A biology-inspired recurrent oscillator network for computations in high-dimensional state space

Felix Effenberger,[1,2*] Pedro Carvalho,[1] Igor Dubinin,[1,2] Wolf Singer[1,2,3*]

[1]Ernst Strüngmann Institute, Frankfurt am Main, Germany
[2]Frankfurt Institute for Advanced Studies, Frankfurt am Main, Germany
[3]Max Planck Institute for Brain Research, Frankfurt am Main, Germany

*To whom correspondence should be addressed;
E-mail: felix.effenberger@esi-frankfurt.de, wolf.singer@esi-frankfurt.de.

Abstract

Biological neuronal networks have the propensity to oscillate. However, it is unclear whether these oscillations are a mere byproduct of neuronal interactions or serve computational purposes. Therefore, we implemented hallmark features of the cerebral cortex in recurrent neuronal networks (RNNs) simulated in silico and examined their performance on common pattern recognition tasks after training with a gradient-based learning rule. We find that by configuring network nodes as damped harmonic oscillators (DHOs), performance is substantially improved over non-oscillating architectures and that the introduction of heterogeneous nodes, conduction delays, and network modularity further improved performance. We furthermore provide a proof of concept of how unsupervised Hebbian learning can work in such networks. Analyses of network activity illustrate how the nonlinear dynamics of coupled DHOs drive performance, and provide plausible a posteriori explanations for a number of physiological phenomena whose function so far has been elusive.

Introduction

Biological neuronal systems such as the cerebral cortex are characterized by several canonical dynamical and anatomical features. Among those are recurrent connections between network nodes within and between processing layers [1, 2], the propensity of nodes and subnetworks to oscillate in different preferred frequency ranges [3, 4], the expression of heterogeneous conduction delays [5, 6], and the activity-dependent adjustment of the gain of connections [7, 8] via Hebbian learning principles.

As neurobiological investigations of the cerebral cortex indicate, these features endow neuronal networks with highly complex, nonlinear dynamics capable of generating high-dimensional spatiotemporal patterns in a self-organized manner [9], see [2] for a review. In simultaneous recordings from multiple network nodes, these patterns manifest themselves as frequency-varying oscillations [10, 11], transient synchronization or desynchronization phenomena [12], resonance [13], entrainment [14], phase shifts [15], and traveling waves [16, 17].

The implementation of principles realized in natural neuronal systems has proven extremely successful for the design of computational architectures, from simple feedforward networks such as the perceptron [18], over deep and convolutional neural networks (DNNs) [19], to the numerous applications of reservoir computing [20] and recurrent networks (RNNs) [19, 21, 22].

Despite the computational advantages offered by the high-dimensional state space resulting from recurrent dynamics, RNNs are still explored to a lesser degree than pure feedforward DNNs because their dynamics are difficult to tame and they are difficult to train using gradient-based methods [23]. As

gated architectures such as LSTM [24] and GRU [25] address some of these issues, these architectures became the status quo for RNNs [19].

Another hallmark characteristic of cortical networks is their tendency to produce oscillating dynamics. Several hypotheses suggest that oscillations can be exploited for computation [26, 27, 2] and the role of oscillating neurons and microcircuits in the generation of motor patterns is well established [28]. However, it remains a matter of discussions whether oscillations support computations in the context of cognitive processes in the cerebral cortex and, if so, how [29, 30, 31, 32]. While a number of attempts have recently been made to take advantage of the complex dynamics of recurrently coupled oscillator networks for computations [33, 34], these approaches have either focused on possible hardware implementations using relatively simple tasks [33] or mere task performance [34], without providing explanations on how the observed gain of function is brought about by means of network dynamics.

In this study, we introduce biologically inspired harmonic oscillator RNNs (HORNs) designed to capture the aforementioned characteristics of the mammalian neocortex (Fig.1A). We train different HORN models on benchmark pattern classification tasks using the backpropagation through time (BPTT) learning algorithm and compare their performance to other state of the art RNN architectures such as LSTM and GRU (Fig. 1C). Benchmark tasks consisted of handwritten digit and letter recognition (MNIST and EMNIST), as well as recognition of spoken digit from audio recordings (see Methods). We find that HORNs demonstrate an outstanding efficiency with respect to learning speed, noise tolerance, and hardware requirements on all data sets compared to other state of the art RNN models (Fig. 1D,G,H). Moreover, we find that the inclusion of additional features found in biological networks, such as heterogeneity, coupling delays, and network modularity, can further increase task performance at the same number of trainable parameters. An analysis of HORN dynamics is able to explain the mechanisms leading to the observed performance increase and surprisingly also provides plausible a posteriori explanations for a number of physiological phenomena whose function has so far been elusive or has given rise to controversial discussions.

Oscillating units

To capture the tendency of cortical networks to engage in oscillatory activity [26], we configured the nodes of RNNs as damped harmonic oscillators (DHOs). The phenomenon of damped harmonic oscillations generically results from excitatory/inhibitory interactions and negative feedback and is ubiquitous in nature, making DHOs the most basic choice for introducing oscillations at the level of a single network node. Importantly, a DHO unit in our networks does not necessarily represent a single biological neuron with oscillatory properties, such as a pacemaker neuron [28], but it can also be understood as representing an oscillating microcircuit composed of recurrently coupled populations of excitatory (E) and inhibitory (I) neurons such as a (P)ING circuit [35]. In this sense, a HORN network can be understood as a mesoscale model of neuronal activity (see Fig. 1A and Methods).

Each DHO unit has one state variable x, the time-varying amplitude of the oscillator, and three parameters: The natural frequency ω, the damping coefficient γ, and the excitability coefficient α. It is defined by the second order ODE $\ddot{x}(t) + 2\gamma\dot{x}(t) + \omega^2 x(t) = \alpha\sigma\left(I\left(t\right) + F\left(t\right)\right)$, where σ is an input nonlinearity that models local resource constraints, $F(t)$ models feed-back input of the unit to itself which captures characteristics of the dynamics of the underlying microcircuit, and $I(t)$ denotes the external and recurrent input (see Methods). The DHO parameters jointly control the relaxation dynamics of the unit and can be thought of as the unit's innate adaptation to the characteristics of the signals to be processed by it (Fig. 1B). Despite their simplicity, DHO units can capture the essential dynamical characteristics of different types of neural mass models [36, 37, 26, 38] (see Supplementary Materials). Thus, the oscillatory properties of the nodes implemented in our simulations are both physiologically plausible and experimentally confirmed [37, 26, 36, 38]. Already a single DHO unit possesses intriguing computational capabilities not accessible to non-oscillating units such as leaky integrators. For example, DHOs

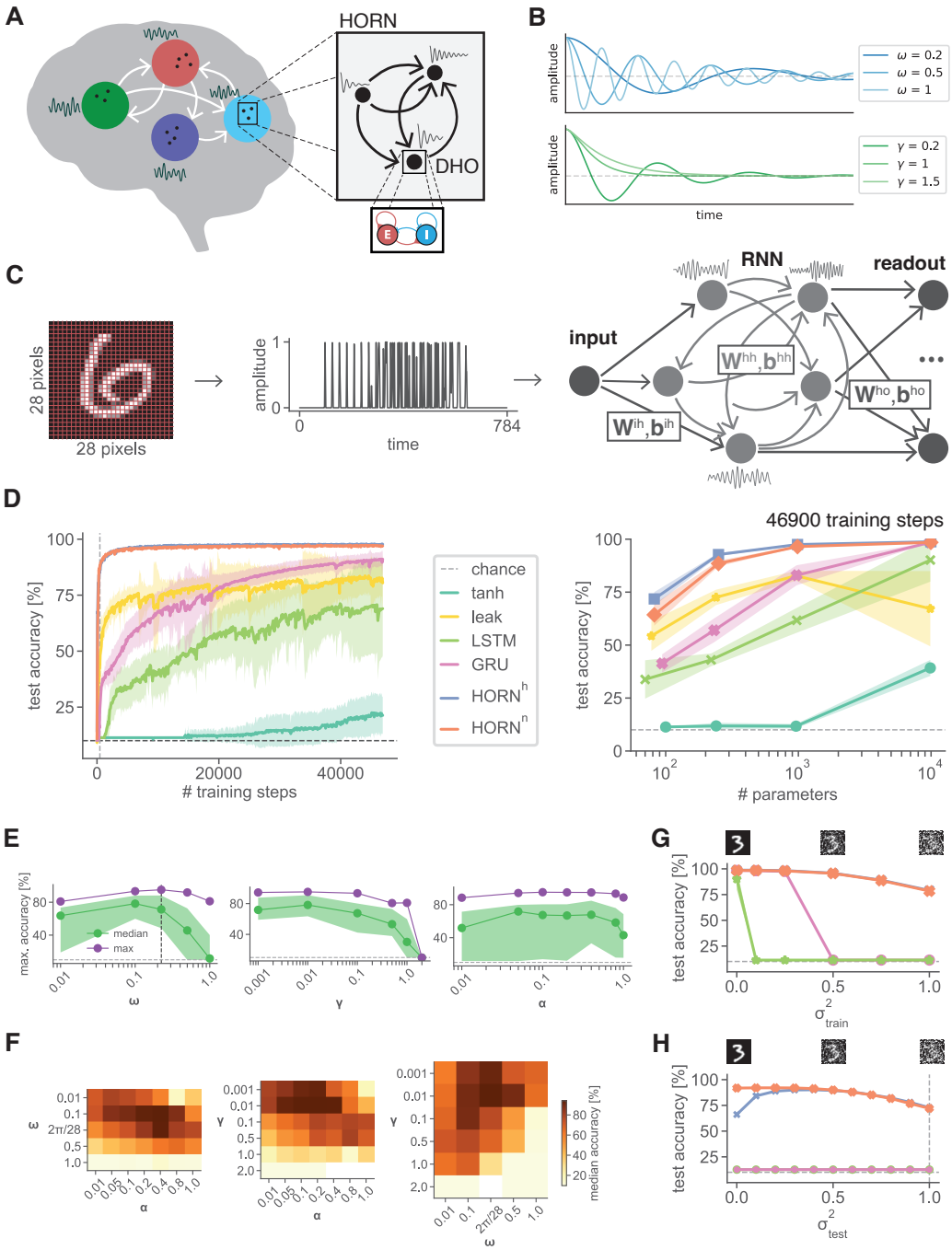

Fig. 1. Architecture of HORN networks and their performance in pattern recognition tasks. **A**. A HORN network composed of DHO units as a model of mesoscale brain activity. Each DHO unit models the compound activity of a recurrently connected E-I population. **B**. Influence of the parameters ω (natural frequency, top) and γ (damping factor, bottom) on the relaxation dynamics of an isolated DHO unit. Initial condition $x_0 = 1$, $y_0 = 0$. Dashed line indicates $x = 0$. **C**. Serialization of an MNIST digit to a time series (i.e. an sMNIST stimulus, left and middle panel) and feeding the stimulus into an RNN with trainable parameters (right panel): $\mathbf{W}^{\text{ih}}, \mathbf{b}^{\text{ih}}$ (input weights and biases), $\mathbf{W}^{\text{hh}}, \mathbf{b}^{\text{hh}}$ (recurrent weights and biases), and $\mathbf{W}^{\text{ho}}, \mathbf{b}^{\text{ho}}$ (readout weights and biases). For sMNIST, all networks are trained using a readout after 784 time steps. **D**. Maximal test accuracy on sMNIST attained for different RNN architectures as a function of training steps over 100 training epochs (left) and after 100 gradient descent update steps as a function of trainable system parameters (right). Lines show mean accuracy over 10 network instances with random weight initialization, shaded areas standard deviation. **E**. Distributions of maximal test accuracy on sMNIST of HORN^{h} networks (32 units, 10 training epochs) with variable parameter values ω (natural frequency), γ (damping factor), α (excitability). Each panel marginalizes the values for the remaining parameters. Lines show median and maximum, shaded areas indicate 25% and 75% percentiles. Note that the maximal network performance peaks around a value of $\omega = 2\pi/28$ (dashed vertical line, see main text). **F**. Median test accuracies of HORN^{h} networks as in E. Each matrix entry shows the median test accuracy (color scale on the right) for the marginalized remaining parameter. **G**. Maximal test accuracy of different RNNs architectures (10^4 trainable parameters) after training on noisy sMNIST for 100 epochs as a function of training noise level σ^2_{train}. Colors as in D. **H**. Mean test accuracy of different network architectures trained on noisy sMNIST with $\sigma^2_{\text{train}} = 1$ (dashed vertical line) as a function of test stimulus noise level σ^2_{test}, calculated over $n = 1000$ test stimuli. Colors as in D.

can gain-modulate an input signal in a nonlinear way as a function of its frequency profile and encode stimulus information in the oscillation phase (fig. S1C and Supplementary Information). An interplay of the input nonlinearity and the feedback connections was found to equip the units with a dynamical repertoire that goes far beyond the one of a classic damped harmonic oscillator and, for example, allows them to express self-sustained oscillations or resonate at fractional harmonics of their natural frequency (see figs. S2,29 and Supplementary Materials).

