the off-diagonal entries are now complex numbers which we write in terms of a magnitude Δ^l and phase φ^l. Using the same techniques as before we can write down a propagation equation for this covariance. There is now a crucial difference arising from the equivariance: we find that

$$\varphi^{l+1} = \varphi^l \tag{11}$$

[2]It is possible in principle to have $C_{11} \neq C_{22}$; this is set by the initial conditions, i.e. the ensemble from which z_1^0 and z_2^0 are drawn. Here we assume that $C_{11} = C_{22}$; the evolution will then not change this fact.

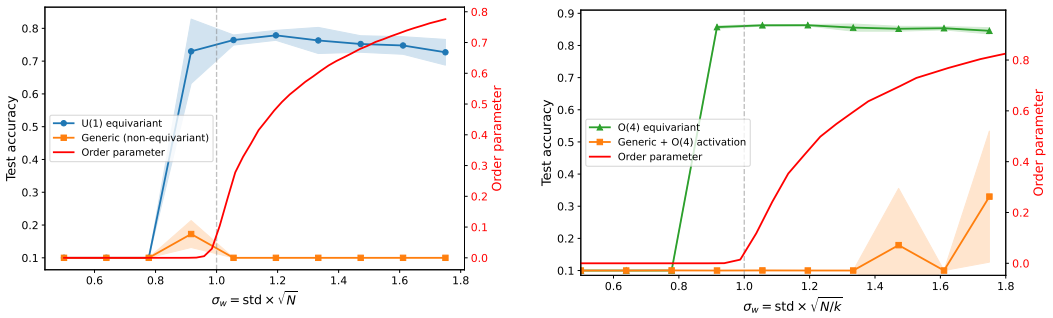


Figure 3: Test accuracy after 5 epochs on Fashion-MNIST for $U(1)$ (left) and $O(4)$ equivariant (right) models with 64 features and 100 layers. We compare models with the same nonlinearity but generic linear layers, isolating the effect of equivariance. We vary σ_w to probe the symmetry breaking phase transition, normalizing it so that the transition is at $\sigma_w = 1$. Training of equivariant models appears to begin precisely when we enter the SSB phase, as measured by the order parameter $c^\star$, which we compute at initialization from equation 6 averaged over the last 30 layers. Training of the non-equivariant models is poor, as expected for deep networks with no mitigation.

i.e. the *phase* of the 2-input covariance *does not change as we process through the layers*. This captures the idea that the $U(1)$ equivariance preserves relative angles between the inputs.

Formally, φ^l can be thought of as a Goldstone mode corresponding to the spontaneous breaking of $U(1)_a \times U(1)_b$ to a diagonal subgroup, as we explain in more detail in equation 43. In physics, time-evolution of Goldstone modes is highly constrained and the modes generally evolve slowly. In our setting the expression equation 11 can be thought of as an extreme version of this, where the mode turns out to be completely independent of time.

The evolution equation for the magnitude Δ is

$$\Delta^{l+1} = \frac{4(c^2 - \Delta^2)}{c^2} \int du_1\, du_2\, u_1 u_2 \exp\left(-(u_1^2 + u_2^2)\right) I_1\left(2u_1 u_2 \frac{\Delta}{c}\right) \phi(\frac{u_1}{\alpha})\phi(\frac{u_2}{\alpha}) \tag{12}$$

where to lighten the notation we do not make explicit that all quantities on the right-hand side are evaluated at l, where $\alpha \equiv \frac{\sqrt{c}}{\sigma_w \sqrt{c^2 - \Delta^2}}$, and where I_1 is a modified Bessel function. Given a solution for c^l this can be explicitly solved for Δ^l, and we can search for fixed points by searching for a solution with $\Delta^{l+1} = \Delta^l = \Delta^\star$.

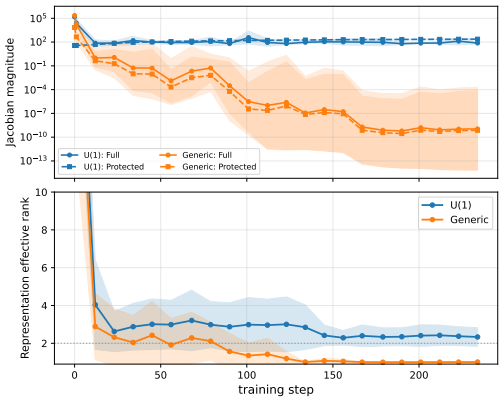
Interestingly, we do not find any non-trivial fixed points in Δ; on both sides of the phase transition of $\sigma_W = 1$ we find that the only stable fixed-point is $\Delta = 0$. At large l Δ approaches 0 as $\Delta^l \sim \exp\left(-\frac{l}{\xi_\Delta}\right)$, where the depth scale ξ_Δ can be computed from these expressions. In experiments we find that this depth scale is large enough that Δ^l remains nonzero over many tens of layers.

Although the magnitude Δ^l decays with depth, it does so sufficiently slowly to preserve relative phase information over many layers, while the phase remains exactly conserved due to equation 11. We thus believe that correlation information is propagated by the phase degree of freedom. [3]

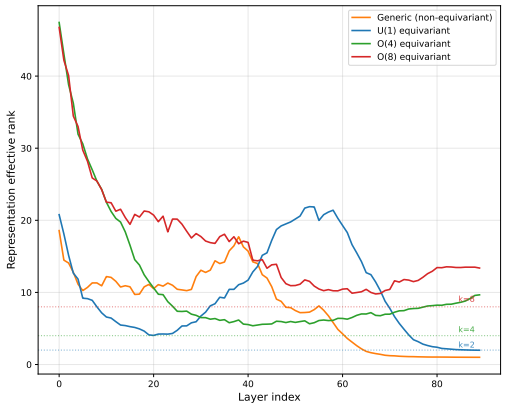
2.4 Empirical study of training dynamics

To summarize, we expect that very deep neural networks with $U(1)$ equivariance should train better provided we are in the symmetry spontaneously broken phase, i.e. that $\sigma_W > 1$. Empirically, this is precisely what we see in Figure 3. Unlike classical signal propagation theory, which requires fine-tuning to a critical point (the edge of chaos), here stable trainability emerges over a wide range of parameters provided we are in the symmetry-broken phase, see Figure 10 in the Appendix.

[3] It does not seem impossible that a refinement of the dynamics could result in a true non-trivial fixed point for Δ as $l \to \infty$; such a model would presumably have even better depth-generalization capacities, and is an interesting topic for future study.



(a) (Top) Norm of the input/output Jacobian and its protected component (equation 7) for $U(1)$ equivariant and generic models. (Bottom) Effective rank of the penultimate-layer representation during 1 training epoch.



(b) How effective rank of representation changes with layer for a very deep network trained on Fashion-MNIST. The generic network can collapse completely, but for equivariant networks the minimum rank is k for $O(k)$.

Figure 5: Effective rank dynamics during training (left) and across network depth (right).

We also consider enlarging the equivariance to larger groups such as $O(k)$: such larger groups allow more information to be stored in the Goldstone modes, and we see that they correspondingly have stronger large-depth performance, as shown in Figure 4.

It is interesting to further investigate the mechanism behind this improved performance. We believe that part of this arises from the coherent phase φ^l explained in subsection 2.3. It can also be understood in terms of the flow of gradients: in equation 7, we noted that a certain component of the Jacobian of the input-output map is *protected*, in that it is related to an order parameter which is nonzero in the SSB phase.

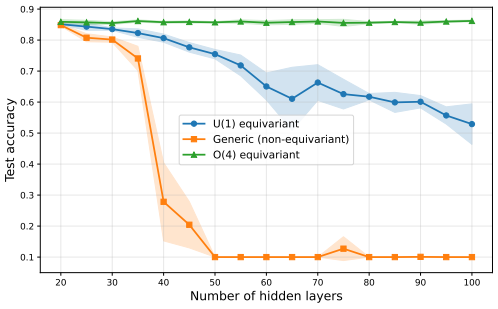


Figure 4: Performance on Fashion-MNIST vs. num layers for a generic network and equivariant networks. Generic networks degrade with depth while the $O(4)$ model does not.

In Figure 5a we plot the norm of the full Jacobian and its protected component along training for equivariant and generic models on Fashion-MNIST. Note that the Jacobian for the generic model collapses as we train; this appears to be related to the rank-collapse phenomenon reported in Daneshmand et al. (2020) when training a deep generic network without mitigations. However, the Jacobian for the equivariant model remains healthy: due to equation 7, the protected component *cannot* vanish, and this acts as a clean channel for information to flow. Relatedly, the rank of the representation in the equivariant model cannot vanish, as the minimum dimension of a representation on which the $U(1)$ symmetry can act non-trivially is 2. In Figure 5b we study this effective rank as we move through the layers for various symmetry groups. For $O(k)$ the dimension k of the fundamental representation always provides a floor on the effective rank of the representation. Thus we see that equivariance provides a new and very simple way to ensure representation diversity in deep networks.

3 Experimental Results

To study the practical implications of the information propagation capabilities afforded by spontaneously broken symmetries, we now consider tasks where significantly 'deeper' long-range correlations are natural. Specifically, we consider a suite of sequence modeling benchmarks from the liter-

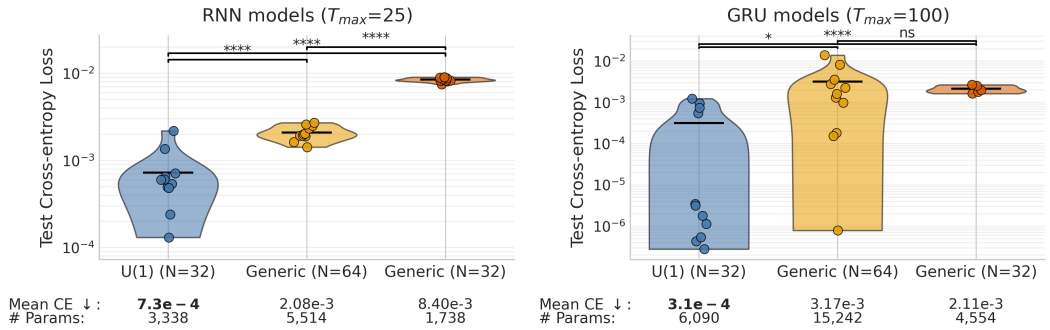


Figure 6: RNNs and GRUs versus $U(1)$ equivariant counterparts on copy tasks with $T_{max} = 25$ and $T_{max} = 100$ respectively. $U(1)$ models significantly outperform non-equivariant counterparts for both architectures, even when generic baselines have more real trainable params.

ature where long-time dependency modeling is critical: a simple synthetic long-delay copying task (Hochreiter & Schmidhuber, 1997), a more complex sequential image classification task, permuted sequential MNIST (Arjovsky et al., 2016), and a broader UEA time series classification test suite which includes 30 datasets of varying sequence lengths (Bagnall et al., 2018). Similar to the above Fashion-MNIST experiments, for each dataset we compare generic models to equivariant counterparts, but now reconsider the 'depth' over which information must propagate as recurrent iterations.

Models, Training and Evaluation. In the following, we use simple Recurrent Neural Networks (RNN) (Elman, 1990) and Gated Recurrent Unit networks (GRU) (Cho et al., 2014) as our baseline architectures. To construct the equivariant counterparts, we follow the same program as outlined in Section 2. Specifically, let $\mathbf{h}^t$ denote the hidden state at timestep t, with N counting capsules ($U(1)$: has $2N$ real coordinates; $O(k)$: has Nk), and let $\mathbf{u}^t = \mathbf{E}\mathbf{x}^t$ embed real inputs into this capsule space (e.g. via a learned map and reshaping). With $\mathbf{h}^0 = \mathbf{0}$, the equivariant RNNs for symmetry group $G \in \{U(1), O(k)\}$ are defined as

$$\mathbf{h}^t = \phi_G\big(\mathbf{u}^t + \mathbf{W}_G\mathbf{h}^{t-1}\big), \quad \text{with} \quad \mathbf{W}_{U(1)} \in \mathbb{C}^{N\times N}, \quad (\mathbf{W}_{O(k)}\mathbf{h}^t)_{\alpha a} = \sum_\beta \mathbf{A}_{\alpha\beta}\mathbf{h}^t_{\beta a}, \tag{13}$$

and $\mathbf{A} \in \mathbb{R}^{N\times N}$. For $G = U(1)$, $\mathbf{h}^t \in \mathbb{C}^N$ and ϕ_G is the radial nonlinearity from equation 5; for $G = O(k)$, $\mathbf{h}^t \in \mathbb{R}^{N\times k}$ has an additional k internal 'capsule' dimensions, ϕ_G acts on the $O(k)$-invariant radius of each block, and the recurrent map mixes capsules but acts isotropically within each capsule, i.e. the recurrent map has the form $\mathbf{A} \otimes I_k$, as detailed in Appendix B. For the $U(1)$-equivariant GRU, the gates are real, phase-invariant functions of the previous hidden state:

$$\begin{aligned} \tilde{\mathbf{h}}^t &= \phi_{U(1)}\big(\mathbf{u}^t + \mathbf{W}_h(\mathbf{r}^t \odot \mathbf{h}^{t-1})\big), & \mathbf{r}^t &= \sigma\big(\mathbf{W}_{xr}\mathbf{x}^t + \mathbf{W}_{hr}|\mathbf{h}^{t-1}| + \mathbf{b}_r\big), \\ \mathbf{h}^t &= \mathbf{z}^t \odot \mathbf{h}^{t-1} + (1 - \mathbf{z}^t) \odot \tilde{\mathbf{h}}^t, & \mathbf{z}^t &= \sigma\big(\mathbf{W}_{xz}\mathbf{x}^t + \mathbf{W}_{hz}|\mathbf{h}^{t-1}| + \mathbf{b}_z\big). \end{aligned} \tag{14}$$

Here $|\mathbf{h}|$ is computed elementwise, so the gates are invariant under the global $U(1)$ phase rotation, while the candidate hidden update remains equivariant. For readouts, we flatten the hidden state capsule dimensions and perform a linear map to the output space. For all models, we sweep over learning rates and input weight initializations; for the copy and psMNIST tasks, we additionally sweep over activation functions (except for baseline GRUs) and recurrent initializations. Model selection is performed on a validation set, and all results are reported on the test set. Parameter counts are reported in real trainable parameters, where complex weights count as two real parameters. Full details are provided in Appendix D.2.

Variable Delay Copy Task. We first study a variable-delay version of the standard copy task. Each example is a length-10 sequence of one-hot data tokens sampled from 8 data symbols, followed by $T - 11$ 'blank' symbols and a 'go' token. We sample $T \sim \mathrm{Unif}\{11, \ldots, T_{\max}\}$ independently for each example, requiring the 'go' token to trigger readout. The target sequence is blank at all timesteps except for the 10 positions following the 'go' token, where the model must

output the initially presented sequence. In Figure 6 (left), we compare $U(1)$ equivariant RNNs with $N = 32$ complex capsules (64 real coordinates) against non-equivariant RNNs with 32 or 64 real hidden units for $T_{\max} = 25$. The equivariant model significantly outperforms both baselines across 11 random seeds, including the 64-unit baseline with comparable real hidden dimensionality. In Figure 6 (right), we compare $U(1)$ equivariant GRUs with $N = 32$ complex capsules on a longer task ($T_{\max} = 100$) against non-equivariant GRUs with 32 or 64 real hidden units. Across 11 random seeds, the $U(1)$ GRUs solve the task more often, suggesting more stable training, and achieve lower mean loss with fewer real scalar parameters (6K vs. 15k). On $T_{\max} = 200$, we again find that with the same hyperparameter sweep, the only model to have a seed which solves the task is the $U(1)$ GRU, achieving a test cross-entropy loss of 3.9×10^{-7}.

Sequential Image Classification. We next study permuted sequential MNIST (Le et al., 2015; Arjovsky et al., 2016), which requires simultaneous long-sequence memory and information processing. Each image is presented one pixel at a time, with the digit class predicted from the final hidden state after 784 steps. We perform a fixed random permutation of the pixel order, shared over both training and testing, which reduces short-range spatial correlations and increases effective task length. As seen in Figure 7, the equivariant RNNs and GRUs consistently outperform their non-equivariant counterparts across all parameter ranges tested. In particular, an $O(k)$ equivariant RNN ($k = 24$) reaches the performance of a comparable GRU without internal gating. While not state-of-the-art, these models perform comparably to specialized long-sequence architectures such as IndRNN Li et al. (2018), Lipschitz RNN Erichson et al. (2021), coRNN Rusch & Mishra (2021), and LEM Rusch et al. (2022), which achieve 96% to 97% accuracy on the same task. The improvement for GRUs suggests that spontaneous symmetry breaking may also benefit other architectures, a possibility we leave to future work.

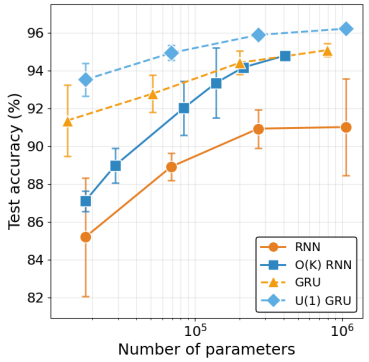


Figure 7: Test accuracy vs. # trainable params on psMNIST. Equivariant RNNs and GRUs consistently outperform non-equivariant models at all parameter ranges.

UEA Time Series Classification. We finally compare the same set of models on the 30 datasets from the UEA time-series classification test suite (Bagnall et al., 2018). In Table 1 we see that the average performance over all datasets is notably higher for the equivariant models compared with parameter matched counterparts. On a per-dataset basis, in Table 5 we see that O(4)-RNNs achieve a statistically significant accuracy improvement over the baseline RNNs on 14 of the 30 datasets, while the baseline RNN only achieves significant improvement on 2 of 30. The U(1)-GRU improves on 6 of 30, while the baseline GRU never achieves a significant win. Furthermore, on the longest sequence task of the benchmark (EigenWorms), containing sequences of length greater than 17,000 elements, we find a performance boost of more than 10 absolute percentage points for both model variants.

Table 1: Average model accuracy over all 30 UEA datasets, compared at matched parameter count. Full results in App. § E.

Model	Accuracy (%)
Tanh RNN	62.55
O(4)-RNN	**66.82**
GRU	66.29
U(1)-GRU	**68.04**

4 Conclusion

We have demonstrated that an analogue of Goldstone modes can lead to concrete benefits in information storage and training in deep networks. This leads to concrete benefits for RNNs, and interestingly allows for training very deep networks without the usual architectural heuristics such as LayerNorm, BatchNorm, skip connections, etc. We conclude by noting that the ideas here provide a new use of equivariance in neural networks – which was generally previously restricted to symmetries associated with the task or data distribution. We anticipate further applications in the future.

Limitations. While our current experiments are limited to simple benchmarks, they demonstrate the theory can apply beyond idealized analyses, including to trained neural networks in standard supervised learning settings, with the potential to improve performance in large-depth regimes.

AI USE STATEMENT

In this work, we used generative AI tools (i.e. the coding assistants Claude Code and Codex) to assist in writing code to run experiments and creating plots for the experimental sections of the paper, which includes creating very rudimentary synthetic data sets (e.g. sampling from Gaussians for Figure 2). We have not used generative AI tools for any of the other tasks with required disclosure in the ICLR 2027 AI Policy for Authors, i.e. we did not use it to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation, clean and reformat dataset, or to support qualitative and thematic data analysis.

Additionally, we used generative AI tools for minor proofreading and elimination of typos (both in text and in formulas) after the text of the paper was written by hand. We have reviewed all AI-assisted work, reviewing code and verifying results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

REPRODUCIBILITY STATEMENT

Code to reproduce all experiments in this paper can be found at the anonymous github repository: anonymous.4open.science/r/goldstone-information-propagation/. Additionally, all experiment protocols including dataset pre-processing, training, validation, and hyperparameter tuning are thoroughly detailed in Appendix D. Implementation of equivariant layers is described in Appendix B, and derivations of the theoretical results from Section 2, including the path integral description of information propagation through the layers of a feedforward network, are described in Appendix C.

REFERENCES

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *Proceedings of the 33rd International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1120–1128. JMLR.org, 2016.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018, 2018. URL https://arxiv.org/abs/1811.00075.

Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. In *International Conference on Learning Representations*, 2016.

Sayak Bhattacharya, Scott L. Brincat, Mikael Lundqvist, and Earl K. Miller. Traveling waves in the prefrontal cortex during working memory. *PLOS Computational Biology*, 18(1):1–22, 01 2022. doi: 10.1371/journal.pcbi.1009827. URL https://doi.org/10.1371/journal.pcbi.1009827.

Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

Paul M Chaikin and Tom C Lubensky. *Principles of condensed matter physics*, volume 10. Cambridge university press Cambridge, 1995.

Minmin Chen, Jeffrey Pennington, and Samuel Schoenholz. Dynamical isometry and a mean field theory of rnns: Gating enables signal propagation in recurrent neural networks. In *International Conference on Machine Learning*, pp. 873–882. PMLR, 2018.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Sidney Coleman. *Aspects of symmetry: selected Erice lectures*. Cambridge University Press, 1988.

A Crisanti and H Sompolinsky. Path integral approach to random neural networks. *Physical Review E*, 98(6):062120, 2018.

Hadi Daneshmand, Jonas Kohler, Francis Bach, Thomas Hofmann, and Aurelien Lucchi. Batch normalization provably avoids ranks collapse for randomly initialised deep networks. *Advances in Neural Information Processing Systems*, 33:18387–18398, 2020.

Anup Das, Erfan Zabeh, Bard Ermentrout, and Joshua Jacobs. Planar, spiral, and concentric traveling waves distinguish behavioral states in human memory. *Nature Communications*, 2026. doi: 10.1038/s41467-026-71386-z. URL https://doi.org/10.1038/s41467-026-71386-z.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. doi: 10.1207/s15516709cog1402_1.

N. Benjamin Erichson, Omri Azencot, Alejandro Queiruga, Liam Hodgkinson, and Michael W. Mahoney. Lipschitz recurrent neural networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-N7PBXqOUJZ.

Davide Gaiotto, Anton Kapustin, Nathan Seiberg, and Brian Willett. Generalized Global Symmetries. *JHEP*, 02:172, 2015. doi: 10.1007/JHEP02(2015)172.

Jeffrey Goldstone, Abdus Salam, and Steven Weinberg. Broken Symmetries. *Phys. Rev.*, 127:965–970, 1962. doi: 10.1103/PhysRev.127.965.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. 2014.

Albert Gu, Caglar Gulcehre, Tom Le Paine, Matt Hoffman, and Razvan Pascanu. Improving the gating mechanism of recurrent neural networks. 2020.

Mikael Henaff, Arthur Szlam, and Yann LeCun. Recurrent orthogonal networks and long-memory tasks. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2034–2042. PMLR, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Diego M. Hofman and Nabil Iqbal. Goldstone modes and photonization for higher form symmetries. 2018.

Nabil Iqbal and Max Welling. Topological defects propagate information in deep neural networks. In *NeurIPS 2025 AI for Science Workshop*, 2025. URL https://openreview.net/forum?id=fM5s2Tqe0t.

Sékou-Oumar Kaba and Siamak Ravanbakhsh. Symmetry breaking and equivariant neural networks. *arXiv preprint arXiv:2312.09016*, 2023.

T. Anderson Keller and Max Welling. Neural wave machines: Learning spatiotemporally structured representations with locally coupled oscillatory recurrent neural networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 16168–16189. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/keller23a.html.

T Anderson Keller, Lyle Muller, Terrence Sejnowski, and Max Welling. Traveling waves encode the recent past and enhance sequence learning. *arXiv preprint arXiv:2309.08045*, 2023.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2017. URL https://arxiv.org/abs/1412.6980.

Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *Advances in neural information processing systems*, 30, 2017.

Ethan Lake. Higher-form symmetries and spontaneous symmetry breaking. 2 2018.

Quoc V. Le, Navdeep Jaitly, and Geoffrey E. Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015. doi: 10.48550/arXiv.1504.00941.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

H. Leutwyler. Phonons as goldstone bosons. *Helv. Phys. Acta*, 70:275–286, 1997.

S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao. Independently recurrent neural network (indrnn): Building a longer and deeper rnn. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5457–5466, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00572. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00572.

Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3367–3375, 2015.

Luisa HB Liboni, Roberto C Budzinski, Alexandra N Busch, Sindy Löwe, Thomas A Keller, Max Welling, and Lyle E Muller. Image segmentation with traveling waves in an exactly solvable recurrent neural network. *arXiv preprint arXiv:2311.16943*, 2023.

Sindy Löwe, Phillip Lippe, Maja Rudolph, and Max Welling. Complex-valued autoencoders for object discovery. *arXiv preprint arXiv:2204.02075*, 2022.

Sindy Löwe, Francesco Locatello, and Max Welling. Binding dynamics in rotating features. *arXiv preprint arXiv:2402.05627*, 2024.

Paul Cecil Martin, Eric D Siggia, and Harvey A Rose. Statistical dynamics of classical systems. *Physical Review A*, 8(1):423, 1973.

Takeru Miyato, Sindy Löwe, Andreas Geiger, and Max Welling. Artificial kuramoto oscillatory neurons. *arXiv preprint arXiv:2410.13821*, 2024.

Hanson Hanxuan Mo. Symmetry-protected lyapunov neutral modes in equivariant recurrent networks. *arXiv preprint arXiv:2605.03338*, 2026.

Lyle Muller, Frédéric Chavane, John Reynolds, and Terrence J. Sejnowski. Cortical travelling waves: mechanisms and computational principles. *Nature Reviews Neuroscience*, 19(5):255–268, 2018. doi: 10.1038/nrn.2018.20. URL https://doi.org/10.1038/nrn.2018.20.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Michael E. Peskin and Daniel V. Schroeder. *An Introduction to quantum field theory*. Addison-Wesley, Reading, USA, 1995. ISBN 978-0-201-50397-5, 978-0-429-50355-9, 978-0-429-49417-8. doi: 10.1201/9780429503559.

Pedro O. Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 82–90. PMLR, 2014.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. *Advances in neural information processing systems*, 29, 2016.

Ramamurti Rajaraman. Solitons and instantons. an introduction to solitons and instantons in quantum field theory. 1982.

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.

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.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017.

Subir Sachdev. Quantum phase transitions. *Physics world*, 12(4):33, 1999.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. In *International Conference on Learning Representations*, 2017a. URL `https://openreview.net/forum?id=H1W1UN9gg`.

Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. A correspondence between random neural networks and statistical field theory. *arXiv preprint arXiv:1710.06570*, 2017b.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Tess E Smidt, Mario Geiger, and Benjamin Kurt Miller. Finding symmetry breaking order parameters with euclidean neural networks. *Physical Review Research*, 3(1):L012002, 2021.

Haim Sompolinsky, Andrea Crisanti, and Hans-Jurgen Sommers. Chaos in random neural networks. *Physical review letters*, 61(3):259, 1988.

Erkan Turan, Gaspard Abel, Maysam Behmanesh, Emery Pierson, and Maks Ovsjanikov. Beyond relu: Bifurcation, oversmoothing, and topological priors. *arXiv preprint arXiv:2602.15634*, 2026.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Yu S. Philip. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. Equivariant and coordinate independent convolutional networks. *A Gauge Field Theory of Neural Networks*, 110, 2023.

Steven Weinberg. Phenomenological Lagrangians. *Physica A*, 96(1-2):327–340, 1979. doi: 10.1016/0378-4371(79)90223-1.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International conference on machine learning*, pp. 5393–5402. PMLR, 2018.

Yiben Xu, Xian Long, Jianfeng Feng, and Pulin Gong. Interacting spiral wave patterns underlie complex brain dynamics and are related to cognitive processing. *Nature human behaviour*, 7(7): 1196–1215, 2023.

Ge Yang and Samuel Schoenholz. Mean field residual networks: On the edge of chaos. *Advances in neural information processing systems*, 30, 2017.

A BACKGROUND ON GOLDSTONE MODES

To make this work self contained we provide a brief review of Goldstone modes in physics. This material is standard, see e.g. Peskin & Schroeder (1995) for a textbook treatment.

Let us imagine an interacting *complex scalar field* ϕ, i.e. a field $\phi(t, x)$ which depends on space and time through an equation of the following form:

$$\left(\partial_t^2 - \partial_x^2\right)\phi = -\frac{dV}{d\phi^\dagger} \tag{15}$$

where we take the *potential energy* $V = r|\phi|^2 + \lambda|\phi|^4$. This system has a $U(1)$ symmetry: e.g. given a solution to the equations of motion equation 15, one can find a new solution by rotating ϕ by a constant phase: $\phi(t, x) \to e^{i\alpha}\phi(t, x)$.

Now the dynamical behavior of this field depends sensitively on the sign of the parameter r, as we show in Figure 1. If $r > 0$, then the configuration on which V is minimized (i.e. the *equilibrium* configuration) is $\phi = 0$. Note that this equilibrium is *invariant* under the $U(1)$ symmetry; for this reason the system with $r > 0$ is sometimes called the phase with **unbroken symmetry**. One can study small excitations around this equilibrium. If we consider a small perturbation with spacetime dependence $e^{-i\omega t+ikx}$, then by linearizing the equations of motion about $\phi = 0$ and plugging in the above ansatz, we can show that the oscillation frequency of the wave behaves as:

$$\omega = \sqrt{k^2 + r}\,. \tag{16}$$

which means that even a very long wavelength wave has a finite oscillation frequency $\omega \sim \sqrt{r}$. Relatedly, this means that wave packets will eventually disperse and dissipate over long distances, as one can see in the left panel of Figure 8.

We can also consider the system with $r < 0$. Now the lowest energy configuration is no longer unique: the energy is minimized on a *ring*, i.e. everywhere that $|\phi| = \sqrt{\frac{-r}{2\lambda}}$, as we show in the right panel of Figure 1. All points along this ring – related by a $U(1)$ rotation – have the same energy, and thus the system will simply choose one of them: this is called **spontaneous symmetry breaking**. We can again now consider small perturbations around this equilibrium. However low energy fluctuations now only modulate the phase of the field, i.e. $\phi(x, t) = \sqrt{\frac{-r}{2\lambda}}e^{i\theta(x,t)}$. Again finding the dispersion relation of small-amplitude oscillations, one finds that the frequency of the excitation now behaves as

$$\omega = |k| \tag{17}$$

i.e. the oscillation frequency of the wave goes to zero as the wavelength becomes very long. This happens because ω is related to the energy of the excitation – but a spatially homogeneous excitation with $k \to 0$ is now just a symmetry action on the equilibrium configuration, which costs no energy, and thus the frequency of the excitation must vanish. Furthermore, the linear dependence of the energy on the wavenumber means that there is no broadening of the wavefront, and the wave travels for long distances with no dissipation, as we can see in the right panel of Figure 8.

This is a specific example of a very general principle, called **Goldstone's theorem** (Goldstone et al., 1962): when a system has a symmetry which is spontaneously broken by the equilibrium configuration, then it has *long-lived* excitations called *Goldstone modes*, of which the wave in equation 17 is an example.

In this work we will demonstrate a similar principle at play in deep learning. In particular, we will show that when a neural network has internal layers which are equivariant under a global symmetry, there is a region in hyperparameter space (analogous to the region $r < 0$) where one can imagine that a symmetry is spontaneously broken, and that its corresponding Goldstone mode allows information to propagate coherently through the network.

B IMPLEMENTATION OF EQUIVARIANT LAYERS

Here we describe how equivariant layers such as those described in equation 2 are implemented.

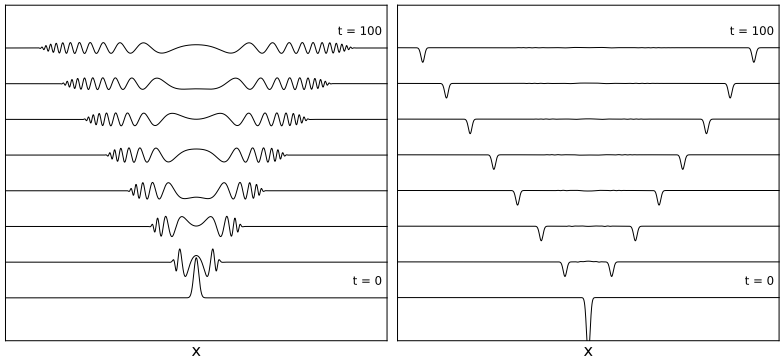


Figure 8: Excitations in symmetry unbroken (left) and symmetry spontaneously broken (right) phases respectively. We plot the time evolution of the real part of the complex scalar field described in equation 15. On left the wave-packet spreads apart and degrades; on the right we see the Goldstone mode propagating coherently across long distances.

$\mathbf{O(k)}$: We begin by studying equivariance under the more general $O(k)$ symmetry. We consider a representation $\mathbf{x} \in \mathbb{R}^{\mathbf{N}\times\mathbf{k}}$. We consider acting on the k index with the $O(k)$ symmetry, i.e. letting α, β run over N blocks and a, b run over the k components of the block, the symmetry acts as

$$x'_{\alpha a} = O_{ab} x_{\alpha b} \tag{18}$$

where $O_{ab} \in O(k)$ is an $O(k)$ rotation, i.e. it rotates the components of each block within itself but does not mix the blocks. We would now like to build a neural network layer which is equivariant under the symmetry.

This is done by considering a weight matrix W which takes the form

$$W_{\alpha a;\beta b} = w_{\alpha\beta}\delta_{ab} \tag{19}$$

i.e. it acts as the identity within each block but mixes the blocks together in an unconstrained manner. This clearly commutes with the general $O(k)$ rotation in equation 18; by Schur's lemma it is also the most general such matrix provided $O(k)$ is non-Abelian. Note that we have only N^2 free parameters, labeled by $w_{\alpha\beta}$, as opposed to the $(Nk)^2$ we could have had for the non-equivariant model.

We also consider a non-linearity which acts on the $O(k)$-invariant radius of each block, i.e.

$$\phi(x)_{\alpha a} = \frac{\tanh(|x_\alpha|)}{|x_\alpha|} x_{\alpha a} \tag{20}$$

where $|x_\alpha| \equiv \sqrt{\sum_{a=1}^{k} x_{\alpha a}^2}$, i.e. it replaces the radius of each block with its hyperbolic tangent while preserving its $O(k)$ orientation. Clearly one can replace $\tanh$ with any nonlinear function of one variable while preserving the equivariance properties. Generally one $O(k)$ layer consists of a single nonlinearity and a single linear layer.

$\mathbf{SO(2) = U(1)}$: The case of $SO(2)$ (or equivalently $U(1)$) is different. In this case the components a, b run over $1, 2$ and the most general choice for equation 19 is

$$W_{\alpha a;\beta b} = (w_{\alpha\beta}\delta_{ab} + u_{\alpha\beta}\epsilon_{ab}) \tag{21}$$

where now ϵ_{ab} is the fully antisymmetric two dimensional matrix and we now have two sets of trainable parameters $w_{\alpha\beta}$ and $u_{\alpha\beta}$. (Note if we considered $O(2)$ instead then the term in ϵ_{ab} would not be allowed, analogous to the case in equation 19 – e.g. the term in ϵ_{ab} is not operation under the group operation $\mathrm{diag}(-1, 1) \in O(2)$). This is equivalent after a change of basis to considering a complex representation $\mathbf{z} \in \mathbb{C}^N$ and allowing the most general complex W acting as equation 4. In this notation the $U(1)$ equivariance comes from the fact that the forward linear map is purely holomorphic (i.e. depending on z and not $z^\dagger$).

C PATH INTEGRAL FOR STOCHASTIC SYSTEMS

Here we present details of the path integral description of information propagation through the layers of a feedforward network. Path integrals provide a convenient calculational framework for discussing stochastic evolution equations Martin et al. (1973), allowing for controlled derivation of the mean-field expansion, along with departures from it. In the context of neural networks these techniques were used in the classic work Sompolinsky et al. (1988) and have since been revisited in the modern context in Crisanti & Sompolinsky (2018); Schoenholz et al. (2017b). To the best of our knowledge they have not been used to study group equivariant neural networks until this work.