Homogeneous networks

We first considered a homogeneous HORN (HORN^{h}) with all-to-all connectivity, no conduction delays, and all nodes having identical parameter values ω, γ, α. The network nodes were additively coupled on the velocity term and trained in a supervised way using the BPTT algorithm (see Methods). The coupling on the velocity term allows HORN units to communicate changes in their amplitude to connected neighbors. This choice of coupling is motivated by the fact that cortical circuits are known to operate in the balanced-state regime where converging excitatory and inhibitory currents cancel in the mean, and activity is fluctuation driven. Stimuli were presented to the networks in the form of time series data. To form a time series version of an MNIST image, intensity values were collected in scan-line order from top left to bottom right, and we refer to the time series version of MNIST as sMNIST (see Fig. 1C and Methods). For the sMNIST data set, a set parameters values ω, γ, α (Table S1) resulting in a HORN (32 units) with high task performance was determined using a grid search (Fig. 1E,F and Methods). We found that the optimal parameter values are insensitive to network size (data not shown) and that networks with low values of the damping factor γ tend to achieve the highest task performance on sMNIST (Fig. 1E,F). Such low values of the damping factor γ put the system in a highly oscillatory state and also allow longer memory spans both within each DHO unit and the whole network (Fig. 1B). The value of the excitability parameter α was found to have no strong effect for performance on sMNIST, with intermediate values of alpha between 0.2 and 0.4 resulting in the best performance. Optimal values of the natural frequency parameter ω were found to be around the value $\omega_v = 2\pi/28$, the fundamental frequency of a straight vertical line in the 28x28 MNIST pixel space when transformed into a time series (see Fig. 1E,F and Methods). This choice of natural frequency enables a DHO unit to resonate to frequencies around ω_v and to extract and store stimulus information carried in the characteristic frequency spectra of continuous lines of different slopes in MNIST stimuli. The value ω_v also coincides with a peak in the variance of power spectral densities (PSDs) calculated on a representative set of sMNIST samples (fig. S4), and ω values leading to well-performing HORNs can be identified from such peaks.

We found that these simple HORN^{h} networks outperformed state-of-the-art gated RNN architectures with respect to learning speed (fig. S6), absolute performance, and noise tolerance at the same number of learnable parameters, particularly in the region of low parameter counts (see Fig.1D,G,H). Decreasing the signal-to-noise ratios or increasing the number and similarity of the test patterns tended to increase the differences in performance between the HORNs and the other networks (figs. S6, 7, 8, 9).

We identified several reasons for the observed gain of function in HORN networks: (i) Unit parameters ω, γ, α act as a smoothing prior of the trajectories of system dynamics that regularize gradients, making BPTT learning more stable (see Methods). (ii) Innate stimulus specificity (as controlled by the values of ω, γ, α) allowed single units to efficiently code for stimulus features already in an untrained state, and learning-induced adjustments of the feedback connections allowed units to adjust their stimulus specificity. (iii) The kinetics of each DHO unit result in dynamic phenomena such as fading memory, resonance, entrainment, synchronization, phase shifts, and desynchronization both on the single unit and on the network level. Together, these mechanisms allow for the generation of high-dimensional dynamic landscapes in the state space of the network.

Before learning, the dynamics of HORN^{h} networks are dominated by large-scale synchronization among the nodes (Fig. 2A,C). As learning progresses, global synchronization decreases (Fig. 2C) and

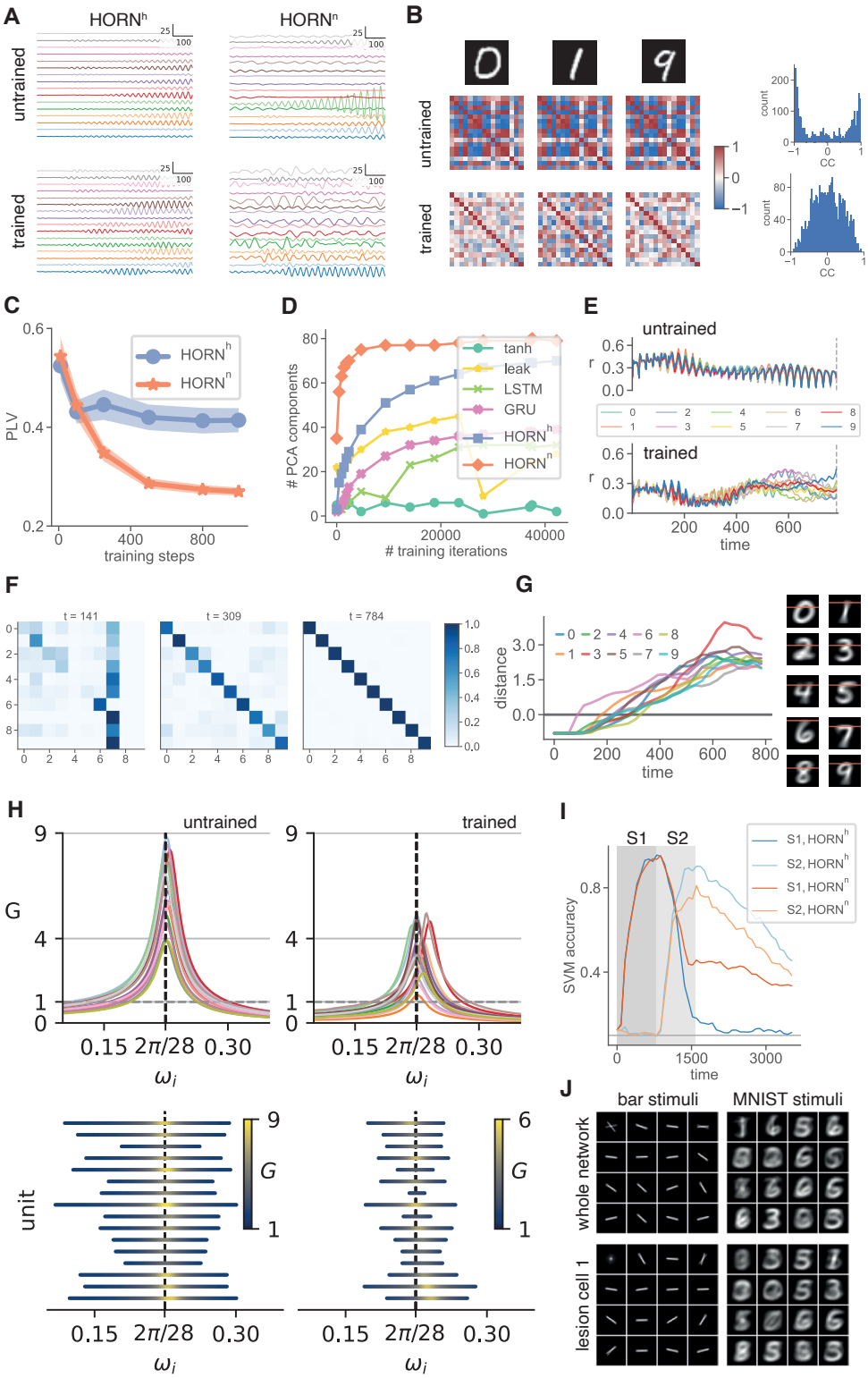

Fig. 2. Dynamics and computational principles in HORNs. **A**. Network activity elicited by an sMNIST stimulus (digit 0) of 16-unit HORNs trained on sMNIST, before (top row) and after (bottom row) training. Left column HORN^h, right column HORN^n. Note the highly irregular temporal dynamics of the HORN^h in the trained state. **B**. Pairwise cross-correlation coefficients of unit activity for different input stimuli in HORN^h (16 units) before (left, top row) and after training (left, bottom row) on sMNIST for 10 epochs. Distribution of cross-correlation coefficients before (right, top) and after (right, bottom) training. **C**. Pairwise phase locking value (PLV) calculated on $n = 100$ test stimuli as a function of training steps for a homogeneous (blue) and heterogeneous (orange) HORN network (16 units) trained on sMNIST. Lines show mean, shaded region indicates standard deviation. **D**. Number of PCA components explaining 99% of the variance of observed network activity as a function of training steps for different RNN architectures (10^4 trainable parameters). PCA performed on network activity data resulting from 10000 test stimuli (see Methods). **E**. Dynamics of mean Kuramoto order parameter r of HORN^h (16 units) activity trained on sMNIST in the untrained network (top) and after training for 10 epochs (bottom). Lines show mean values per digit class. The HORN learns to generate dynamic, stimulus-specific patterns of higher-order synchrony. **F**. Normalized confusion matrices of linear SVMs trained on stimulus classification from activity of a HORN^h (93 units, trained on sMNIST for 100 epochs) activity at different time points of the stimulus presentation period. Matrices show relative frequencies the number of predictions obtained over 10^4 samples. The vanishing of the non-diagonal entries (indicating misclassified stimuli)

shows that the network performs evidence accumulation over the stimulus presentation time. **G**. Mean distance to SVM decision boundary for SVMs as a function of presentation time for the dataset depticted in F. The color of the curves refer to different digits (insert). Note the steady accumulation of evidence. Right: Class-averaged MNIST digits with the image row corresponding to the decision boundary crossing of the SVM colored in orange. **H**. Top: Intrinsic RFs (gain functions G) of units in a 16-unit HORN^{h} before (left) and after (right) trained on sMNIST as a function of input frequency ω_i. Gain values $G > 1$ indicate amplification. Black dashed lines indicate the frequency $\omega_v = 2\pi/28$ that corresponds to a straight vertical bar in pixel space. Colors indicate unit identity. Bottom row: Input frequency dependent gain changes of G of units before (left) and after (right) training. Same data as in top panels. Note the narrowing of the intrisic RFs and the shifts in preferred frequency with training. **I**. Accuracy of stimulus decoding by linear SVMs trained on the activities of different HORN networks (93 units) that were stimulated with a sequence of two temporally segregated stimuli S1, S2 (shaded areas indicate stimulus timing). Light and dark colored curves indicate SVM decoder performance for S1 and S3 for a homogeneous (HORN^{h}) and non-homogeneous (HORN^{n}) network, respectively. Note the longer persistence and superposition of stimulus specific information in the HORN^{n}. **J**. Upper panels: Effective RFs of a HORN^{h} (16 units) when mapped with contour stimuli from LSDSa (left) and sMNIST digits (right). Lower panels: Same network and stimuli, but after deletion (lesion) of the first unit. Note the reconfiguration of most of the eRFS.

---

this drives an increase in the dimensionality of network dynamics (Fig. 2D). In parallel, we found a learning-induced emergence of complex, stimulus-specific correlation patterns (Fig. 2B). This increase in complexity of the spatiotemporally structured activity landscape enhances segregation of stimulus-specific activation patterns by a linear read-out, and thereby improves classification performance. To assess higher-order synchronization phenomena and their temporal dynamics, we calculated the mean Kuramoto order parameter r of network dynamics [39]. We found that HORNs develop stimulus-specific higher-order synchrony patterns that are highly variable over time (Fig. 2E).

To assess noise tolerance, we trained HORNs on sMNIST digits corrupted by increasing levels of additive white Gaussian noise $N(0, \sigma^2)$ that was applied per pixel (see Methods, fig. S5). Stimuli were clipped to the range $[0, 1]$ after noise application, further increasing task difficulty with increasing levels of σ^2. Unlike other architectures, HORNs show only a gradual decline in task performance (Fig. 1G) and the stimulus representations in HORNs were found to be robust to changes in stimulus noise characteristics (Fig. 1H). This noise resilience prevails even when networks are trained on shuffled sMNIST stimuli (pMNIST), in which the dominance of low-frequency information characteristic for the original MNIST digits is destroyed (figs. S5, S7). This robustness can be explained by the strong attenuation of high-frequency signals due to the nonlinear frequency-dependent gain modulation of input signals at each DHO node (see Receptive Fields below).

Heterogeneous networks

The structural and functional organization of cortical networks is characterized by heterogeneity [40, 41]. Whether this variability has a functional role is unclear. In HORN^{h} networks, learning created classifiable patterns through desynchronization and the generation of more heterogeneous dynamics. To test whether increasing network heterogeneity facilitates learning, we simulated non-homogeneous HORNs (HORN^{n}) in which each node had a different natural frequency, damping coefficient, and excitability coefficient (see Methods). As expected, compared to homogeneous HORNs, the heterogeneous HORNs responded already in the untrained state with more complex and less globally synchronized patterns to different stimuli, and synchrony decreased further as learning progressed (Fig. 2A,C). The observed decrease in synchrony also leads to a significant increase in the dimensionality of the dynamics of heterogeneous HORNs compared to their homogeneous counterparts (Fig. 2D). Apparently, the learning process could capitalize on the initial diversification of response patterns and required fewer training steps to reach high-performance levels (Fig. 1D). A comparison between homogeneous and heterogeneous HORNs revealed superior performance of the latter with respect to learning speed and noise

tolerance (Fig. 1D,G,H). These advantages were especially prominent for larger networks, more difficult classification problems, and higher levels of stimulus noise (figs. S6, 7, 8, 9).

In contrast to RNNs composed of gated units that often have chance-level decodability by a linear readout in the untrained state, HORNs typically have much higher decodability even in the untrained state (fig. S10). This suggests that HORNs could serve as efficient reservoirs [20]. To examine this possibility, we simulated HORNs with fixed recurrent weights and trained only the input and readout weights with BPTT. These reservoir HORNs showed a performance drop compared to the corresponding regular HORN (fig. S11), but we hypothesize that they could serve as efficient reservoirs when network size is increased.

Self-connection terms

The DHO self-connection parameters capture dynamical properties of the circuit a given unit represents and allow a unit to tune its oscillation frequency and relaxation dynamics (see Supplementary Information). To assess the role of these feedback parameters for network performance, we trained HORNs with different connectivity patterns on sMNIST and pMNIST (figs. S12,13). We found that networks consisting of fully isolated units were still able to perform sMNIST classification with a high level of accuracy if adjustments to the self-connection strengths were allowed. In this group of isolated experts setting, the units learn to independently code for different stimulus features by adjustments of their self-connection terms (see Receptive Fields) in a way that their combined dynamics allow for stimulus classification. By contrast, if only self-connections were disabled but not the inter-unit connections, the recurrent dynamics could compensate for the lack of self-connections.

Evidence accumulation

The response patterns of HORNs evolve over time as a result of recurrent network reverberations. To find out at which point in time the network converges to states of maximal stimulus specificity, a linear SVM was trained on stimulus classification using activity data of a trained HORN at different time points throughout the stimulus presentation period. We find that the network accumulates evidence, with its dynamics allowing progressively better decoding of the stimulus identity by an SVM as the network approaches the read-out time $t = 784$ on which it was trained (Fig. 2F,G). Furthermore, we find that most sMNIST stimuli can be decoded before $t = 400$ (Fig. 2F), i.e. already when just the upper half of the digit has been presented to the network. Surprisingly, we also found a bias for the digits 6 and 7 in the MNIST data set that surfaced due to the sequential presentation of the digit pixels. Apparently, the digits 6 and 7 have a bias of being vertically shifted upward and downward compared to the rest of the digits, respectively (Fig. 2G, right). HORN network activity allows the readout SVMs to exploit this bias (Fig. 2F,G).

Due to reverberating network activity, HORNs exhibit fading memory and superimpose information about successively presented stimuli. For example, after the successive presentation of two stimuli, a linear SVM can decode the two stimuli from the same segment of reverberating activity (Fig. 2I). This property of fading memory and stimulus superposition resembles the physiological results obtained from the visual cortex [42].

Receptive fields

To better understand the principles of computation in HORNs and how these emerge over training, we investigated how learning changed the response properties of both individual nodes and the entire

network. This would be similar to experience- or learning-dependent modifications of receptive fields (RFs) in natural systems [43].

Each DHO unit in a HORN has a gain curve $G(\omega_i)$ that describes how the unit modulates the amplitude of an input signal as a function of the input signal frequency ω_i. The shape of G is determined by the values of ω, γ and the self-connectivity term v. We call G the *intrinsic receptive field* (iRF) of the unit as it defines the frequency band in which the unit shows feature selectivity. During learning, adjustment of the self-connection weight v alters G and thereby drives changes in the iRF of each unit. This enables BPTT learning to modify the iRF of each DHO unit, allowing the unit to improve its selectivity for stimulus features useful for performing a given task. In homogeneous HORNs where ω, γ, α are constant between units, the intrinsic RFs of all units are initially very similar and even after training only differ in the shift of G induced by modifications of v (Fig. 2H). For heterogeneous HORNs, the iRFs of the untrained network already span a large portion of the frequency space, with units more widely tuned to particular features in frequency space and coding in more diverse frequency bands (fig. S14).

For comparison with natural systems, we mapped the receptive fields of the nodes in the same way as is common practice in electrophysiological studies [44]. We examined which stimuli activate a unit most strongly and address the so-defined RF as *effective RF* (eRF). The structure of the eRF varies greatly depending on the stimuli presented, since it depends not only on the stimulus but also on network connectivity and dynamics. When mapped with very simple, isolated stimuli, neurons in sensory cortices exhibit canonical RFs [44], but these are no longer detectable when more complex stimuli are used that induce large-scale network interactions [45]. We determined the eRFs of units in our models for two canonical choices of stimulus classes: (i) simple line segments of different orientations, as commonly used for RF mapping in visual experiments [44] (fig. S15), and (ii) the original MNIST digits on which the networks were trained, as an example of complex natural stimuli [45].

To measure eRFs, we simulated a 16-unit HORN^{n} trained on MNIST and calculated for each node the mean stimulus that resulted in maximal unit activation. When stimulated with simple line stimuli, we find simple orientation-selective eRFs that closely resemble the RFs of neurons in the primary visual cortex (Fig. 2J, top left). When mapped with complex stimuli, some nodes exhibited specificity for particular features of the MNIST digits (Fig. 2J, bottom left). Other nodes had eRFs that seemed to be unrelated to the stimuli in the training set, and yet others were inhibited.

To determine the influence of recurrent network dynamics on eRFs, we silenced one node of the network. The result was a dramatic reconfiguration of the eRFs of other nodes (Fig. 2J, right, fig. S14 for the case of a heterogeneous network). This indicates that after learning, eRFS are to a large extent the result of dynamic interactions in the network. Thus, the eRFs of HORN units undergo context-dependent dynamic modifications similar to those of natural systems [46].

To obtain a more quantitative description of the nodes' eRFs in frequency space, we additionally stimulated networks with harmonic sine-wave stimuli of different frequencies and determined the frequency responses of the nodes (fig. S16). As expected, most of the nodes resonated best at their preset natural frequencies in untrained networks. However, this preference changed substantially over the course of training, confirming the strong influence of network interactions on the structure of eRFs.

Delays

In the cerebral cortex, neurons interact through relatively slow conducting nerve fibers (0.5 to 10 m/s), which introduces considerable, widely scattered, and distance-dependent coupling delays [47]. To test the influence of introducing coupling delays on task performance, we started with a HORN^{h} and endowed all recurrent connections with variable coupling delays measured in time steps and chosen from a uniform distribution $[1, d_{\max}]$ (see Methods). We found that the introduction of coupling delays increased HORN performance in both maximal classification accuracy and learning speed (Fig. 3A). Furthermore, increasing $d_{\max}$ which results in greater heterogeneity, was found to increase task performance, and this

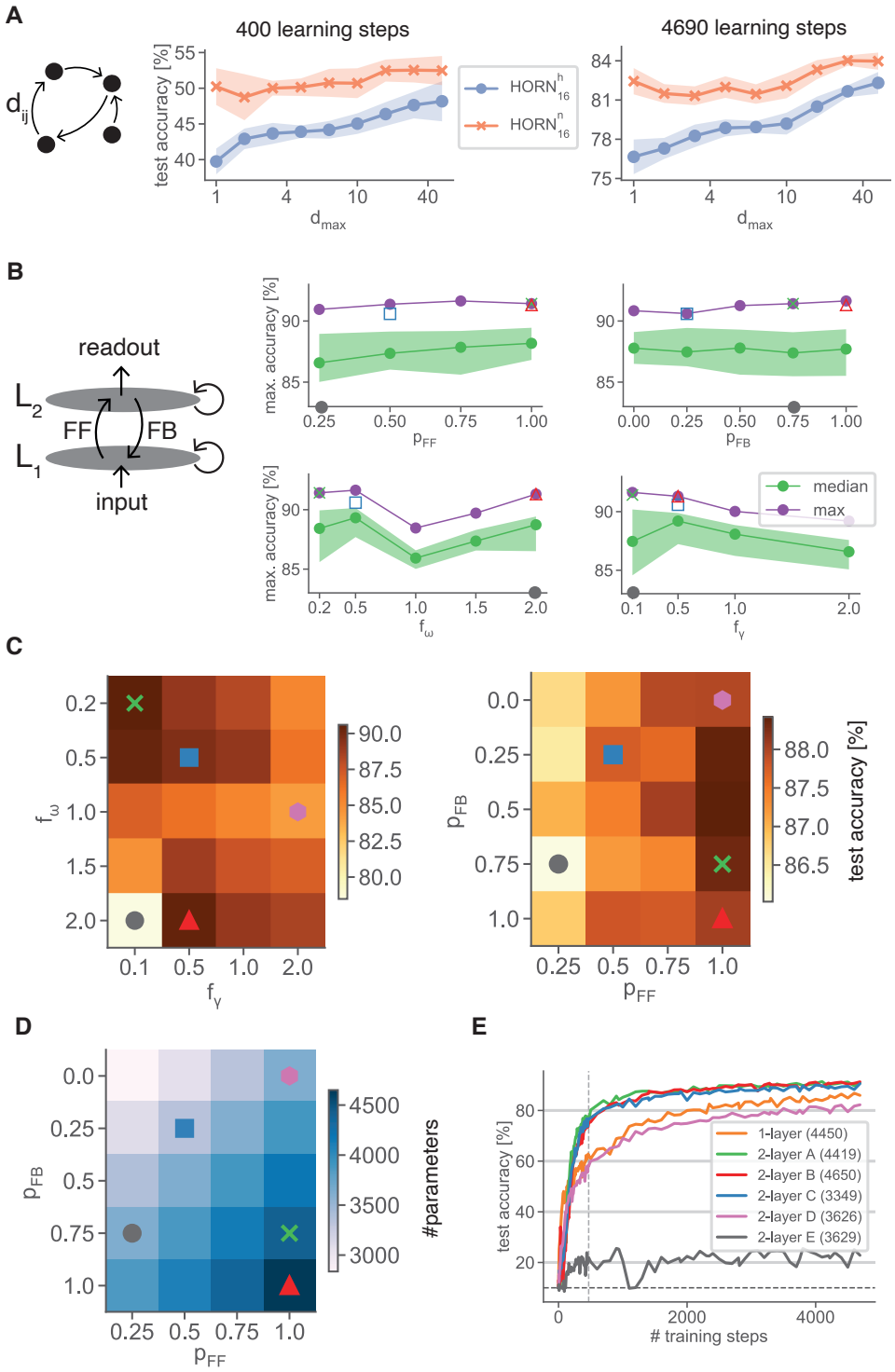

Fig. 3. Biologically-inspired extensions to HORN networks. **A**. Performance of HORNs with connection delays. Maximal test accuracy on pMNIST after 400 training steps (left) and after 4690 training steps (right), respectively, as a function of maximal synaptic delay $d_{\max}$. For each network, delays were sampled from a uniform distribution $U([1, d_{\max}])$. A network with $d_{\max} = 1$ corresponds to a regular HORN. Lines show mean performance over 10 randomly initialized networks, shaded areas standard deviation. **B**. Two-layer HORN^{h} (32 +32 units) performance as a function of the 4 parameters $p_{\text{FF}}, p_{\text{FB}}$ (fraction of feed-forward and feedback connections present, upper panels), f_ω, f_γ (scaling factor of ω and γ between the first layer and the second layer, lower panels). Curves represent median and maximal values of marginal distributions of the maximal accuracy on pMNIST attained during training for each value of the 4 parameters. Colored symbols refer to five exemplary networks whose parameter constellations are shown in C. Note the drop in performance when L1 and L2 have the same preferred oscillation frequency and that increasing connection sparsity does not result in a strong performance drop (gray circle marked on axes lies outside of the accuracy range displayed). **C**. Median HORN^{h} performance (color scale on the right) for the different pairs f_ω, f_γ values (left) and p_{FF}, p_{FB} (right). Symbols refer to the same networks as in B. Note that different parameter combinations support equally high performance. **D**. Number of trainable parameters of HORN networks (color code on the right) as a function of various combinations of p_{FF}, p_{FB}. Symbols refer to the same networks highlighted in B and C. Note that certain network configurations show high performance despite lower numbers of trainable parameters (green cross, red triangle, blue circle). **D**. Evolution of accuracy as a function of learning steps of the 5 examplar two-layer networks marked by symbols in B, C, D (colors of curves correspond to symbol colors) in comparison to a single layer HORN^{h} with a comparable number of trainable parameters. Number of trainable parameters in parentheses.

gain of function increased with increasing values of $d_{\max}$ (Fig. 3A). Thus, as was the case with preferred oscillation frequencies, increasing heterogeneity by introducing inhomogeneous conduction delays also enhances HORN performance. As before, heterogeneity increases the dimensionality of the network's state space and enables the generation of more diverse spatiotemporally structured activity landscapes.