C.1 $U(1)$ EQUIVARIANT FEEDFORWARD NETWORK

We will present our main derivation for the $U(1)$ equivariant version. The generalization to other symmetry groups is mostly a change of notation. In particular, we consider a fully connected network with the following update rule:

$$z_i^{l+1} = \sum_j W_{ij}^l \phi(z_j^l) \tag{22}$$

where now $z \in \mathbb{C}^N$, $W \in \mathbb{C}^{N\times N}$, and the nonlinearity (and thus the entire network) is $U(1)$ equivariant, in that $\phi(e^{i\theta}z) = e^{i\theta}\phi(z)$. For concreteness we take the nonlinearity to be:

$$\phi(z) = \frac{\tanh(|z|)}{|z|} z \tag{23}$$

which replaces the magnitude of the complex number by its hyperbolic tangent.

The primary object of study is the *partition function*, which is a functional of the applied "external fields" h_i^l.

$$Z[h] = \int [dz][dW] p(z^L|z^{L-1}) p(z^{L-1}|z^{L-2}) \cdots p(z^1|z^0) \prod_l p(W^l) \prod_l e^{h_i^l z_i^l + h.c.} \tag{24}$$

Here the integral takes place over all internal representations $[dz]$ as well as the initial ensemble of the inputs, as well as over all the weights $[dW]$. The official purpose of this object is that we can differentiate it with respect to the fields h_i^l to obtain moments of the z_i^l; each derivative of the exponential brings down a factor of the corresponding z_i^l, and we find (e.g.)

$$\frac{\partial}{\partial h_i^l} \frac{\partial}{\partial h_j^{l,\dagger}} \log Z[h] \bigg|_{h=0} = \mathbb{E}[z_i^l (z_j^l)^\dagger] \tag{25}$$

In practice however it is usually sufficient to know that this is possible, and we do not need to carry around the h fields explicitly. Instead the utility of Z is that its construction in terms of integrals directly shows the correlation structure of the activations. For now we will set h to zero for notational simplicity and work with $Z \equiv Z[h=0]$, and we can always restore it if needed.

Next, we assume as usual that at initialization each entry of the W_{ij}^l at each level is i.i.d. Gaussian with variance $\frac{\sigma_W^2}{N}$. Once the W are fixed the evolution of the z_i^l is deterministic according to equation 22, and we find the following iterated integral representing how each input evolves through the network.

$$Z = \int [dz] dW e^{-\frac{N}{\sigma_w^2} \sum_{ij,l} W_{ij}^l W_{ij}^{l\dagger}} \prod_{i,l} \delta \left(z_i^{l+1} - \sum_j W_{ij}^l \phi(z_j^l) \right) \tag{26}$$

To obtain a more useful representation, we write the delta function in terms of an auxiliary field $\hat{z}_i$ introduced at each layer, i.e.

$$\delta \left(z_i^{l+1} - \sum_j W_{ij}^l \phi(z_j^l) \right) = \int d\hat{z}_i^l d\hat{z}_i^{l\dagger} \exp \left(i\hat{z}_i^{l+1} \left(z_i^{l+1} - \sum_j W_{ij}^l \phi(z_j^l) \right) + h.c. \right) \tag{27}$$

(where various factors of 2π have been absorbed into the definition of the measure $[dz]$, as they will not affect our conclusions). We thus end up with the following representation

$$Z = \int [dz d\hat{z} dW] e^{S[z,\hat{z},W]} \tag{28}$$

$$S = -\frac{N}{\sigma_w^2} \sum_{ij} W_{ij}^l W_{ij}^{l\dagger} + i \sum_{l,i} \hat{z}_i^{(l+1)\dagger} \left(z_i^{l+1} - \sum_j W_{ij}^l \phi(z_j^l \right) + h.c. \tag{29}$$

where S is called the *effective action*.

We now note that W_{ij}^l appears in the action quadratically. Thus its integral is a Gaussian and can be explicitly done, leaving us with the following effective action:

$$S = i \sum_{l,i} \hat{z}_i^l z_i^l - \frac{\sigma_W^2}{N} \sum_{l,i,j} \hat{z}_i^{l+1} \hat{z}_i^{(l+1)\dagger} \phi(z_j^l) \phi(z_j^{l\dagger}) \tag{30}$$

We have now taken into account the integral over the W's: the resulting action has a very specific form, in that each component z_i interacts with every other component through the quartic term. This pattern of interactions is exactly that which is well described by mean-field theory, and it suggests that dynamics can be cleanly written in terms of collective variables whose distributions are sharply peaked in the large N limit.

To that end, we introduce the following collective field

$$c^l = \frac{1}{N} \sum_j \phi(z_j^l) \phi(z_j^l) \tag{31}$$

To express the dynamics in terms of this collective field, we insert 1 into the path integral in the following form:

$$1 = \int [dc d\hat{c}] \exp \left[\sum_l iN\hat{c}^l \left(c^l - \frac{1}{N} \sum_j \phi(z_j^l) \phi(z_j^l)) \right) \right] \tag{32}$$

where $\hat{c}^l$ is yet another field we have introduced. Its integral acts as a delta function which enforces the relation equation 31; thus we may replace $\frac{1}{N} \sum_j \phi(z_j^l) \phi(z_j^l)$ with c^l in the effective action, leaving us with

$$S = i \sum_l \left[\sum_i \hat{z}_i^l z_i^l - \sigma_W^2 c^l \sum_i \hat{z}_i^{l+1} \hat{z}_i^{(l+1)\dagger} + iN\hat{c}^l \left(c^l - \frac{1}{N} \sum_j \phi(z_j^l) \phi(z_j^l) \right) \right] \tag{33}$$

Now the integral over the $\hat{z}_i^l$ is Gaussian; performing the integrals we obtain

$$S = \sum_l \left[\frac{1}{\sigma_W^2 c_l} \sum_i z_i^{l+1} (z_i^{l+1})^\dagger + iN\hat{c}^l \left(c^l - \frac{1}{N} \sum_j \phi(z_j^l) \phi(z_j^l) \right) - N \log \sigma_W^2 c^l \right] \tag{34}$$

where the last term arises from the determinant of the Gaussian integral and where (as usual for path-integrals) we ignore constant additive shifts in the action.

So far everything we have done has been exact. We now consider the large N limit. Note that both of the terms in this action scale as N. This indicates that the theory becomes classical in the large N limit. As usual then we can evaluate the functional integral over $[dc d\hat{c}]$ in a saddle-point approximation by varying this action and demanding that it be stationary. The equation of motion for $\hat{c}^l$ simply tells us that c^l satisfies equation 31. Thus we see that each component of the z's has an identical Gaussian distribution, where the width c_l of the distribution at layer $l+1$ is determined by the product of activations at the previous layer in equation 31.

This is a self-consistent evolution equation for c^l:

$$\frac{1}{2\pi i \sigma_w^2 c^l} \int dz dz^\dagger \exp \left(-\frac{1}{\sigma_w^2 c^l} z^\dagger z \right) |\phi(z)|^2 = c^{l+1} \tag{35}$$

It is convenient to go to polar coordinates $z = re^{i\theta}$. The integral over θ can be done trivially, and we rewrite in terms of $u = \frac{r}{\sigma_W \sqrt{c}}$ so that the equation becomes

$$2 \int duue^{-u^2} [\phi(u\sigma_w \sqrt{c^l})]^2 = c^{l+1} \tag{36}$$

This equation is the $U(1)$-equivariant analogue of Eq (3) in Schoenholz et al. (2017a). The implications of this evolution equation are described around equation 8 in the main text. To understand the dynamics near the transition we specialize to the case $\phi = \tanh$ and expand the integrand side in powers of c^l and perform the integral order by order, finding:

$$c^{l+1} = c^l \sigma_w^2 - \frac{4(c^l)^2 \sigma_w^4}{3} + \mathbb{O}(c_l^3) \tag{37}$$

First considering c^l very small so that we may ignore the quadratic term, we see that for $\sigma_w < 1$ there is an attractive fixed point at $c^l \to 0$. For $\sigma_w > 1$ this solution is now unstable and c^l is repelled to a nonzero value. We may search for the non-trivial fixed point near the phase transition by setting $c^l = c^{l+1} = c^\star$, finding:

$$c^\star(\sigma_w \to 1) \approx \frac{3}{4}(\sigma_w^2 - 1) \tag{38}$$

i.e. a mean-field exponent.

C.2 Two-replica solution

We would now like to understand how the distribution of *two* inputs changes as we move through the network. Denoting these inputs as $z_{i,a}$ and $z_{i,b}$ where $a, b \in \{1, 2\}$, we essentially have twice as many dynamical fields – or *replicas* – with a diagonal term in the action describing their dynamics as

$$S \supset \exp\left[i \sum_{l,i,a} \hat{z}_{i,a}^{l+1} \left(z_{i,a}^l(t+1) - \sum_j W_{ij} \phi(z_{j,a}^l) \right) + h.c. \right] \tag{39}$$

We do not repeat the full derivation here; it essentially proceeds exactly as above, except that integrating out the W_{ij}'s now creates a coupling between replicas, and we should now consider the following matrix-valued collective field:

$$C_{ab}^l = \frac{1}{N} \sum_j \phi^\dagger(z_{j,a}^l) \phi(z_{j,b}^l) \tag{40}$$

The important term of the resulting two-replica action is that controlling the distribution of the z_a^i which is a simple generalization of equation 34:

$$S = \sum_l \left[\frac{1}{\sigma_W^2} (C^l)_{ab}^{-1} \sum_{i,a,b} z_{i,a}^{l+1} (z_{i,b}^{l+1})^\dagger + \cdots \right] \tag{41}$$

It is helpful to explicitly write down the components of C_{ab} as:

$$C_{ab}^l = \begin{pmatrix} c^l & \Delta^l e^{i\varphi^l} \\ \Delta^l e^{-i\varphi^l} & c^l \end{pmatrix} \tag{42}$$

where the diagonal correlation is precisely the c^l studied above and we have decomposed the off-diagonal correlation into a magnitude Δ^l and phase φ^l.

Note that the symmetry group of equation 41 is $U(1)_a \times U(1)_b$. Formally speaking, provided $\Delta \neq 0$, the *mixed* symmetry is spontaneously broken and ϕ can be thought of as its Goldstone mode, as we have the following transformation that leaves the action invariant:

$$z_{i,a}^l \to e^{i\alpha_a} z_{i,a}^l \qquad z_{i,b}^l \to e^{i\alpha_b} z_{i,b}^l \qquad \varphi^l \to -\alpha_a + \alpha_b \tag{43}$$

This nonlinear shift is precisely what defines a Goldstone mode. On general grounds one might then expect the time-dependence of this mode to be slow. We see an extreme version of this: ϕ^l is actually *time-independent*, i.e.

$$\varphi^{l+1} = \varphi^l \tag{44}$$

We will verify this explicitly in a calculation, but it can immediately be seen by realizing that a shift in $\phi^{l=0}$ may be obtained by performing a $U(1)$ rotation of z_a as an input. This $U(1)$ rotation must then propagate through each layer by equivariance, rotating the off-diagonal component C^l_{ab} in precisely the same way at each layer.

We now derive an equation for the evolution of C^l_{ab}. Explicitly, the inverse is:

$$[C^{-1}]_{ab} = \frac{1}{c^2 - \Delta^2} \begin{pmatrix} c & -\Delta e^{i\varphi} \\ -\Delta e^{-i\varphi} & c \end{pmatrix} \tag{45}$$

The generalization of equation 35 now becomes:

$$C^{l+1}_{ab} = \frac{1}{\det C^l (\pi \sigma_w^2)^2} \int dZ dZ^\dagger \exp\left(-\frac{1}{\sigma_w^2} Z^\dagger (C^l)^{-1} Z\right) \phi^\dagger(z_a)\phi(z_b) \tag{46}$$

where we have combined the two replicas into one vector $Z = (z_1, z_2)$. Change variables as $z_a = r_a e^{i\theta_a}$ and then we have

$$Z^\dagger C^{-1} Z = \frac{1}{c^2 - \Delta^2} \left(c^2(r_1^2 + r_2^2) - 2\Delta r_1 r_2 \cos(\varphi + \theta_1 - \theta_2)\right) \tag{47}$$

We now evaluate equation 46 at the interesting off-diagonal point $a = 1, b = 2$.