Multi-layer networks

In the mammalian cerebral cortex, sensory signals are processed in hierarchically organized cortical areas that are reciprocally coupled [48]. To investigate the potential benefits of distributed multistage processing, we generated two-layer homogeneous HORNs consisting of a "lower" and an "upper layer" (L1 and L2, both consisting of 32 units, $\text{HORN}^{\text{h}}_{32,32}$), and implemented sparse reciprocal connections between the layers (Fig. 3B). The input signals were presented to L1 as before, and the result was read out at the nodes of L2. We performed a grid search on the feed-forward and feed-back connection probabilities f_{F}, f_{B}, as well as the scaling factors f_ω, f_γ that controlled the scaling of the parameters ω, γ with respect to the first layer. For each parameter configuration, we trained a $\text{HORN}^{\text{h}}_{32,32}$ and assessed task performance by means of the maximally obtained test accuracy on pMNIST within 10 training epochs. We found that for all values of each parameter, there exists a parameter configuration that results in a high-performing network (close to 90% accuracy) (Fig. 3B). In addition, we show learning dynamics for four selected two-layer $\text{HORN}^{\text{h}}_{32,32}$ and a single-layer HORN^{h} with a comparable number of trainable parameters in Fig. 3C-E. In particular, Fig. 3E shows two examples of two-layer HORNs that exceed the performance of a single-layer HORN while having lower numbers of trainable parameters. Higher task performance could be observed when both the preferred oscillation frequencies and the damping coefficients were lower in the upper layer (Fig. 3B, bottom row). In this setting, the dynamics of the faster lower layer is partly opaque to that of the slower upper layer because of the stronger attenuation of high frequencies in the upper layer (see Receptive Fields). Layered networks can therefore operate at different frequency bands and perform parallel analyses of input patterns at different temporal scales. Furthermore, we found that a two-layer network in which the separation of frequency bands across layers breaks down fails to learn (Fig. 3E). This provides another hint as to why it might be advantageous to endow the nodes of higher layers with lower preferred frequencies.

Analogous analysis of oscillatory activity in the mammalian brain indicates that the preferred oscillation frequencies of the cortical areas decrease from low to high levels of the processing hierarchy [10]. This enhanced temporal integration is paralleled by an enhanced spatial integration, as indicated by the increase in RF sizes at higher processing levels [49]. Taken together, this suggests that the span over which temporally and spatially segregated features can be related to each other increases as one proceeds along the processing hierarchy.

Learning priors

In the primary sensory cortices of the mammalian brain, statistical relationships between features of the natural environment become installed in the architecture and the synaptic weight distribution of reciprocal connections by learning [50, 2]. This stored information can then serve as a prior for perceptual grouping and scene analysis. To test how the installation of priors in HORNs influences their learning and task performance, we first installed a set of canonical priors by training a HORN^{n} (93 units) to discriminate simple elongated contours of different orientations placed at random locations in the MNIST pixel matrix using the LSDSb data set (see fig. S15 and Methods). We then froze the input and recurrent weights and trained only the readout layer on the sMNIST classification task with BPTT. We found that the stimulus representation resulting from the installation of canonical priors allowed the pre-trained HORN to perform MNIST digit classification after training of the readout layer (fig. S17). Only freezing the input weights and training both the recurrent weights and the readout weights on the other hand

lead to a greatly increased learning speed (fig. S17). This indicates that a priori knowledge of the statistical contingency of features substantially facilitates classification of complex objects composed of such elementary features in HORN networks. To test how HORNs respond to unfamiliar stimuli, we trained a HORN on sMNIST data and then presented shuffled sMNIST stimuli (pMNIST) to the trained networks. We found that the networks had learned to produce activity that well separates different classes of the "natural" sMNIST stimuli, but that this was not the case for the unfamiliar pMNIST stimuli (fig. S18). This is reminiscent of recent experimental findings in the visual cortex of behaviorally trained monkeys. These revealed that population vectors of multi-unit activity contained more stimulus-specific information when evoked by natural scenes than by manipulated stimuli, in which part of the statistical regularities of natural scenes were removed [51].

Hebbian Learning

Theoretical considerations suggest that biological neuronal networks learn to represent stimuli through Hebbian modifications of the synaptic gain of recurrent connections [52]; connections strengthen between nodes that often engage in correlated activity and weaken between nodes exhibiting mostly decorrelated responses. Thus, neurons coding for features that frequently co-occur become more strongly coupled and therefore self-organize to form a functionally coherent, stimulus-specific assembly [53].

To test whether the gradient-based training of HORNs produces weight distributions compatible with those predicted by Hebbian learning principles, we compared recurrent weight distributions and the corresponding correlation structure of responses before and after learning. We found that this indeed is the case (Fig. 4A,C). The BPTT learning algorithm apparently is able to capitalize on the stimulus-specific correlation structure emerging from the dynamic interactions in the network and enhances those activity patterns that are characteristic of a particular stimulus, as predicted by a Hebbian mechanism. The observed effect was found to be stronger for heterogeneous HORNs (Fig. 4C). This can be explained by the fact that heterogeneous HORNs respond already in the untrained state with more desynchronized network activity, while homogeneous HORNs first need to learn to desynchronize to increase the dimensionality of their state space and task performance (Fig. 2A,B). To investigate how correlation-based learning could work in HORNs, simple unsupervised additive Hebbian and anti-Hebbian rules acting on the recurrent connections were implemented (see Methods). During training with the Hebbian plasticity rules, we disabled backpropagation learning on the recurrent connections, while keeping it active for the input and readout connections. As a performance baseline, we trained a reservoir HORN where the gain of the recurrent connections was frozen and only the input and the readout connections were trained. We found that in the case of homogeneous networks that expressed highly synchronized dynamics in the untrained state, the fully unsupervised anti-Hebbian rule that desynchronizes network activity resulted in a task performance comparable to that attained by the fully supervised BPTT rule, and a Hebbian rule also led to performance increases above baseline (Fig. 4B). For heterogeneous networks that already express more decorrelated activity in the untrained state (Fig. 4A), a Hebbian rule was able to increase performance above baseline, but this was not the case for an anti-Hebbian rule (Fig. 4C). This provides a proof of principle for unsupervised Hebbian learning in HORNs (Fig. 4B,D) and lets us hypothesize that heterogeneous networks could facilitate Hebbian learning due to their tendency to express more decorrelated activity ab initio.

Geometric input drives self-organization

To test how HORNs process simultaneous geometric input as encountered by the visual system, we trained networks that receive spatially organized input in the form of MNIST digits. For this, we endowed each unit of a HORN^{h} with a RF of 2x2 pixels so that the 28x28 pixel matrix of an MNIST digit

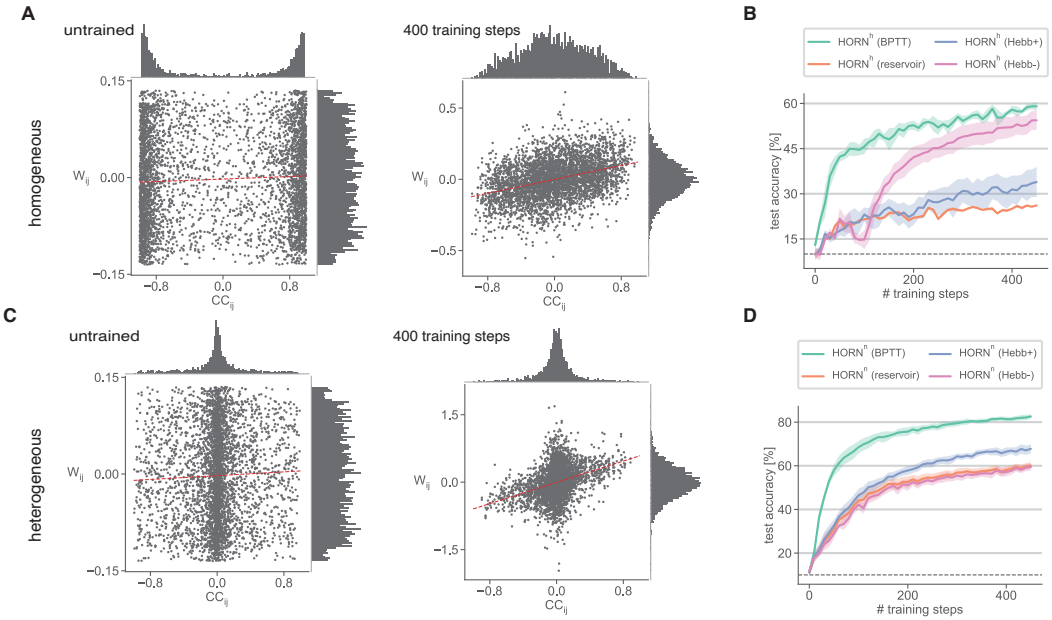

Fig. 4. Backpropagation learning in HORN networks results in Hebbian-like weight changes. **A**. Scatter plots of connection weights $\mathbf{W}^{\mathrm{hh}}_{ij}$ and mean cross-correlation coefficients CC_{ij} of unit activities of a homogeneous HORN^{h} (64 units) before training (left) and after training on pMNIST for 500 training steps (right). CC_{ij} computed over 100 samples. Linear regression lines are shown in red. Marginal distributions of $\mathbf{W}^{\mathrm{hh}}_{ij}$ and CC_{ij} are shown on the right and top, respectively. Note the bimodal distribution of correlation coefficients with modes around -1, 1 in the untrained state and the more decorrelated network activity as a result of learning. **B**. Performance of a homogeneous HORN^{h} (64 units) network as a function of training steps when instances of the same network are trained with correlation-based Hebbian (suffix "Hebb+") or anti-Hebbian (suffix "Hebb-") learning rules in comparison to instances trained with BPTT (suffix "BPTT") and when $\mathbf{W}^{hh}$ was fixed (suffix "reservoir"). The input and readout parameters are trained with BPTT for all instances. Curves show mean performance over 10 network instances with random weight initialization, shaded areas standard deviation. Note the strong performance of the anti-Hebbian rule for this initially highly synchronized network. **C**. Scatter plots of the connection weights and cross-correlation coefficients of a heterogeneous HORN^{n} as in A. Note the more decorrelated activity compared to a HORN^{h} already in the untrained state. **C**. Performance of different instances of a heterogeneous HORN^{n} (64 units) network as a function of training steps when trained with different learning rules. Learning rules as in B. Note the baseline (reservoir HORN) performance of the anti-Hebbian rule in this initially more desynchronized network (compared to the case in B) and the performance increase compared to baseline resulting from a Hebbian rule.

---

was covered by the RFs of a 196 unit network (Fig. 5A). To provide input to the network, we activated each unit for one time step at an intensity corresponding to the sum of the pixel intensity values within its RF (Fig. 5A). We then gave the network 150 time steps to process the stimulus before performing a read-out (Fig. 5B). For training, we started with an all-to-all random connectivity and kept the weights of the input layer fixed, while the recurrent and readout weights were plastic. After training for 10 epochs, the classification accuracy on the test set was found to be around 90% (data not shown). When stimulated, each of the simultaneously activated nodes responded with a damped oscillation (Fig. 5B). These activities spread across the network and led to complex interference patterns and traveling waves. The direction and shape of the waves differed for responses evoked by different stimuli (Movies S1,S2).

Analysis of the temporal evolution of responses and of learning-induced changes revealed several phenomena that closely resemble observations in natural systems. The brief direct activation of the nodes covered by the stimulus was followed by a period of relatively low activity that then gradually increased to levels far above those of the initial stimulus-driven response (Fig. 5B). Similar dynamic effects have been observed in the visual cortex in response to flashed stimuli [53], where this amplification of the input signal was hypothesized to be due to reverberating interactions among the nodes that eventually lead to the convergence towards a metastable stimulus-specific dynamic state. This state can be interpreted as the result of a highly parallelized search for the best match between sensory evidence and

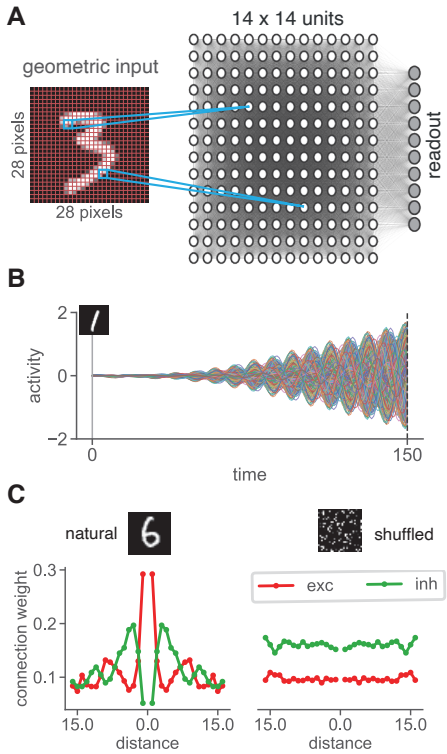

Fig. 5. Self-organization in HORNs driven by geometric input. **A**. Schematic of geometric input provided to an all-to-all connected HORN. Each DHO unit receives input from its 2x2 pixel receptive field. Input weights are constant and fixed, recurrent and readout weights are trainable. Readout units are connected to the entire HORN. **B**. Example network dynamics of a HORN^{h} (196 units) stimulated as in A after training with BPTT for 2000 steps (achieved classification accuracy around 90% for both natural and shuffled stimuli). Input is flashed to the network during one time step at $t = 1$. Readout is performed after 150 time steps of network activity during which no input is given. Curves represent the activity of all units in the network (unit identity is color coded). Note the increasing amplitudes of reverbrating activity and the drifts in phase of the oscillatory responses. **C**. Distance-dependent connection weights of excitatory (red) and inhibitory (green) connections in HORN^{h} networks after training with BPTT for 2000 steps on geometric stimuli. Stimuli were either natural (left) or shuffled (right) MNIST digits. Note the Mexican-hat-like connectivity structure among nodes in networks trained on natural stimuli and the lack of structure in the connectivity in networks trained with shuffled stimuli.

the priors installed by learning [2]. Such matching operations are the core of predictive coding strategies, and experiments in the visual cortex provide preliminary support for this notion [53]. Another interesting observation was that after learning, nodes initially activated by the stimulus exhibited synchronized oscillations whose amplitude increased as time elapsed. Thus, nodes activated by a particular stimulus become associated with one another by self-organizing interactions and form a cooperating, stimulus-specific assembly distinguished by synchronized and jointly enhanced responses. Such dynamics have also been observed in cortical networks [11, 53] and this has led to the binding by synchrony hypothesis (BBS) [29]. Moreover, such selective synchronization effects are to be expected if learning modifies recurrent connections according to a Hebbian correlation rule [2].

Analysis of learning-dependent circuit modifications in our networks revealed that the initial unspecific all-to-all connections had changed in a way that reflected the geometric distance relationships present in the stimuli. When HORN units are placed on a regular grid according to the geometric organization of their RFs, connections in the trained network were pruned down to spatially restricted connections, resulting in a distant-dependent decay of coupling strength following a Mexican-hat-like shape (Fig. 5C, left). Such connectivity favors nearest-neighbor interactions, and thus captures the essential structural feature of our MNIST stimuli, the continuity of their contours, and the spatial vicinity of activated nodes. When trained on shuffled MNIST stimuli (fig. S5) on the other hand, the network performed with an equal classification accuracy of around 90%, but such distant connectivity patterns did not emerge (Fig. 5C, right). This means that the Gestalt criteria of continuity and vicinity that support feature binding in perception were installed by training in the architecture of the coupling connections of the HORN. The selectively enhanced coupling of the nodes participating in the representation of individual stimuli in turn accounts for the amplification of figure coding responses, as observed in our simulations and postulated by theories on assembly coding and binding by synchrony [2].

Discussion

In this work, we have shown how the implementation of physiologically plausible mechanisms in RNNs influences their dynamics and how this results in their substantially increased task performance in comparison to non-oscillating architectures. The configuration of nodes as damped harmonic oscillators, a quintessential model of oscillatory phenomena and motivated by experimental findings [54, 38], enables HORN networks to better exploit the temporal domain for computations than non-oscillating architectures due to the rich dynamics resulting from reciprocally coupled oscillators [10, 11, 13, 15].

These rich dynamics allow the networks to exploit for computations not only spatial distributions of response amplitudes but also temporal variables such as synchrony, asynchrony, nonlinear gain modulations, resonance, entrainment, and fading memory. The introduction of physiologically plausible heterogeneity and modularity (scattered preferences for oscillation frequencies, inhomogeneous coupling delays, experience-dependent anisotropy in coupling strength, multi-layer networks) was shown to further enhance performance without increasing the number of trainable variables by increasing the dimensionality of the networks' state spaces and enabling them to process stimuli at several time scales simultaneously by using different frequency bands. We showed that gradient-based BPTT learning resulted in weight distributions compatible with those predicted by Hebbian principles and provided a proof-of-concept of how biologically plausible unsupervised correlation-based learning could work in such networks. From a neurobiological point of view, our findings enable us to assign distinct functions to a number of experimentally verified phenomena whose functional role has been elusive or controversial. Plausible functional interpretations are now possible for (i) the propensity of nodes to oscillate and the ensuing dynamical phenomena such as synchronization, desynchronization, resonance, entrainment, and traveling waves [12, 13, 14, 15, 16, 17]; (ii) the diversity of preferred oscillation frequencies, their instationarity and context dependence [3, 4], (iii) the heterogeneity of conduction velocities of recurrent connections [5, 6], (iv) the Hebbian adaptivity of recurrent connections [7, 8]; (v) the emergence of context-dependent dynamic receptive fields by network interactions [43, 45], and (vi) the observed decrease of oscillation frequencies in higher areas of the cortical processing hierarchy [48]. We also hope that these results could instruct the design of artificial systems that are better able to take advantage of the temporal dimension for computation and that share some of the attractive features of natural systems.

Acknowledgments

The authors thank Martin Vinck, Andreas Bahmer, and Andreea Lazar for their helpful discussions and comments on an earlier version of this manuscript. We thank Claudia Kernberger for her help with some of the illustrations. We acknowledge computing support from the Ernst Strüngmann Institute and the Max Planck Computing and Data Facility. **Data and materials availability:** Model implementations and simulation scripts will be made publicly available upon publication. All data sets used in this study are publicly available.

References

[1] Nikola T. Markov, Maria Ercsey-Ravasz, Camille Lamy, Ana Rita Ribeiro Gomes, Loïc Magrou, Pierre Misery, Pascale Giroud, Pascal Barone, Colette Dehay, Zoltán Toroczkai, Kenneth Knoblauch, David C. Van Essen, and Henry Kennedy. The role of long-range connections on the specificity of the macaque interareal cortical network. *Proceedings of the National Academy of Sciences*, 110(13):5187–5192, March 2013.

[2] Wolf Singer. Recurrent dynamics in the cerebral cortex: Integration of sensory evidence with stored knowledge. *Proceedings of the National Academy of Sciences*, 118(33):e2101043118, August 2021.

[3] Rodolfo R. Llinás. The Intrinsic Electrophysiological Properties of Mammalian Neurons: Insights into Central Nervous System Function. *Science*, 242(4886):1654–1664, December 1988.

[4] Andreas K. Engel, Peter König, Andreas K. Kreiter, Thomas B. Schillen, and Wolf Singer. Temporal coding in the visual cortex: new vistas on integration in the nervous system. *Trends in Neurosciences*, 15(6):218–226, June 1992.

[5] Stephen G. Waxman. Determinants of conduction velocity in myelinated nerve fibers. *Muscle & Nerve*, 3(2):141–150, March 1980.

[6] H. A. Swadlow. Physiological properties of individual cerebral axons studied in vivo for as long as one year. *Journal of Neurophysiology*, 54(5):1346–1362, November 1985.

[7] Jeffrey C. Magee and Daniel Johnston. A Synaptically Controlled, Associative Signal for Hebbian Plasticity in Hippocampal Neurons. *Science*, 275(5297):209–213, January 1997.

[8] Henry Markram, Joachim Lübke, Michael Frotscher, and Bert Sakmann. Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs. *Science*, 275(5297):213–215, January 1997.

[9] Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, November 2013.

[10] György Buzsáki and Xiao-Jing Wang. Mechanisms of Gamma Oscillations. *Annual Review of Neuroscience*, 35(1):203–225, July 2012.

[11] Charles M. Gray, Peter König, Andreas K. Engel, and Wolf Singer. Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, 338(6213):334–337, March 1989.

[12] W Singer. Synchronization of Cortical Activity and its Putative Role in Information Processing and Learning. *Annual Review of Physiology*, 55(1):349–374, October 1993.

[13] Jessica A. Cardin, Marie Carlén, Konstantinos Meletis, Ulf Knoblich, Feng Zhang, Karl Deisseroth, Li-Huei Tsai, and Christopher I. Moore. Driving fast-spiking cells induces gamma rhythm and controls sensory responses. *Nature*, 459(7247):663–667, June 2009.

[14] Keith B. Doelling and David Poeppel. Cortical entrainment to music and its modulation by expertise. *Proceedings of the National Academy of Sciences*, 112(45), November 2015.

[15] Jean-Philippe Lachaux, Eugenio Rodriguez, Jacques Martinerie, and Francisco J. Varela. Measuring phase synchrony in brain signals. *Human Brain Mapping*, 8(4):194–208, 1999.

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

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

[18] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[20] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, August 2009.

[21] David Sussillo and L.F. Abbott. Generating Coherent Patterns of Activity from Chaotic Neural Networks. *Neuron*, 63(4):544–557, August 2009.

[22] Andreea Lazar. SORN: a Self-organizing Recurrent Neural Network. *Frontiers in Computational Neuroscience*, 3, 2009.

[23] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training Recurrent Neural Networks. *arXiv:1211.5063 [cs]*, February 2013. arXiv: 1211.5063.

[24] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[25] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555 [cs]*, December 2014. arXiv: 1412.3555.

[26] György Buzsáki and Andreas Draguhn. Neuronal Oscillations in Cortical Networks. *Science*, 304(5679):1926–1929, June 2004.

[27] Gyorgy Csaba and Wolfgang Porod. Coupled oscillators for computing: A review and perspective. *Applied Physics Reviews*, 7(1):011302, March 2020.

[28] Eve Marder and Dirk Bucher. Central pattern generators and the control of rhythmic movements. *Current Biology*, 11(23):R986–R996, November 2001.

[29] Wolf Singer. Neuronal Synchrony: A Versatile Code for the Definition of Relations? *Neuron*, 24(1):49–65, September 1999.

[30] Michael N. Shadlen and J.Anthony Movshon. Synchrony Unbound. *Neuron*, 24(1):67–77, September 1999.

[31] Supratim Ray and John H. R. Maunsell. Differences in Gamma Frequencies across Visual Cortex Restrict Their Possible Use in Computation. *Neuron*, 67(5):885–896, September 2010.

[32] Supratim Ray and John H.R. Maunsell. Do gamma oscillations play a role in cerebral cortex? *Trends in Cognitive Sciences*, 19(2):78–85, February 2015.

[33] Miguel Romera, Philippe Talatchian, Sumito Tsunegi, Flavio Abreu Araujo, Vincent Cros, Paolo Bortolotti, Juan Trastoy, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Maxence Ernoult, Damir Vodenicarevic, Tifenn Hirtzlin, Nicolas Locatelli, Damien Querlioz, and Julie Grollier. Vowel recognition with four coupled spin-torque nano-oscillators. *Nature*, 563(7730):230–234, November 2018.

[34] T. Konstantin Rusch and Siddhartha Mishra. Coupled Oscillatory Recurrent Neural Network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies. *arXiv:2010.00951 [cs, stat]*, March 2021. arXiv: 2010.00951.

[35] Ben H. Jansen and Vincent G. Rit. Electroencephalogram and visual evoked potential generation in a mathematical model of coupled cortical columns. *Biological Cybernetics*, 73(4):357–366, September 1995.

[36] Jack D. Cowan, Jeremy Neuman, and Wim van Drongelen. Wilson–Cowan Equations for Neocortical Dynamics. *The Journal of Mathematical Neuroscience*, 6(1):1, December 2016.

[37] R J Douglas and K A Martin. A functional microcircuit for cat visual cortex. *The Journal of Physiology*, 440(1):735–769, August 1991.

[38] Georgios Spyropoulos, Matteo Saponati, Jarrod Robert Dowdall, Marieke Louise Schölvinck, Conrado Arturo Bosman, Bruss Lima, Alina Peter, Irene Onorato, Johanna Klon-Lipok, Rasmus Roese, Sergio Neuenschwander, Pascal Fries, and Martin Vinck. Spontaneous variability in gamma dynamics described by a damped harmonic oscillator driven by noise. *Nature Communications*, 13(1):2019, December 2022.