We first track the evolution of the phase ϕ^l. On the right-hand side it appears only in the exponent through equation 47; we can remove it from here by shifting $\theta_1 \to \theta_1 - \varphi^l$. The integration measure is invariant under this change of variables and the explicit appearance of $\phi^\dagger(z_1)$ transforms equivariantly; the right-hand side thus contains an explicit factor of $e^{i\varphi^l}$.

We now perform the angular integrals. Define the angles $\delta = \theta_1 - \theta_2$ and $\Sigma = \frac{1}{2}(\theta_1 + \theta_2)$. The integral over Σ factors out, and we find an integral over δ which can be done using Bessel functions. Ultimately only the radial integrals remain, and we find:

$$\begin{aligned} \Delta^{l+1} e^{i\varphi^{l+1}} = \frac{4 e^{i\varphi^l}}{(c^2 - \Delta^2)\sigma_w^4} \int dr_1 dr_2 r_1 r_2 \exp\left(-\frac{c}{\sigma_w^2 (c^2 - \Delta^2)}(r_1^2 + r_2^2)\right) \\ I_1\left(2 r_1 r_2 \frac{\Delta}{\sigma_w^2 (c^2 - \Delta^2)}\right) \tanh(r_1) \tanh(r_2) \end{aligned} \tag{48}$$

Note, as claimed, that we have that $\varphi^{l+1} = \varphi^l$: the Goldstone mode does not evolve across the layers. The remainder of the integral is an evolution equation for the real degree of freedom Δ^l. To make this integral easier to evaluate we perform the change of variables

$$u_a = \frac{\sqrt{c}}{\sigma_w \sqrt{c^2 - \Delta^2}} r_a \equiv \alpha r_a \tag{49}$$

after which we have:

$$\Delta^{l+1} = \frac{4(c^2 - \Delta^2)}{c^2} \int du_1 du_2 u_1 u_2 \exp\left(-(u_1^2 + u_2^2)\right) I_1\left(2 u_1 u_2 \frac{\Delta}{c}\right) \tanh\left(\frac{u_1}{\alpha}\right) \tanh\left(\frac{u_2}{\alpha}\right) \tag{50}$$

where everything on the right-hand side is evaluated at layer l.

C.3 Jacobian of input-output map

Here we demonstrate that the following order-parameter

$$d^l \equiv \frac{1}{N} \sum_i \mathbb{E}(z_i^l z_i^{l\dagger}) \tag{51}$$

i.e. essentially the same information as c^l in equation 31, but evaluated on pre-activations instead – determines a certain component of the Jacobian of the input-output map. Consider a $U(1)$-equivariant network with L layers. We will study the Jacobian of the output z_i^L with respect to the input z_i^0, i.e. the components of the matrices:

$$\frac{\partial z_i^L}{\partial z_j^0} \qquad \frac{\partial z_i^L}{\partial z_j^{0\dagger}} \tag{52}$$

and their complex conjugates.

Due to the equivariance, a $U(1)$ rotation on the input $\delta z_i^0 = i\epsilon z_i^0$ results in the corresponding transformation of the *output* $\delta z_i^L = i\epsilon z_i^L$, leading to the following constraint:

$$i\epsilon z_i^L = \sum_j \left(\frac{\partial z_i^L}{\partial z_j^0} i\epsilon z_j^0 - \frac{\partial z_i^L}{\partial z_j^{0\dagger}} i\epsilon z_j^{0\dagger} \right) \equiv i\epsilon J_i \tag{53}$$

where we have defined a particular component of the Jacobian – correlated in a specific fashion with the input z_i^0 – to be

$$J_i \equiv \sum_j \left(\frac{\partial z_i^L}{\partial z_j^0} z_j^0 - \frac{\partial z_i^L}{\partial z_j^{0\dagger}} z_j^{0\dagger} \right) \tag{54}$$

Note that this is an algebraic constraint relating derivatives to the output of the network: one does not expect such a relation unless equivariance is involved. We now multiply both sides of equation 53 with their complex conjugate and take the expectation value to find

$$d^L = \frac{1}{N} \sum_i \mathbb{E}(|J_i|^2) \tag{55}$$

This is a non-trivial relation. To understand why, consider dynamics at initialization. For a conventional neural network in the chaotic (ordered) phase, one expects the Jacobian to grow (decay) exponentially with layer L. However, as discussed in the main text, in the presence of equivariance the chaotic phase can also be interpreted as a phase with symmetry breaking, and in that case we see that a *particular component* of the Jacobian is controlled by the order parameter d^L, which is $O(1)$ at large L. Thus we see that everywhere in the SSB phase the combination of derivatives constituting J_i is $O(1)$, as we verify in Figure 9. The implications of this for training are discussed around Figure 5 in the main text.

We note that our argument – showing how a symmetry transformation must propagate across distance in the SSB phase, resulting in a long-range correlation – is very similar to that used to derive the physics version of Goldstone's theorem from the path integral formalism in Hofman & Iqbal (2018).

C.4 Comparison to symmetry breaking in physics

We note that in this work though we have borrowed the terminology from equilibrium statistical mechanics in physics, the dynamics are slightly different. In the latter, there is no scope for considering the input/output map, which is our primary motivation. Conversely, in the latter, one usually has some notion of translational invariance in space that relates different degrees of freedom to each other, and order parameters generally measure how aligned these degrees of freedom are, with sharp transitions happening in the thermodynamic limit where we consider infinitely large systems. Thus spontaneous symmetry breaking is generally considered to be equivalent to the idea of *long-range order* (see e.g. Chaikin & Lubensky (1995)). In our current framework we do not generally have such an invariance: for an MLP processing a given data distribution, different neurons are not really related by a symmetry[4], and we believe it is not really meaningful to ask whether distinct neurons are "aligned". Thus our order parameter is different from that in equilibrium statistical mechanics.

D Experimental details

In the following two subsections, we provide some further experimental details for the MLP investigations that are used to support the theory of Section 2, and the RNN models and tasks presented in Section 3.

[4]This is interestingly different in e.g. the case of a convolutional or transformer architecture, where indeed different neurons *are* related by translational or permutation symmetry respectively.

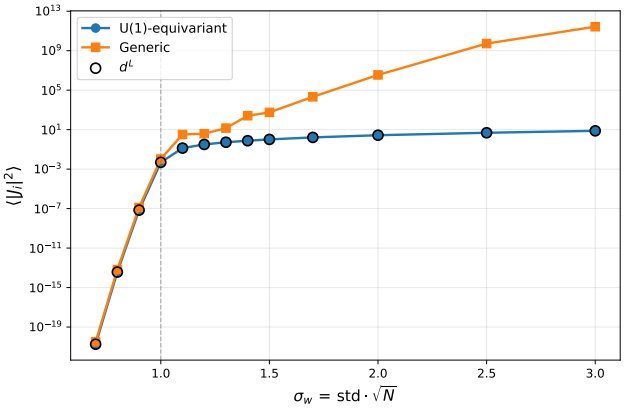


Figure 9: Protected component of Jacobian at initialization. In the symmetry unbroken phase, all components of the Jacobian decay exponentially with layer. In the symmetry broken-phase, one of the components of the equivariant Jacobian can be related to d^L (plotted as circles above) and thus remains $O(1)$ for all σ_W, whereas the corresponding quantity for the non-equivariant model grows exponentially with L. This can be interpreted as the existence of a protected channel for information propagation.

D.1 MLP EXPERIMENTAL IMPLEMENTATION DETAILS

We implement the models in PyTorch Paszke et al. (2019) and used a learning rate of 10^{-3} with Adam Kingma & Ba (2017) for optimization with a batch size of 256. As the idea is to demonstrate the effects of equivariance rather than to optimize over performance, we did not sweep over hyperparameters except to add and remove equivariant architectures, as documented in the text.

The experiments were very lightweight on toy datasets MNIST and Fashion-MNIST Xiao et al. (2017) with the standard splits, and could generally be accomplished on CPU or in less than 1 hour on an RTX 3090.

In Figures 3, 11, and 10 we used $N = 64$ features and trained for 5 epochs with 5 seeds. The order parameter is computed at initialization and averaged over 100 instantiations.

In Figure 4 we used $Nk = 128$ and initialized with a fixed variance for the initial weights of 0.20, which is always deep in the SSB phase for all architectures. We trained for 4 epochs and show results averaged over 3 seeds.

In Figure 5 we plot an effective rank by computing $\exp(H)$, where H is an entropy computed from the singular values σ_i of the representations in (batch, feature) space as in Daneshmand et al. (2020)

$$p_i \equiv \frac{\sigma_i}{\sum_j \sigma_j} \qquad H \equiv -\sum_{i:,\sigma_i>0} p_i \log p_i, \tag{56}$$

In the left panel of Figure 5 we use $N = 64$ and $L = 64$, fixing $\sigma_w = 1.5$ to be deep in the SSB phase and training over one epoch. In the right panel, which is over a single seed, we set $N = 128$ and $L = 90$ to obtain a wider range of layers.

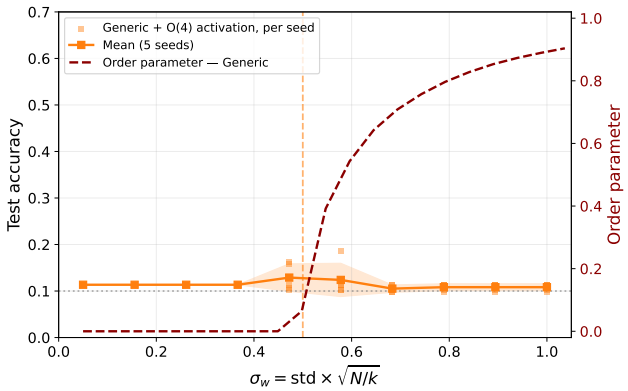


Figure 10: Here we demonstrate the "conventional" edge of chaos for a model with the same nonlinearity as in the right panel of Figure 3 but with generic linear layers that do not preserve the $O(4)$ activation. This is zoomed in on the region of the transition, which is at the left edge of Figure 3. Note that the order parameter is still activated above the transition, but it is *not* accompanied by any performance gain, which only happens modestly near the transition itself, in agreement with classical signal propagation theory. This highlights that it is the equivariance (which allows nontrivial modulations of the order parameter by the symmetry group) that is crucial for the training advantage in the SSB phase.

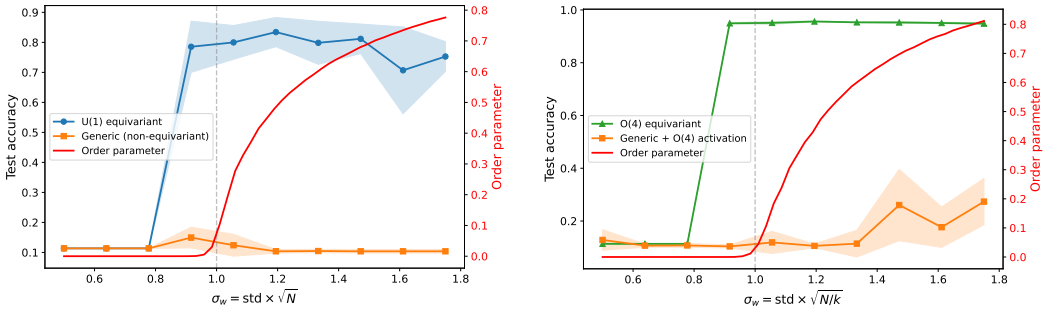


Figure 11: Test accuracy after 5 epochs on MNIST (as compared to Fashion-MNIST in Figure 3) for $U(1)$ (left) and $O(4)$ (right) equivariant models. We compare against models with the same nonlinearity but generic linear layers, thus isolating the effect of the equivariance. We vary σ_w to probe the symmetry breaking phase transition. The training of the equivariant models appears to begin precisely when we enter the SSB phase, as measured by the order parameter $c^\star$, which we compute at *initialization* from equation 8. Training of the non-equivariant models is poor. These results are essentially the same as in Figure 3 except that the numbers are slightly higher as the task is easier.

D.2 RNN EXPERIMENTAL IMPLEMENTATION DETAILS

Below we provide the full details for the copy task, sequential image classification experiments, and UEA dataset experiments.

D.2.1 MODELS

We use N to denote the number of (potentially) symmetry-carrying hidden units. $U(1)$ models have $2N$ real hidden coordinates and $O(k)$ models have Nk. The inputs come as a sequence of real vectors $\mathbf{x}^t \in \mathbb{R}^{d_{\text{in}}}$ for $t \in [0, L]$, and all recurrent models are initialized with $\mathbf{h}^0 = \mathbf{0}$. For the copy task, a linear readout is applied to the hidden state at every timestep to predict the target sequence; for permuted sequential MNIST and UEA classification, the readout is applied only to the final hidden state to predict the class.

Baseline RNNs. The non-equivariant simple RNN baseline is a standard one-layer Elman RNN,

$$\mathbf{h}^t = \psi\big(\mathbf{W}_x\mathbf{x}^t + \mathbf{W}_h\mathbf{h}^{t-1} + \mathbf{b}\big), \qquad \hat{\mathbf{y}}^t = \mathbf{R}\mathbf{h}^t + \mathbf{c}, \tag{57}$$

where $\psi \in \{\tanh, \mathrm{ReLU}\}$ is selected by validation and $\mathbf{R}$ is the linear readout. For the copy and psMNIST tasks, the output $\hat{\mathbf{y}}^t$ is in $\mathbb{R}^{10}$, treated as a vector of logits, and passed through a softmax to predict the current target one-hot vector.

Baseline GRUs. The GRU baseline is a standard one-layer GRU,

$$\begin{aligned}
\mathbf{r}^t &= \sigma\big(\mathbf{W}_{xr}\mathbf{x}^t + \mathbf{W}_{hr}\mathbf{h}^{t-1} + \mathbf{b}_r\big), \\
\mathbf{z}^t &= \sigma\big(\mathbf{W}_{xz}\mathbf{x}^t + \mathbf{W}_{hz}\mathbf{h}^{t-1} + \mathbf{b}_z\big), \\
\tilde{\mathbf{h}}^t &= \tanh\big(\mathbf{W}_{xn}\mathbf{x}^t + \mathbf{b}_{xn} + \mathbf{r}^t \odot \big(\mathbf{W}_{hn}\mathbf{h}^{t-1} + \mathbf{b}_{hn}\big)\big), \\
\mathbf{h}^t &= \mathbf{z}^t \odot \mathbf{h}^{t-1} + (1 - \mathbf{z}^t) \odot \tilde{\mathbf{h}}^t.
\end{aligned} \tag{58}$$

Where the baseline readout is again a learned real affine map from the hidden state to the task output with the same non-linearities as for the copy task.

Equivariant input embeddings. For the equivariant models, the real input is first embedded into the hidden representation on which the symmetry acts. For $U(1)$ models, this is implemented as a learned complex embedding

$$\mathbf{u}^t = \mathbf{E}_{\text{Re}}\mathbf{x}^t + i\,\mathbf{E}_{\text{Im}}\mathbf{x}^t \in \mathbb{C}^N, \tag{59}$$

where $\mathbf{E}_{\text{Re}}, \mathbf{E}_{\text{Im}} \in \mathbb{R}^{N \times d_{in}}$. For $O(k)$ models, the embedding matrix $\mathbf{E}_{\text{O(k)}} \in \mathbb{R}^{Nk \times d_{in}}$ has k times greater output dimensions, and the output is reshaped into N real k-vectors,

$$\mathbf{u}^t_\alpha \in \mathbb{R}^k, \qquad \alpha = 1, \ldots, N. \tag{60}$$

$U(1)$-equivariant RNN. The $U(1)$ hidden state is complex, $\mathbf{h}^t \in \mathbb{C}^N$, designed to be equivariant to the global action $\mathbf{h}^t \mapsto e^{i\theta}\mathbf{h}^t$. The recurrent map is complex-linear and the nonlinearity is radial:

$$\mathbf{h}^t = \phi_{U(1)}\big(\mathbf{u}^t + \mathbf{W}\mathbf{h}^{t-1}\big), \qquad \mathbf{W} \in \mathbb{C}^{N \times N}, \tag{61}$$

with

$$\big[\phi_{U(1)}(\mathbf{z})\big]_\alpha = \frac{\tanh(|z_\alpha|)}{|z_\alpha| + \epsilon} z_\alpha. \tag{62}$$

Thus the activation changes the magnitude of each complex hidden coordinate while preserving its phase. In practice we use $\epsilon = 10^{-8}$. The reported $U(1)$ RNN and GRU runs use a real readout

$$\hat{\mathbf{y}}^t = \mathbf{R}\begin{bmatrix}\mathrm{Re}[\mathbf{h}^t] \\ \mathrm{Im}[\mathbf{h}^t]\end{bmatrix} + \mathbf{c}, \tag{63}$$

which maps the complex hidden representation back to the real task output space.

$O(k)$-equivariant RNN. For $O(k)$ models, the hidden state is $\mathbf{h}^t \in \mathbb{R}^{N\times k}$, where each $\mathbf{h}^t_\alpha \in \mathbb{R}^k$ is designed to be equivariant to the action

$$\mathbf{h}^t_\alpha \mapsto O\mathbf{h}^t_\alpha, \qquad O \in O(k), \tag{64}$$

with the same orthogonal matrix acting on every hidden block indexed by α. The recurrent update is

$$\mathbf{q}^t_{\alpha a} = \mathbf{u}^t_{\alpha a} + \sum_{\beta=1}^{N} \mathbf{A}_{\alpha\beta}\mathbf{h}^{t-1}_{\beta a}, \qquad \mathbf{h}^t_\alpha = \phi_{O(k)}(\mathbf{q}^t_\alpha), \tag{65}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$. In flattened coordinates the recurrent matrix is $\mathbf{A} \otimes I_k$, so it mixes the N hidden blocks but acts identically on each of the k internal coordinates. The nonlinearity acts only through the $O(k)$-invariant radius:

$$\phi_{O(k)}(\mathbf{v}_\alpha) = \frac{\tanh(\|\mathbf{v}_\alpha\|_2)}{\|\mathbf{v}_\alpha\|_2 + \epsilon}\mathbf{v}_\alpha. \tag{66}$$

Equivalently, this replaces the length of each k-vector by the scalar nonlinearity of its length, while preserving its direction in $\mathbb{R}^k$. The implementation allows either an invariant norm readout,

$$\hat{\mathbf{y}}^t = \mathbf{R}\mathbf{s}^t + \mathbf{c}, \qquad s^t_\alpha = \|\mathbf{h}^t_\alpha\|_2, \tag{67}$$

or a flattened real readout, $\hat{\mathbf{y}}^t = \mathbf{R}\,\mathrm{vec}(\mathbf{h}^t) + \mathbf{c}$. In practice, we find the flattened readout to perform best and thus use it in all reported experiments.

$U(1)$-equivariant GRU. The $U(1)$ GRU uses the same complex hidden state and radial candidate update as the $U(1)$ RNN, but adds real-valued reset and update gates. The gates are computed from the real input and from phase-invariant hidden features:

$$\begin{aligned}
\mathbf{r}^t &= \sigma\big(\mathbf{W}_{xr}\mathbf{x}^t + \mathbf{W}_{hr}|\mathbf{h}^{t-1}| + \mathbf{b}_r\big)\,, \\
\mathbf{z}^t &= \sigma\big(\mathbf{W}_{xz}\mathbf{x}^t + \mathbf{W}_{hz}|\mathbf{h}^{t-1}| + \mathbf{b}_z\big)\,, \\
\tilde{\mathbf{h}}^t &= \phi_{U(1)}\big(\mathbf{u}^t + \mathbf{W}_h\left(\mathbf{r}^t \odot \mathbf{h}^{t-1}\right)\big)\,, \\
\mathbf{h}^t &= \mathbf{z}^t \odot \mathbf{h}^{t-1} + (1 - \mathbf{z}^t) \odot \tilde{\mathbf{h}}^t.
\end{aligned} \tag{68}$$

Here $|\mathbf{h}|$ is the complex magnitude, taken elementwise, σ denotes a sigmoid nonlinearity, $\odot$ denotes the elementwise product, and $\mathbf{r}^t, \mathbf{z}^t \in [0,1]^N$ are real. Since the gates depend on the hidden state only through phase-invariant magnitudes, multiplying $\mathbf{h}^{t-1}$ by a global phase leaves the gates unchanged, while the candidate branch transforms equivariantly. We use the default gate initialization $\mathbf{b}_r = \mathbf{0}$ and $\mathbf{b}_z = 2$, so the initialized update gate is biased toward copying the previous hidden state.

Parameter counts. All parameter counts in Tables 2, 3 & 4 and Figures 6 & 7 report trainable real scalar parameters. Thus each trainable complex scalar counts as two real parameters.

UEA model settings. For the UEA experiments, we fix the baseline RNN and GRU candidate nonlinearities to $\tanh$, and use radial $\tanh$ for both equivariant models, with $k = 4$ for the $O(k)$ RNN. We use $N = 16$ complex hidden units for the $U(1)$ GRU and approximately match its parameter count for the other models. This gives approximately $2{,}000$ parameters for a typical dataset ($8{,}000$ on average), with average widths of 20 and 38 real hidden units for the baseline GRU and RNN, respectively, and 21 four-dimensional hidden units for the $O(4)$ RNN.

D.2.2 Initialization and hyperparameter sweeps.

For the copy and psMNIST experiments, we use the following initialization and hyperparameter sweeps. For all models we initialize the hidden state to zero at the start of each sequence: $\mathbf{h}^0 = \mathbf{0}$. We sweep over identity and random recurrent initializations. Identity initialization corresponds to $\mathbf{W} = I_N \in \mathbb{C}^{N\times N}$ for the $U(1)$ RNN, $\mathbf{A} = I_N$ for the $O(k)$ RNN, and identity recurrent blocks for the baseline recurrent models. For the $U(1)$ GRU, this choice applies only to $\mathbf{W}_h$; the recurrent gate matrices $\mathbf{W}_{hr}$ and $\mathbf{W}_{hz}$ always initialize to zero. Random recurrent initialization uses the PyTorch defaults for the baseline and $U(1)$ models; for the $O(k)$ RNN, entries of $\mathbf{A}$ are sampled from $\mathcal{N}(0, 1/N)$. For the copy task, we consider standard and low-variance input initializations.

For psMNIST, the baseline RNN uses standard input initialization and the $O(k)$ RNN uses low-variance input initialization, while the GRUs support both choices. Standard input initialization uses the PyTorch defaults: uniform bounds $\pm 1/\sqrt{d_{\text{in}}}$ for linear input embeddings and $\pm 1/\sqrt{N}$ for the baseline GRU. Low-variance initialization samples real input weights from a zero-mean Gaussian with standard deviation 0.001, including the separate real and imaginary input coordinates of the $U(1)$ models. The $U(1)$ GRU gate input weights always use this Gaussian initialization, independently of the candidate input initialization. For the copy task, the blank-token input columns are clamped to zero before every recurrent update in the baseline RNN, $U(1)$ RNN, and $U(1)$ GRU, independently of the input initialization. The baseline GRU instead initializes its first input column to zero only under low-variance initialization and subsequently allows it to train; this also applies to its psMNIST implementation. For the $O(k)$ RNN on psMNIST, we performed a random search over values of neurons $N \in \{64, 128, 256, 512\}$, and $k \in \{6, 8, 12, 18, 24, 48\}$ up to our compute budget, and report the best performing combination at each parameter range. For the reported results we additionally sweep over learning rates from the set $\{10^{-3}, 10^{-4}, 10^{-5}\}$.

UEA hyperparameter sweep. For each UEA dataset and model, we sweep over learning rates $\{10^{-5}, 10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$ and two input initializations. We sample each real input weight from a zero-mean Gaussian with standard deviation either 0.001 or $1/\sqrt{d}$, where d is the number of measurement channels. This applies to both the input embedding and the gate input matrices. We initialize the candidate recurrent maps to the identity and the recurrent gate matrices to zero. All biases start at zero except for the update gate bias (the input bias for the baseline GRU), which is set to $+2$. Readout weights have Gaussian standard deviation $1/\sqrt{H}$, where H is the number of real hidden coordinates. The added time coordinate described below (in the datasets section) is excluded from d, and its input weights start at zero but remain trainable. We train each of these ten hyperparameter combinations with two random seeds.