[39] Steven H. Strogatz. From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D: Nonlinear Phenomena*, 143(1-4):1–20, September 2000.

[40] Yun Wang, Henry Markram, Philip H. Goodman, Thomas K. Berger, Junying Ma, and Patricia S. Goldman-Rakic. Heterogeneity in the pyramidal network of the medial prefrontal cortex. *Nature Neuroscience*, 9(4):534–542, April 2006.

[41] John D. Murray, Alberto Bernacchia, David J. Freedman, Ranulfo Romo, Jonathan D. Wallis, Xinying Cai, Camillo Padoa-Schioppa, Tatiana Pasternak, Hyojung Seo, Daeyeol Lee, and Xiao-Jing Wang. A hierarchy of intrinsic timescales across primate cortex. *Nature Neuroscience*, 17(12):1661–1663, December 2014.

[42] Danko Nikolić, Stefan Häusler, Wolf Singer, and Wolfgang Maass. Distributed Fading Memory for Stimulus Properties in the Primary Visual Cortex. *PLoS Biology*, 7(12):e1000260, December 2009.

[43] Colin Blakemore, Jacob Nachmias, and Peter Sutton. The perceived spatial frequency shift: evidence for frequency-selective neurones in the human brain. *The Journal of Physiology*, 210(3):727–750, October 1970.

[44] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154, January 1962.

[45] S. V. David. Natural Stimulus Statistics Alter the Receptive Field Structure of V1 Neurons. *Journal of Neuroscience*, 24(31):6991–7006, August 2004.

[46] Aniruddha Das and Charles D. Gilbert. Topography of contextual modulations mediated by short-range interactions in primary visual cortex. *Nature*, 399(6737):655–661, June 1999.

[47] D Ferster and S Lindström. An intracellular analysis of geniculo-cortical connectivity in area 17 of the cat. *The Journal of Physiology*, 342(1):181–215, September 1983.

[48] Nikola T. Markov, Mária Ercsey-Ravasz, David C. Van Essen, Kenneth Knoblauch, Zoltán Toroczkai, and Henry Kennedy. Cortical High-Density Counterstream Architectures. *Science*, 342(6158):1238406, November 2013.

[49] Sabine Kastner, Peter De Weerd, Mark A. Pinsk, M. Idette Elizondo, Robert Desimone, and Leslie G. Ungerleider. Modulation of Sensory Suppression: Implications for Receptive Field Sizes in the Human Visual Cortex. *Journal of Neurophysiology*, 86(3):1398–1411, September 2001.

[50] Siegrid Löwel and Wolf Singer. Selection of Intrinsic Horizontal Connections in the Visual Cortex by Correlated Neuronal Activity. *Science*, 255(5041):209–212, January 1992.

[51] Yang Yiling, Katharine Shapcott, Alina Peter, Huang Xuhui, Andreea Lazar, and Wolf Singer. Robust Encoding of Natural Stimuli by Neuronal Response Sequences in Monkey Visual Cortex. *Nature Communications*, in revision, 2022.

[52] D.O. Hebb. *The Organization of Behavior*. Psychology Press, 0 edition, 1949.

[53] Andreea Lazar, Christopher Lewis, Pascal Fries, Wolf Singer, and Danko Nikolic. Visual exposure enhances stimulus encoding and persistence in primary cortex. *Proceedings of the National Academy of Sciences*, 118(43):e2105276118, October 2021.

[54] Rosalyn Moran, Dimitris A. Pinotsis, and Karl Friston. Neural masses and fields in dynamic causal modeling. *Frontiers in Computational Neuroscience*, 7, 2013.

[55] Ernst Hairer, Christian Lubich, and Gerhard Wanner. Geometric numerical integration illustrated by the Störmer–Verlet method. *Acta Numerica*, 12:399–450, May 2003.

[56] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. Publisher: Wiley Online Library.

[57] H. Sompolinsky, A. Crisanti, and H. J. Sommers. Chaos in Random Neural Networks. *Physical Review Letters*, 61(3):259–262, July 1988.

[58] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[59] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[60] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926, Anchorage, AK, USA, May 2017. IEEE.

[61] P. Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *ArXiv e-prints*, April 2018. arXiv: 1804.03209 [cs.CL].

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[63] Michael Rosenblum, Arkady Pikovsky, Jurgen Kurths, Carsten Schafer, and Peter A Tass. Phase synchronization: from theory to data analysis. *Handbook of biological physics*, 4:279–321, 2001.

[64] Fritz K. Kneubühl. *Oscillations and Waves*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

Materials and Methods

DHO units. We model each HORN unit as a driven damped harmonic oscillator (DHO) with feedback, defined by the second-order ordinary differential equation (ODE)

$$\ddot{x}(t) + 2\gamma\dot{x}(t) + \omega^2 x(t) = \alpha\sigma\left(I\left(t\right) + F\left(x\left(t\right), \dot{x}\left(t\right)\right)\right), \tag{1}$$

with $x(t)$ denoting the oscillator amplitude, $\gamma > 0$ the damping factor (fig. S1A), $\omega > 0$ the natural angular frequency (fig. S1B), and $\alpha > 0$ an excitability parameter. $\sigma(\cdot)$ denotes an input nonlinearity that

models local resource constraints and is chosen as $\tanh(\cdot)$. $I(t)$ denotes a time-varying external input and $F(x(t), \dot{x}(t)) = vx(t) + w\dot{x}(t)$, $v, w \in \mathbb{R}$ the feedback input (see below).

Depending on the quotient $s = \omega/\gamma$, the solutions of (1) have qualitatively different relaxation dynamics (fig. S1A,B): (i) When $s > 1$, solutions are given by oscillations with decaying amplitude (subcritical damping). (ii) When $s = 1$ or $s < 1$, solutions are given by non-oscillating exponentially decaying curves (critical damping and supercritical damping, respectively).

To define a discrete-time update equation for a DHO unit, we first rewrite (1) as a system of two first order ODEs

$$\begin{aligned} \dot{x}(t) &= y(t) \\ \dot{y}(t) &= \alpha \tanh\left(I(t) + F(x(t), y(t))\right) - 2\gamma y(t) - \omega^2 x(t) \end{aligned} \tag{2}$$

and then discretize (2) using a microscopic time constant h to obtain the update equation for a discrete-time HORN network in which each unit performs an Euler integration of (2) at each time step. As an explicit Euler integration scheme can be numerically unstable when used for the integration of (2), we implemented a symplectic Euler integration scheme [55] for each unit and obtain the update equations

$$\begin{aligned} x_{t+1} &= x_t + h y_{t+1}, \\ y_{t+1} &= y_t + h\left[\alpha \tanh(I_t + F_t(x_t, y_t)) - 2\gamma y_t - \omega^2 x_t\right]. \end{aligned} \tag{3}$$

Here, t denotes a discrete discrete time index $1 \leq t \leq T$, I_t is the external input at time t, and F_t the feedback input of the unit to itself. Initial conditions are chosen as $x_1 = y_1 = 0$ unless otherwise noted. The natural frequency ω corresponds to a period $\omega_p = 2\pi/(\omega h)$ and the memory half-life time is proportional to $1/(\gamma h)$, both measured in input time steps.

The microscopic time constant h defines a scaling factor of input time to system time. It can be interpreted as the sampling frequency of the system, and the linear Nyquist frequency is $h_{\text{Nyquist}} = 1/2h$. Therefore, the theoretical upper limit of the natural (circular) frequency parameter ω for any DHO unit must be $\omega_{\max} < \pi/h$. In practice, this bound is lower as the Euler integration step in (3) becomes numerically unstable (data not shown).

Since the number of integration time steps always equals the length T of the input time series, the numerical integration of the system (3) is invariant under changes in $h \to h'$ when system parameters are rescaled according to

$$\gamma' = c\gamma, \quad \omega' = c\omega, \quad \alpha' = c^2\alpha, \tag{4}$$

where $c = h/h'$. To allow a better interpretability of the natural frequency parameter, we set $h = 1$ without loss of generality to link the two time scales and express the natural frequency ω in period units of the input as $\omega_p = 2\pi/\omega$ when convenient. A value of $h = 1$ will be assumed throughout the manuscript if not stated otherwise.

HORN Networks. A HORN network is a system of n additively coupled DHOs of the form (3). Given a time-varying external input $S_{\text{ext}} = (\mathbf{s}_1, \ldots, \mathbf{s}_T)$, $\mathbf{s}_t \in \mathbb{R}^m$, the update equations of a HORN are

$$\begin{aligned} \mathbf{x}_{t+1} &= \mathbf{x}_t + h\mathbf{y}_{t+1} \\ \mathbf{y}_{t+1} &= \mathbf{y}_t + h\left[\boldsymbol{\alpha} \cdot \tanh\left(1/\sqrt{n}\mathbf{I}_t^{\text{rec}} + \mathbf{I}_t^{\text{ext}}\right) - 2\boldsymbol{\gamma} \cdot \mathbf{y}_t - \boldsymbol{\omega}^2 \cdot \mathbf{x}_t\right], \end{aligned} \tag{5}$$

where I_t^{rec} and I_t^{ext} denote the recurrent and external input to each unit at time t, respectively, defined as

$$\begin{aligned} \mathbf{I}_t^{\text{rec}} &= \mathbf{W}^{hh}\mathbf{y}_t + \mathbf{b}^{hh} + \mathbf{v} \cdot \mathbf{x}_t, \\ \mathbf{I}_t^{\text{ext}} &= \mathbf{W}^{ih}\mathbf{s}_t + \mathbf{b}^{ih}. \end{aligned}$$

Above, lowercase boldface symbols denote vectors of length n, uppercase boldface symbols denote matrices, $\cdot$ denotes element-wise multiplication, $\mathbf{W}^{hh}$ is the $n \times n$ matrix of recurrent coupling weights, $\mathbf{W}^{ih}$

is the $n \times m$ matrix of input weights, and $\mathbf{b}^{hh}$, $\mathbf{b}^{ih}$ are bias vectors. As for the single-unit case, the update equations implement a symplectic Euler integration scheme. The feedback input from a unit to itself $F(x_t, y_t)$ is $F(x_t, y_t) = vx_t + wy_t$ with $w = \mathbf{W}^{hh}_{ii}$. Note that the values of the parameters ω, γ, α can vary between DHO units. To train a HORN on a task, an affine readout of the network state $\mathbf{x}_{t_0}$ at time t_0 is defined as a map $\mathbf{R}(\mathbf{x}_{t_0}) = \mathbf{W}^{ho}\mathbf{x}_{t_0} + \mathbf{b}^{ho}$, where $\mathbf{W}^{ho}$ is the $k \times n$ readout matrix and $\mathbf{b}^{ho}$ is a bias vector. As previously, we will set $h = 1$ to link the input and system time scales, unless otherwise stated. See below for a description of network simulations.

Other RNN Architectures. A simple Elman RNN [56] with a linear readout (referenced by the label "tanh" throughout all plots) was implemented by the update equation

$$\mathbf{x}_{k+1} = \tanh\Big(\mathbf{W}^{hh}\mathbf{x}_k + \mathbf{b}^{hh} + \mathbf{I}_k\Big),$$

where the input $\mathbf{I}_k$ is defined as above. Similarly, a "leak" RNN consisting of leaky integrators was implemented by the update equation

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h\left[-a\mathbf{x}_k + \tanh\Big(\mathbf{W}^{hh}\mathbf{x}_k + \mathbf{b}^{hh} + \mathbf{I}_k\Big)\right],$$

where a is a leak constant that controls memory decay, see also [57] in which $a = 1, h = 1$. For the LSTM and GRU based RNNs, the implementations of LSTM and GRU units provided by PyTorch (version 1.9.0) [58] were used, see below.

Number of Trainable Parameters. Performance comparisons for different RNN architectures were carried out at the same number of trainable parameters. Assuming an RNN with n_h nodes receiving input of dimensionality n_i through an affine projection layer, and a readout to an n_o-dimensional output vector via an affine readout layer, the number of trainable parameters is $N = c(n_i + b_i)n_h + cn_h(n_h + b_h) + (n_h + b_o)n_o$, where c is the number of state variables per unit ($c = 3$ for GRU, $c = 4$ for LSTM, $c = 1$ for the simple and leak RNNs) and b_i, b_h, b_o is either 1 (bias term present) or 0 (no bias term present). HORNs also have one state variable per unit (i.e. $c = 1$), but an additional amplitude self-connection term for each unit, leading to $N_{\text{HORN}} = (n_i + b_i)n_h + n_h(n_h + 1 + b_h) + (n_h + b_o)n_o$. To construct a network of a given architecture with a given number N^* of trainable parameters, we solve the quadratic equation $N = N^*$ after plugging in the known values for n_i, n_o, b_i, b_h, b_o and take the integer part of the positive solution to obtain the largest number n^* of nodes so that the number of trainable parameters does not exceed N^*. See fig. S3 for the scaling behavior of trainable parameters as a function of the number system units for different RNN architectures for $n_i = 1$, $n_o = 10$, $b_i = b_h = b_o = 1$.