Best performing hyperparameters. For the copy and psMNIST experiments, we report the hyperparameters found to be optimal for each architecture, dataset, and hidden state size in Table 2 below.

D.2.3 Extended Results.

In Tables 3 and 4, we provide the exact values for the data presented in Figures 6 and 7. In Figure 6, the significance stars are calculated via a pairwise Welch's two-sample t-test between each pair of models.

D.2.4 Datasets

Variable-Delay Copy Task. We use a variable-delay version of the standard copy task Hochreiter & Schmidhuber (1997); Graves et al. (2014); Henaff et al. (2016); Gu et al. (2020). Each example begins with a length-10 sequence of randomly sampled one-hot data tokens which the model must store in memory. The model is then presented with a sequence of blank delay tokens, followed by a special go token which indicates that the stored sequence should be reproduced. Each token is a 10-dimensional one-hot vector, where the delay token is index zero, the memory sequence is randomly sampled from indices 1 to 8, and the 'go' token is fixed to index 9. Here T counts the timesteps before recall, including the 10 data tokens and the 'go' token. We sample $T \sim \mathcal{U}\{11, \ldots, T_{\max}\}$ independently for each example, giving $T-11$ blank delay tokens. The target is set to the blank token at every timestep except for the 10 timesteps following the go token, where the network must output the original sequence in the same order. When batching examples with different delays, sequences are padded with blank tokens to $T_{\max} + 10$ timesteps, and the padded positions are assigned blank targets. Given the limited nature of the synthetic dataset, no fixed training, validation, or test sets are created. Instead, fresh batches are randomly sampled for each training, validation, and test iteration. The test batch size is set to 50, and evaluated every 1000 training iterations.

The variable delay modification prevents the model from relying on a fixed output time, and instead requires it to use the go token to determine when the memory should be read out. We found that without this modification, simple linear RNNs outperformed all non-linear models, making the task less relevant to more real-world problems where non-linearity is almost always required. In the experiments of Figure 6, we use $T_{\max} = 25$ for the simple RNN experiments and $T_{\max} = 100$ for

Table 2: Best-performing hyperparameters selected by validation performance for each dataset, architecture, and hidden size. We sweep recurrent initialization, input/encoder initialization, activation function, and learning rate (and k when appropriate), and report the setting with the best mean validation performance across sweep seeds.

Dataset	Model	N	Rec. init.	Input init.	Activation	LR	k	# Params
Copy	RNN	32	identity	standard	$\tanh$	10^{-4}	-	1.7K
Copy	RNN	64	identity	standard	$\tanh$	10^{-4}	-	5.5K
Copy	GRU	32	identity	standard	$\tanh$	10^{-3}	-	4.5K
Copy	GRU	64	identity	low-var.	$\tanh$	10^{-3}	-	15K
Copy	$U(1)$ RNN	32	identity	low-var.	radial $\tanh$	10^{-3}	-	3.3K
Copy	$U(1)$ GRU	32	random	low-var.	radial $\tanh$	10^{-3}	-	6.1K
psMNIST	RNN	128	identity	standard	ReLU	10^{-4}	-	18K
psMNIST	RNN	256	identity	standard	ReLU	10^{-5}	-	69K
psMNIST	RNN	512	identity	standard	ReLU	10^{-4}	-	269K
psMNIST	RNN	1024	identity	standard	ReLU	10^{-4}	-	1.06M
psMNIST	GRU	64	identity	standard	$\tanh$	10^{-3}	-	14K
psMNIST	GRU	128	identity	standard	$\tanh$	10^{-3}	-	52K
psMNIST	GRU	256	identity	standard	$\tanh$	10^{-3}	-	201K
psMNIST	GRU	512	identity	standard	$\tanh$	10^{-3}	-	796K
psMNIST	$O(k)$ RNN	64	identity	low-var.	radial $\tanh$	10^{-4}	18	18k
psMNIST	$O(k)$ RNN	128	identity	low-var.	radial $\tanh$	10^{-4}	8	29k
psMNIST	$O(k)$ RNN	256	identity	low-var.	radial $\tanh$	10^{-4}	6	84k
psMNIST	$O(k)$ RNN	256	identity	low-var.	radial $\tanh$	10^{-4}	24	139k
psMNIST	$O(k)$ RNN	256	identity	low-var.	radial $\tanh$	10^{-4}	48	213k
psMNIST	$O(k)$ RNN	512	identity	low-var.	radial $\tanh$	10^{-4}	24	410k
psMNIST	$U(1)$ GRU	64	identity	low-var.	radial $\tanh$	10^{-3}	-	18K
psMNIST	$U(1)$ GRU	128	identity	low-var.	radial $\tanh$	10^{-3}	-	69K
psMNIST	$U(1)$ GRU	256	identity	low-var.	radial $\tanh$	10^{-3}	-	269K
psMNIST	$U(1)$ GRU	512	identity	low-var.	radial $\tanh$	10^{-3}	-	1.06M

Table 3: Summary of the variable-delay copy task results presented in Figure 6. We report test cross-entropy loss, averaged over sequence length and examples, then divided by the test batch size of 50. Values are mean $\pm$ standard deviation over 11 random seeds.

T_{max}	Model	N	Test CE loss $\pm$ Stdev.	# Params
25	RNN	32	8.40e-3 $\pm$ 4.39e-4	1.7K
25	RNN	64	2.08e-3 $\pm$ 3.84e-4	5.5K
25	$U(1)$ RNN	32	**7.26e-4 $\pm$ 5.47e-4**	3.3K
100	GRU	32	2.11e-3 $\pm$ 4.02e-4	4.5K
100	GRU	64	3.17e-3 $\pm$ 4.03e-3	15K
100	$U(1)$ GRU	32	**3.14e-4 $\pm$ 4.41e-4**	6.1K

the GRU experiments; we additionally consider $T_{\max} = 200$ with the same hyperparameter sweep and find that while nearly all of the models struggle, the $U(1)$ GRU model is the only one to have a single seed which successfully 'solves' the task (achieving a test cross-entropy loss of less than 10^{-4}) and achieves a loss of 3.9×10^{-7}.

Permuted Sequential MNIST. For the sequential image classification experiments, we use the standard permuted sequential MNIST benchmark (Le et al., 2015; Arjovsky et al., 2016). Each 28×28 MNIST image is flattened into a sequence of 784 scalar pixel values and presented to the

Table 4: Summary of the psMNIST results presented in Figure 7. We report test set accuracy. Values are mean ± standard deviation over 3 random seeds.

Model	N	k	Test Acc. ± Stdev.	# Params
RNN	128	-	85.19 ± 3.13	18K
RNN	256	-	88.90 ± 0.73	69K
RNN	512	-	90.92 ± 1.02	269K
RNN	1024	-	91.00 ± 2.57	1.06M
$O(k)$ RNN	64	18	87.09 ± 0.55	18K
$O(k)$ RNN	128	8	88.98 ± 0.92	29K
$O(k)$ RNN	256	6	92.01 ± 1.43	84K
$O(k)$ RNN	256	24	93.34 ± 1.86	139K
$O(k)$ RNN	256	48	94.16 ± 0.23	213K
$O(k)$ RNN	512	24	94.79 ± 0.22	410K
GRU	64	-	91.35 ± 1.88	14K
GRU	128	-	92.77 ± 0.99	52K
GRU	256	-	94.41 ± 0.63	201K
GRU	512	-	95.08 ± 0.35	796K
$U(1)$ GRU	64	-	93.51 ± 0.86	18K
$U(1)$ GRU	128	-	94.92 ± 0.39	69K
$U(1)$ GRU	256	-	95.88 ± 0.22	269K
$U(1)$ GRU	512	-	96.21 ± 0.14	1.06M

model one pixel at a time. The model receives no additional spatial information, and the digit class is predicted only from the final hidden state after the full sequence has been processed.

To make the task more difficult than standard sequential MNIST, we sample a single random permutation of the 784 pixel locations before training and apply this same fixed permutation to every image in the training, validation, and test sets. This removes the short-range spatial correlations present in the usual raster ordering of the pixels, forcing the model to integrate information over the full sequence length rather than exploiting local image structure to reduce the effective task length. We use the standard MNIST train/test split, with model selection performed on held-out validation performance and final numbers reported on the test set. The validation set is built from 3,000 random samples from the original 60,000 sample train set.

UEA Datasets. We use the standard train/test splits for the 30 UEA datasets (Bagnall et al., 2018), holding out approximately 20% of the training examples from each class for validation. We construct a separate validation split for each of the two sweep seeds and use the same split across all models and hyperparameters. Identical records are kept together, with at least one group retained for training in each class. We standardize each measurement channel using its mean and population standard deviation over valid timesteps in the training subset, excluding padding. We also add a time coordinate $(t-1)/L_{\text{train}}$, where $t = 1, \ldots, L$ and L_{train} is the maximum length in that training subset. Variable-length sequences are padded for batching, with the hidden state held fixed after the last valid timestep. We use the variable-length InsectWingbeat data with 2–22 timesteps, which differs from the length-78 version in the original archive paper.

D.2.5 Training and Evaluation

We use the Adam optimizer Kingma & Ba (2017) for all experiments with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ (PyTorch defaults).

Copy task. All copy task models were trained for 1 million iterations with a batch size of 128. The models are trained and evaluated using cross-entropy loss at each timestep, averaged over the full sequence length and over the batch, then divided by the batch size (128 during training and 50 during evaluation). Thus, the copy-task losses reported in Figure 6 and Table 3 correspond to this scaled test cross-entropy, averaged over sequence length, test batches and random seeds. We

run 3 random seeds for each model from the sweep, totaling to 72 runs for each model and hidden state size (2 recurrent initializations, 2 input initializations, 2 activation functions, 3 learning rates, 3 seeds). For the copy task we identified the best performing hyperparameters from this initial sweep (reported in Table 2) based on mean validation performance and ran an additional 7 random seeds for the best performing models to get the full distributions presented in Figure 6. In full, for the copy task, this resulted in 478 trained models.

Permuted Sequential MNIST. All psMNIST models are trained for 120 epochs with a batch size of 128. The models are trained with a cross-entropy loss over the 10 MNIST digit class labels, and the loss is averaged over the batch. We use gradient clipping and a learning rate decay schedule found to be optimal from prior work Keller et al. (2023) using similar RNNs on this task. Specifically, we used global gradient norm clipping at 1.0, and a step-based learning rate schedule where the learning rate decreases by a factor of 10 after 100 epochs.

UEA Datasets. We train all models with cross-entropy loss on the final sequence label, using a batch size of four, a constant learning rate, and global gradient norm clipping at 1.0. We use full backpropagation through each sequence, without truncation or subsampling. Each sweep run is trained for at most 100,000 updates, with validation every 1,000 updates and early stopping after 11 consecutive checks without an improvement in validation accuracy. For each run, we select the checkpoint with the highest validation accuracy, breaking ties by lower validation cross-entropy. For each dataset and model, we then select the hyperparameters with the highest mean validation accuracy at these checkpoints over the two sweep seeds.

We train the selected configuration from scratch with five new random seeds on the full training set, including the held-out validation examples, and recompute the normalization statistics on this full set. We set the training duration to the median number of epochs at the two selected sweep checkpoints, converted to updates on the full training set. We then evaluate the last checkpoint on the test set, without further model selection. We report mean accuracy and sample standard deviation over the five runs. The paired t-tests in Table 5 use matching seeds, which share the training-example order but can differ in initialization and selected training duration. We use two-sided tests without multiple-comparison correction. The average accuracy in Table 1 gives equal weight to each dataset.