Data Sets. The **MNIST** data set [59] of hand written digits consists of a training data set of 60,000 samples and a test set of 10,000 samples which was used to assess classification performance (measured by classification accuracy) of different networks. Each sample is given by a 28x28 matrix of intensity values between $[0, 255]$ which are scaled to $[0, 1]$. Two variants of the MNIST data set are considered here, the sequential MNIST (sMNIST), and the permuted sequential MNIST (pMNIST). For sMNIST, the 28x28 intensity values of a sample are collected in scanline order from top left to bottom right to form a time series of length 784 that is then fed to the networks step by step. For pMNIST, a random but fixed permutation on 784 elements is applied to the stimulus before feeding the input to the network. This removes the dominant low frequency structure present in sMNIST digits and distributes stimulus information across all frequencies (figs. S5, S4). Noise was added to sMNIST or pMNIST time series in form of i.i.d. additive white Gaussian noise on a per-pixel level and sampled from $N(0, \sigma^2)$. Stimulus values were clamped to $[0, 1]$ after noise was added.

The **Line segments data set** (LSDS) consists of samples of 28x28 pixel intensity values with one or more line segments of a given angle. The data set has four parameters that define the nature of its samples: The number of angles n_a, the maximal number of segments per sample n_s, the minimal and maximal length of the line segments l_{min} and l_{max}. It furthermore comes in two flavors: with random segment locations or centered segment locations. To construct one sample, first each entry of a 28×28

pixel canvas is set to 0 and an angle $\alpha \in \{\alpha_1, \ldots, \alpha_{n_a}\}$ is chosen at random with $\alpha_i = (i-1)\pi/n_a$, i.e. each sample only shows line segments of the same angle. Then the number of line segments k to be shown is chosen as a random integer in $1, \ldots, n_s$. For each line segment i, first a random integer length l_i is drawn from $l_{\min}, \ldots, l_{\max}$. Then a line segment of length l_i pixels is drawn on the 28×28 canvas with intensity 1.0 (using the Python package `opencv-python`, version 4.6). If line segments are to be centered, they are drawn so that their central point coincides with the center of the 28×28 canvas. If segment locations are random, a random starting coordinate from the set $\{0, \ldots, 21\}$ is chosen for both the horizontal and vertical starting point of the line segment. The label for each sample is the angle α present in the sample and the networks are trained on a n_a-class classification problem of predicting the angle shown in a given sample. We will write LSDS(n_a, n_s, $l_{\min}$, $l_{\max}$, L) with $L \in \mathrm{r}, \mathrm{c}$ to refer to the data set with random ($L = \mathrm{r}$) and centered ($L = \mathrm{c}$) segment locations. We use two versions of the LSDS in the study, LSDS(32,5,11,c) which we will refer to as LSDSa, and LSDS(10,3,8,24,r) which we will refer to as LSDSb. Samples of line segment data sets are shown in fig. S15.

The **EMNIST** (balanced) data set [60] consists of hand-written 131,600 characters belonging to 47 classes, representing hand written digits as well as uppercase and lowercase latin characters. Each character is represented by a matrix of 28x28 pixel intensity values as for MNIST. The training data set is of size 112,800, the test set of size 18,800. EMNIST samples have the same format as MNIST, and their signal properties as time series are similar to those of MNIST digits.

The of **Spoken Digits Data Set** (SDDS), was chosen as a subset of the Speech Commands data set (version 0.0.2) [61] (SCDS). The SCDS consists of 65,000 audio audio recordings (1 second length, mono, 16kHz sampling rate) of spoken words and digits obtained from thousands of people and spans 30 different classes. The SDDS data set was formed by choosing the 10 classes from the SCDS representing spoken digits 0 to 9, forming a data set of 23,666 samples (with each class consisting of around 2730 samples). Of the SDDS, a random selection of 10% of all samples were used as test set, and the rest was used as the training set. To construct the input time series to networks, each spoken digit was first sub-sampled to 4000 Hz using the Python library `librosa`, followed by a normalization of the amplitude to the interval $[-1, 1]$, performed per sample.

The power spectral densities of the time series calculated on samples from each data set is shown in fig. S4.

Simulations. All simulations were performed in PyTorch (version 1.9.0) [58]. All networks consisted of three trainable affine layers (input, recurrent, readout). For the Elman, leak and HORN networks, custom implementations were used and the code is available at `https://github.com/....` For the LSTM and GRU networks, the standard implementations provided by PyTorch were used.

The initial weights and biases of all affine layers were randomly drawn from a uniform distribution $U(-1/\sqrt{n_{\text{in}}}, 1/\sqrt{n_{\text{in}}})$, where n_{in} denotes the input dimension of the layer (default implementation in PyTorch `torch.nn.Linear`).

All parameters were learned with the BPTT algorithm. As loss function for the studied stimulus classification problems we used the cross entropy loss function (`torch.nn.CrossEntropyLoss`). Weight updates were performed according to the AdamW optimizer with default settings (`betas=(0.9,0.999)`, `eps=1e-8`, `weight_decay=0.01`) using a batch size of 128 for all data sets, unless otherwise stated. For the SDDS data set, we used a batch size of 64.

For each model, a grid search on hyperparameters was performed using a network with 10^4 trainable parameters to find the parameter values that resulted in a network with the highest classification accuracy on the test set after 10 training epochs. The parameters contained in the grid search were different per model and are described below. For all models, the grid search included the learning rate $\texttt{lr} \in \{10^{-5}, 5 \cdot 10^{-5}, 10^{-4}, 10^{-3}, 5 \cdot 10^{-3}, 10^{-2}\}$. Optimal learning rates for HORNs and the leak RNN were found to be 10^{-2} except for SDDS where they were $5 \cdot 10^{-3}$. Optimal learning rates for the gated (LSTM, GRU) RNNs were determined to be 10^{-3} for all data sets (SDDS: 10^{-4}). Learning rates for the Elman RNN were set to 10^{-4} for all data sets (SDDS: 10^{-5}). Gradient clipping (`pytorch.nn.utils.clip_grad_norm_`) was applied in the case of the Elman, GRU, and LSTM networks, as without gradient

clipping these architectures showed unstable learning trajectories or failed to depart from chance-level classification performance for sMNIST. For these networks, the grid search included the learning rate and clipping values $c \in \{0, 0.1, 1, 5, 10, 100\}$. A clipping value of 1 was chosen across all data sets. For the leak RNNs, the grid search included the learning rate, the time constant $h \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ and the leak constant $a \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.75, 1.0\}$. See Table S1 for the parameter values for each data set used. For homogeneous HORNs, the grid search included the learning rate, the time constant $h \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$, the natural frequency $\omega \in \{0.1, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0, 10.0\}$, and the damping factor $\gamma \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.25, 0.5\}$, setting $\alpha = 1.0$. For heterogeneous HORNs, as a first step the spread of values h, ω, γ among the best 10 performing homogeneous HORNs from the previous step were calculated as $[h_{\min}, h_{\max}]$, $[\omega_{\min}, \omega_{\max}]$, $[\gamma_{\min}, \gamma_{\max}]$, respectively. Subsequently, each node of a HORN^n was randomly assigned parameter values h_i, ω_i, γ_i sampled in an i.i.d. manner from $U([h_{\min}, h_{\max}))$, $U([\omega_{\min}, \omega_{\max}))$, $U([\gamma_{\min}, \gamma_{\max})$, respectively, where $U([a, b))$ denotes the uniform distribution over the half-open interval $[a, b)$.

See Table S.1 for dataset-dependent parameters for the HORN and leak networks. For HORN networks, normalized parameters with $h = 1$ are given. For the implementation details of HORN networks with synaptic delays and multi-layer networks see below.

Gradient Stability. Given a loss function L, the gradients of L with respect to a recurrent weight w can be computed as [23]

$$\frac{\partial L}{\partial w} = \sum_{1 \le k \le T} \frac{\partial L_k}{\partial w} = \sum_{1 \le k \le T} \sum_{1 \le l \le k} \frac{\partial L_k}{\partial H_k} \frac{\partial H_k}{\partial H_l} \frac{\partial H_l^+}{\partial w}, \tag{6}$$

where H_t is hidden state of the network at the time t, T is depth of recurrent network (i.e. the number of input time steps), $\frac{\partial H_l^+}{\partial w}$ is the immediate derivative and where

$$\frac{\partial H_k}{\partial H_l} = \prod_{k \le t < l} \frac{\partial H_{t+1}}{\partial H_t}, \tag{7}$$

by iterated application of the chain rule. The propagation of the gradient at time t is defined by the Jacobian's behavior $\frac{\partial H_{t+1}}{\partial H_t}$. The vanishing gradient problem is characterized by $\left\| \prod_{k \le t < l} \frac{\partial H_{t+1}}{\partial H_t} \right\| \ll 1$, and $\left\| \prod_{k \le t < l} \frac{\partial H_{t+1}}{\partial H_t} \right\| \gg 1$ is called exploding gradient problem.

To study the dynamics of gradients in HORNN networks, we first make the following substitutions in the expression (5):

$$\mathbf{H}_t = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{y}_t \end{bmatrix}, \tag{8}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} - h^2 \operatorname{diag} \boldsymbol{\omega}^2 & (h - 2h^2) \operatorname{diag}(\boldsymbol{\gamma}) \\ -h \operatorname{diag}(\boldsymbol{\omega}^2) & \mathbf{I} - 2h \operatorname{diag}(\boldsymbol{\gamma}) \end{bmatrix}, \tag{9}$$

and denote by

$$\phi(\mathbf{H}_t, \mathbf{s}_t) = \begin{bmatrix} h \tanh(\mathbf{I}_t^{\text{rec}}(\mathbf{H}_t) + \mathbf{I}_t^{\text{ext}}(\mathbf{s}_t)) \\ \tanh(\mathbf{I}_t^{\text{rec}}(\mathbf{H}_t) + \mathbf{I}_t^{\text{ext}}(\mathbf{s}_t)) \end{bmatrix}, \tag{10}$$

the transformation defined by the tanh non-linearity, where $\mathbf{s}_t, \mathbf{I}_t^{\text{rec}}, \mathbf{I}_t^{\text{ext}}$ are as in (5), $\mathbf{I}$ denotes the identity matrix, and $\operatorname{diag}(\mathbf{x})$ the diagonal matrix with diagonal entries from the vector $\mathbf{x}$.

These substitutions allow us to write the recurrent update step (5) as

$$\mathbf{H}_{t+1} = \mathbf{A}\mathbf{H}_t + h\boldsymbol{\alpha}\phi(\mathbf{H}_t, \mathbf{s}_t). \tag{11}$$

The sufficient condition to avoid both vanishing and exploding gradients is $\frac{\partial \mathbf{H}_{t+1}}{\partial \mathbf{H}_t} \approx \mathbf{I}$. This condition can be achieved in (11) by choosing an Euler discretization time step $h \ll 1$. The scaling relationships for the system parameters ω, γ, α that result in equivalent system behavior under changes of h is described in (4). Similar arguments hold for the RNN composed of leaky integrators, explaining their

increased performance compared to the simple Elman RNN. In essence, the Euler integration scheme introduces residual connections between subsequent steps of HORNN dynamics, which ensure stable gradient propagation [62].

Phase Locking Value. To analyze the pairwise synchronization between DHO units, we used the analytical representation of the amplitude signal $x(t)$ which can be computed by means of the Hilbert transform as $X(t) = x(t) + iH(x(t))$, where $H(\cdot)$ is the Hilbert transform and i is the imaginary unit. Then, the instantaneous phase $\phi(t)$ and the instantaneous magnitude $A(t)$ can be computed as $S(t) = A(t)\exp(i\phi(t))$. As a measure of nonlinear interactions, we use the pairwise phase locking value (PLV) computed over the time window T, where T denotes the input signal length [63]. The PLV for two signals x_1 and x_2 in analytical representation is defined as

$$\mathrm{PLV}(x_1, x_2) = \frac{1}{T}\left|\sum_{t=1}^{T} e^{i\Delta\phi(t)}\right|, \tag{12}$$

where $\Delta\phi(t) = \phi_1(t) - \phi_2(t)$.

The phase locking value PLV takes values on the real interval $[0, 1]$. The case of $\mathrm{PLV}(x_1, x_2) = 1$ indicates fully phase-locked synchronization of x_1 and x_2, while a value of $\mathrm{PLV}(x_1, x_2) = 0$ indicates fully asynchronous signals x_1, x_2.