Computational requirements. We ran the copy and psMNIST experiments on a combination of H100 and A100 NVIDIA GPUs. Each model presented here comfortably fits within a single A100 GPU's memory (40GB), with all but the largest $O(k)$ RNNs using less than a few GB. For the copy task, with $T_{max} = 25$, the average runtimes for the RNN and U(1) RNNs were similar at around 8 hours for 1M iterations. For the GRU models, the runtime comparison was confounded by a non-optimized implementation of the $U(1)$ GRU causing it to run significantly slower than the non-equivariant baseline ($\approx$ 2 days per $U(1)$ model vs. $\approx$ 4 hours per generic GRU). In total we estimate around 4,000 GPU hours were used to produce the copy task results. For psMNIST, the average runtime was around 2.5 hours for the RNN models, 7 hours for the $O(k)$ RNN models, 14 hours for the $U(1)$ GRU models, and around 1 hour for the generic GRU models. Again, these runtimes are with highly unoptimized PyTorch implementations in a relatively uncontrolled setting (shared machines) where performance may vary. In theory, we see no reason why an optimized implementation of the equivariant models should not run with the same computational complexity as an equivalent non-equivariant model with the same hidden state dimensionality. In total, we estimate around 4,500 GPU hours were used to produce the psMNIST results. For the UEA experiments, we used NVIDIA RTX A6000 GPUs, and an accelerated Jax implementation where outputs were matched numerically to the PyTorch implementation. With this accelerated implementation, UEA training times ranged from minutes to tens of minutes, reaching several hours for EigenWorms. Including the full hyperparameter sweeps and all random seeds, we estimate around 200 GPU hours in total on this dataset, with over 90% spent on tuning.

E Full UEA Time-Series Classification Results

Table 5: Approximately matched parameter count. Test accuracy (%), mean ± std over five seeds; p values are two-sided paired t-tests, with the higher number of each pair bolded only when $p < 0.05$. Total number of significant wins by each model is compiled below.

Dataset	U(1)-GRU	GRU	p	Tanh RNN	O(4)-RNN	p
EigenWorms	$\mathbf{60.15 \pm 7.21}$	48.85 ± 3.10	0.040	42.14 ± 1.13	$\mathbf{53.28 \pm 5.08}$	0.006
MotorImagery	51.00 ± 2.55	52.00 ± 3.24	0.706	44.60 ± 3.85	48.20 ± 1.64	0.171
StandWalkJump	18.67 ± 5.58	18.67 ± 8.69	1.000	37.33 ± 14.61	26.67 ± 10.54	0.256
EthanolConcentration	30.11 ± 3.15	32.78 ± 1.12	0.199	28.97 ± 2.76	31.56 ± 2.48	0.260
Cricket	94.17 ± 2.67	87.50 ± 2.95	0.054	82.50 ± 4.97	$\mathbf{93.61 \pm 4.00}$	0.030
SelfRegulationSCP2	49.56 ± 3.98	52.33 ± 2.27	0.329	$\mathbf{53.22 \pm 1.44}$	48.89 ± 1.42	0.001
SelfRegulationSCP1	81.02 ± 2.08	83.75 ± 4.49	0.258	85.53 ± 5.23	81.84 ± 1.04	0.232
AtrialFibrillation	30.67 ± 5.96	29.33 ± 5.96	0.749	34.67 ± 9.89	34.67 ± 7.30	1.000
Heartbeat	70.63 ± 1.90	72.29 ± 2.14	0.150	70.73 ± 2.18	69.56 ± 3.28	0.645
HandMovementDirection	$\mathbf{52.70 \pm 2.14}$	39.19 ± 8.81	0.042	50.81 ± 3.77	$\mathbf{59.19 \pm 7.55}$	0.029
UWaveGestureLibrary	$\mathbf{82.69 \pm 1.51}$	75.94 ± 1.65	0.005	67.06 ± 2.64	$\mathbf{82.12 \pm 2.45}$	0.001
DuckDuckGeese	64.00 ± 1.41	63.20 ± 1.79	0.542	58.00 ± 1.41	59.20 ± 6.10	0.716
PhonemeSpectra	19.45 ± 0.77	19.64 ± 0.84	0.790	9.36 ± 0.56	$\mathbf{13.44 \pm 1.07}$	0.002
Epilepsy	91.74 ± 1.50	92.61 ± 1.19	0.071	71.74 ± 4.72	$\mathbf{88.84 \pm 2.33}$	0.002
CharacterTrajectories	98.72 ± 0.23	98.02 ± 1.14	0.267	67.10 ± 13.21	$\mathbf{97.97 \pm 0.36}$	0.006
Handwriting	33.06 ± 2.70	29.69 ± 2.82	0.163	11.22 ± 1.36	$\mathbf{22.45 \pm 7.08}$	0.029
ArticularyWordRecognition	91.93 ± 3.97	92.73 ± 1.12	0.650	86.33 ± 1.62	$\mathbf{90.47 \pm 1.30}$	0.010
PEMS-SF	76.07 ± 1.94	76.42 ± 1.89	0.756	74.57 ± 2.52	70.64 ± 2.14	0.069
BasicMotions	91.00 ± 7.20	87.50 ± 6.61	0.581	85.00 ± 12.87	99.00 ± 1.37	0.060
SpokenArabicDigits	$\mathbf{97.54 \pm 0.60}$	95.76 ± 0.76	0.028	93.48 ± 0.65	92.81 ± 0.65	0.066
ERing	$\mathbf{78.96 \pm 4.31}$	67.41 ± 6.64	0.026	79.41 ± 2.90	$\mathbf{86.89 \pm 1.79}$	0.015
FaceDetection	61.81 ± 1.21	62.73 ± 0.85	0.344	62.47 ± 0.47	62.10 ± 0.49	0.402
NATOPS	91.67 ± 0.88	92.67 ± 1.59	0.322	86.33 ± 2.77	89.78 ± 1.74	0.093
FingerMovements	52.60 ± 4.10	55.40 ± 3.05	0.284	58.40 ± 2.61	52.00 ± 3.94	0.076
Libras	80.22 ± 4.07	77.22 ± 3.02	0.318	60.56 ± 2.19	$\mathbf{71.67 \pm 3.14}$	0.004
LSST	58.06 ± 2.35	58.22 ± 1.17	0.865	48.86 ± 1.12	$\mathbf{52.47 \pm 1.91}$	0.033
RacketSports	$\mathbf{79.74 \pm 2.25}$	74.87 ± 1.64	0.009	75.79 ± 1.27	$\mathbf{82.11 \pm 1.77}$	0.003
JapaneseVowels	96.76 ± 0.90	95.95 ± 0.83	0.158	96.00 ± 0.48	$\mathbf{98.05 \pm 0.59}$	0.008
InsectWingbeat	58.94 ± 0.27	58.77 ± 0.12	0.218	$\mathbf{57.55 \pm 0.29}$	47.63 ± 0.53	< 0.001
PenDigits	97.62 ± 0.27	97.38 ± 0.45	0.466	96.77 ± 0.56	97.47 ± 0.33	0.067
Significant dataset wins	6	0		2	14	

F Topological Defects

We note that in physics spontaneously broken symmetries are associated with two sets of phenomena which control the long-time physics: Goldstone modes (whose deep-learning analogue we have extensively studied here), and *topological defects*, localized objects such as vortices, domain walls and hedgehogs (monopoles) which persist stably in time (Coleman, 1988; Rajaraman, 1982). The latter exist even in the case of spontaneously broken discrete symmetries such as $\mathbb{Z}_2$. They provide a logically distinct route towards stable information propagation through layers of a network. Interestingly, there are some indications that such structures may play a role in neuroscience (Xu et al., 2023). They remain relatively unexplored in a deep learning context (though see Iqbal & Welling (2025) for some preliminary investigations). In this section, Figures 12 and 13, we present a preliminary study of such topological defects in a 2D convolutional RNN (similar to Recurrent CNNs (Pinheiro & Collobert, 2014; Liang & Hu, 2015) and ConvLSTMs (Shi et al., 2015; Ballas et al., 2016; Wang et al., 2017; Keller & Welling, 2023) used for spatiotemporal forecasting) augmented with $U(1)$ symmetry, where we see long-lived vortices emerging when solving the copy task, though their precise role remains to be understood.

In the following we present visualizations of the hidden state dynamics for a 2D convolutional RNN with $U(1)$ symmetry trained on the copy task with a fixed recall-onset parameter $T = 100$, corresponding to 89 blank delay tokens. The model is identical to the $U(1)$-equivariant RNN described in Section D.2 above, except the complex-linear hidden to hidden recurrent connectivity matrix is replaced with a complex convolutional kernel of size 3×3 with 16 channels. The spatial dimension of hidden state vector is of size $N = 400$, and shaped into a 20×20 regular grid, over which the 2D convolution operates. The model has 16 convolutional channels, of which two are plotted below. This model learns to solve the copy task well. We plot the magnitude of the complex hidden state neurons over sequence length in the top row, and we can see that vortices emerge in the hidden state (bottom row), as measured by computing the winding numbers for each point in the phase plot in the middle row. As would be expected, we find that vortices always come in positive and negative pairs (defined by inverse winding directions), and can be observed to annihilate with one another on contact. While we have yet to demonstrate convincingly that these vortices are used to store memory in the model, it is promising that they emerge during the memory storage period of the task, i.e. the first 10 steps, and remain stable over the remaining sequence length. We see in the two plots that different channels learn to have different densities of vortices.

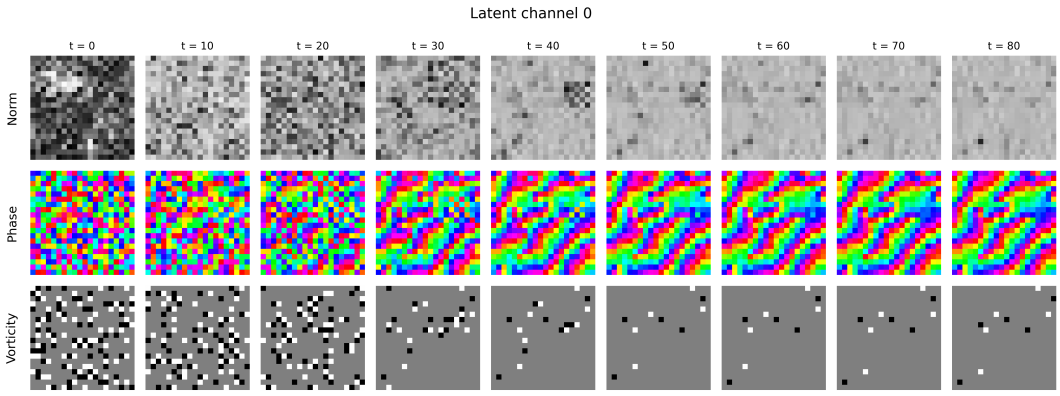


Figure 12: Visualization of hidden state magnitude (top), phase (middle), and vorticity (bottom) for a 2D convolutional U(1) equivariant RNN trained on the copy task with recall-onset parameter $T = 100$ (89 blank delay tokens) and a hidden state size of 400. We show a single channel which appears to exhibit few long lasting vortices which are robustly conserved over the iterations after settling in. Note vortices always come in positive/negative pairs and can annihilate on contact. We find that larger hidden states allow more vortices to persist for longer.

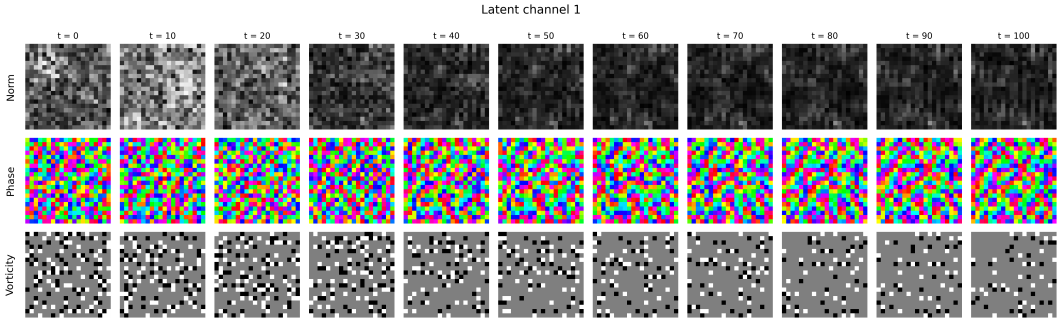


Figure 13: A second channel from the same 2D convolutional U(1) equivariant RNN. In this channel, we observe significantly higher vorticity that persists throughout the sequence length.