PCA dimensionality of network activity. To measure PCA dimensionality, network activity vectors were collected using 1000 samples from the sMNIST test dataset. The activity vectors for all 784 time steps of stimulus presentation were treated as independent samples, resulting in $1000 \cdot 784$ samples. PCA was performed on the resulting dataset (Python package `scikit-learn`, `sklearn.decomposition.PCA`). As a result, every principal component constitutes the activation vector which explains the amount of variance as defined by the corresponding eigenvalue of the covariance matrix. The number of PCA components explaining 99% variance of the network's activity was taken as measure of PCA dimensionality. PCA dimensionality measures the linear dimensionality and takes integer values on the interval $[1, N]$, where N is the number of units in the network. To show the dynamics in PCA dimensionality over learning, it was computed for a sequence of intermediate instances over training for each network.

Decoding of HORN activity. To decode the stimulus based on RNN activity, activity vectors were collected at each time step t of the stimulus presentation phase ($t = 1, \ldots, 784$ for sMNIST), using 10000 samples from the test dataset. For each time step t, the resulting set of activity vectors was randomly split into a training set and a test set using a 80%/20% ratio. Using this dataset, an SVM classifier was trained on the 10-class stimulus classification problem for each time point t (Python package `scikit-learn`, `sklearn.svm.LinearSVC`). A regularization constant of $C = 0.1$ was chosen to ensure the convergence of all SVM classifiers. The confusion matrices were calculated using `sklearn.metrics.confusion_matrix`. Distance to the SVM decision hyperplane was calculated with `sklearn.svm.LinearSVC.decision_function`.

Receptive Fields. For each DHO unit, the intrinsic receptive field (iRF) was calculated as the gain curve of the unit [64]. The gain curve is computed using harmonic (sinusoidal) inputs. It is defined as the quotient between the DHO unit amplitude, i.e. the stationary forced oscillation, and the input amplitude as a function of input frequency ω_i. Due to the clipping induced by the tanh nonlinearity, these curves vary depending on the amplitude chosen of the input signal (Section **??** and fig. S30). For the intrinsic receptive field, these gain curves were calculated for an input amplitude of $A = 1$ over the time of $T = 50000$ time steps to allow the transient dynamics to settle (fig. S14).

The stimulus-dependent effective receptive field (eRF) of each DHO unit in a HORN network was calculated as the mean most strongly driving stimulus (as the mean absolute amplitude computed over the entire presentation time). For this, network activity was first recorded for $n = 10000$ samples in a test set, and then the average of the 500 stimuli that resulted in the highest activation of a given DHO unit was calculated to form the effective receptive field of a DHO unit.

Networks with Synaptic Delays. To implement synaptic conduction delays in a network of n nodes, we first defined a maximum delay $d_{\max}$ present in a network (measured in the number of discrete time

steps) and subsequently assigned to each connection from a unit j to a unit i a delay of d_{ij} time steps. We then modified the recurrent input $\mathbf{I}_t^{\text{rec}}$ of the HORN update equation (5) to

$$\mathbf{I}_t^{\text{rec}} = \sum_{d=1}^{d_{\max}} \mathbf{M}^d \mathbf{W}^{hh} \mathbf{y}_{t+1-d} + \mathbf{b}^{hh} + \mathbf{v} \cdot \mathbf{x}_t, \tag{13}$$

where $\mathbf{M}^d \in \{0,1\}^{n \times n}$, $1 \leq d \leq d_{\max}$ are binary masks and the value of $\mathbf{M}_{ij}^d = 1$ iff the connection from unit j to unit i has a delay of d, and $\mathbf{M}_{ij}^d = 0$ otherwise. Furthermore, we assume $\mathbf{y}_i = 0$ for all $i < 0$. This approach allows us to continue using the dense gradient methods implemented in PyTorch while effectively performing operations on $d_{\max}$ sparse matrices in each time step.

Multi-Layer Networks. To implement a two-layer network with per-layer connectivity, we first choose the number of units per layer as n_1 and n_2. Furthermore, we introduce connection probability parameters p_1, p_2, p_{F}, p_{B} where p_1 and p_2 model the recurrent connection probabilities within the layer 1 and 2, respectively, p_{F} is the feedforward connection probability from layer 1 to layer 2, and p_{B} is the feedback connection probability from layer 2 to layer 1. We set $p_1 = p_2 = 1$ unless otherwise noted, i.e. we choose fully connected layers. We then model the two-layer network by simulating a HORN network of $n = n_1 + n_2$ nodes, where the first n_1 nodes represent layer 1, and the rest of the nodes layer 2. To model layer-specific connectivity patterns, we modify the HORN update equation (5) using masks as above and write

$$\mathbf{I}_t^{\text{rec}} = \mathbf{M}^{hh} \mathbf{W}^{hh} \mathbf{y}_t + \mathbf{b}^{hh} + \mathbf{v} \cdot \mathbf{x}_t, \tag{14}$$

where the masking matrix $\mathbf{M}^{hh} \in \{0,1\}^{n \times n}$ has a block structure and defines the connections present in the network. The entries of $\mathbf{M}^{hh}$ are sampled from a binomial distributions $B(p)$ in an i.i.d. manner as follows: $\mathbf{M}_{ij}^{hh} \sim B(p_1)$ for $1 \leq i, j \leq n_1$ (recurrent connections of layer 1), $\mathbf{M}_{ij}^{hh} \sim B(p_{\text{F}})$ for $n_1 + 1 \leq i \leq n_1 + n_2$, $1 \leq j \leq n_1$ (feedforward connections from layer 1 to layer 2), $\mathbf{M}_{ij}^{hh} \sim B(p_{\text{F}})$ for $1 \leq i \leq n_1$, $n_1 + 1 \leq j \leq n_1 + n_2$ (feedback connections from layer 2 to layer 1), $\mathbf{M}_{ij}^{hh} \sim B(p_2)$ for $n_1 + 1 \leq i, j \leq n_1 + n_2$ (recurrent connections of layer 2). Note that in a two-layer network the input is only provided to the first layer and activity is only read out from the second layer. To implement this, we masked both the input layer and the readout layer using binary masks as above to only project input to the first layer and to only read out activity from the second layer, respectively.

To simulate the presence and absence of different types of recurrent connections within a single layer HORN network of n units, we use the update equation (14) and set the mask matrix $\mathbf{M}^{hh}$ as follows: (i) $\mathbf{M}^{hh} = 0$ for a network consisting of isolated units without self-connections, (ii) $\mathbf{M}_{ii}^{hh} = 1$ for $1 \leq i \leq n$ and $\mathbf{M}_{ij}^{hh} = 0$ otherwise for a network of isolated units with self-connections, (iii) $\mathbf{M}_{ii}^{hh} = 0$ for $1 \leq i \leq n$ and $\mathbf{M}_{ij}^{hh} = 1$ otherwise, resulting in a fully connected network of units without self-connections, and (iv) $\mathbf{M}_{ij}^{hh} = 1$ for all i, j for a fully connected network of units with self-connections. We note that case (iv) is equivalent to that of an unmasked network with update equation (5).

Hebbian Learning Rule. To simulate Hebbian weight updates of the recurrent connection matrix $\mathbf{W}^{hh}$, the weights of the input layer $\mathbf{W}^{ih}$ and the read-out layer $\mathbf{W}^{ho}$ were kept plastic for the BPTT algorithm, but the recurrent weights $\mathbf{W}^{hh}$ were frozen. Instead, a simple additive rule was implemented that modifies the weight of a synaptic connection $\mathbf{W}_{ij}$ from unit j to unit i by an additive term

$$\Delta \mathbf{W}_{ij} = \sigma_\Delta \lambda_h a_{ij} r(x_i(t), x_j(t)), \tag{15}$$

where $\sigma_\Delta \in \{+1, -1\}$ determines the type of the learning rule ($\sigma_\Delta = +1$: Hebb+, $\sigma_\Delta = -1$: Hebb-), λ_h denotes the learning rate, $a_{ij} \in \{0,1\}$ an activity modulator described below, and $r(x_i(t), x_j(t))$ denotes the Pearson correlation between the activity vectors of the units i and j computed over the time steps $t = c, \ldots, T$, respectively, where $T = 784$ denotes the stimulus length. The parameter c controls the starting point in time after which the correlation coefficients are computed. During training, the update rule (15) was applied to the entries of $\mathbf{W}_{ij}$ according to the following logic. For each sample,

the distribution of pairwise cross-correlation coefficients $r(x_i(t), x_j(t))$ of the evoked HORN activity was calculated for all unit pairs. Subsequently, the upper and lower P percentiles of this distribution were determined to identify the most strongly correlated and most strongly anticorrelated units, excluding self-connections i, i (the active connections). The activity modulator a_{ij} was set to 0 for all unit pairs i, j for which $r(x_i(t), x_j(t))$ did not belong to the set of active connections (the inactive connections). For active connections, the values of a_{ij} were sampled from a binomial distribution $B(p_s)$, where p_s is the selection probability. The simulation was performed using the pMNIST data set, see Table S1 for the base HORN parameters. For all HORNS we disabled the self-connections in the amplitude as well as velocity terms as these cannot be meaningfully updated with the correlation-based rules, and also disabled bias vectors for the recurrent connections for the same reason. A grid search on $\lambda_h \in \{5 \cdot 10^{-5}, 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}, 5 \cdot 10^{-3}\}$, $c \in \{0, 300, 600\}$, $P \in \{1, 5, 10, 50\}$, $p_s \in \{0.1, 0.5, 1.0\}$ was performed to determine optimal parameters for each case. For the homogeneous HORN, the following parameters were used for the correlation-based learning rules: Hebb+: $\lambda_h = 10^{-4}$, $c = 300$, $P = 5$, $p_s = 0.5$. Hebb-: $\lambda_h = 5 \cdot 10^{-4}$, $c = 300$, $P = 5$, $p_s = 1.0$. For the heterogeneous HORN, the following parameters were used: Hebb+: $\lambda_h = 10^{-4}$, $c = 300$, $P = 5$, $p_s = 0.1$. Hebb-: $\lambda_h = 10^{-4}$, $c = 600$, $P = 5$, $p_s = 0.1$.

Geometric Input. Geometric input was provided to a network by defining non-overlapping receptive fields of size 2×2 pixels and covering the 28×28 intensity matrix of each MNIST digit by 14×14 receptive fields ordered from top left to bottom right of the stimulus matrix in a scanline fashion. The pixel intensities in each of the 196 receptive fields were mapped to 196 DHO units that form a fully connected HORN^h with unit input connection weights and no input bias in a way that preserves geometric relations between the input receptive fields (so that the first DHO unit receives input from the top left receptive field, and the last DHO unit from the bottom right receptive field). To simulate a flashed stimulus, the network only received input during the first time step. Following the stimulus presentation period, we let the network dynamics unfold for 150 time steps. The whole network was trained using BPTT on the MNIST digit classification task for a readout at time $T = 150$. We trained a HORN^h network on both the original and a shuffled version of the MNIST digits (with shuffling performed by applying a random but fixed permutation of the 28×28 pixel values of each digit).

Supplementary Text

1 Relation to the Wilson-Cowan model

The Wilson-Cowan (WC) equations [1] are a mean-field model that describes the dynamics of two recurrently coupled populations of neurons, an excitatory (E) and an inhibitory (I) one. Denoting by $E(t)$ and $I(t)$ the instantaneous firing rate of the excitatory and inhibitory population, respectively, the Wilson-Cowan equations are

$$\begin{aligned} \tau\dot{E}(t) &= -E(t) + (1 - rE(t))f_E\left[w_{EE}E(t) - w_{EI}I(t) + h_E(t)\right] \\ \tau\dot{I}(t) &= -I(t) + (1 - rI(t))f_I\left[w_{IE}E - w_{II}I + h_I(t)\right], \end{aligned} \tag{S1}$$

where τ is the characteristic time scale of the system, w_{XY} is the coupling strength of the subpopulation X to Y (with $X, Y \in \{E, I\}$), h_X is the external input of the subpopulation X, and f_X is a non-linear (usually sigmoidal) activation function of the subpopulation X. Here, we use a slightly modified version of the Wilson-Cowan model, as in [2], which is known to express oscillations. Here, the activation function f_X is given by

$$f_X = \frac{1}{1 + \exp(-a_X(X - \theta_X))}, \tag{S2}$$

where θ_X define the center and a_X the steepness of the nonlinearity for the population $X = E, I$, respectively.

Following [2], we take the external inhibitory and excitatory inputs h_I and h_E as free parameters and fix the values of the other parameters as $w_{EE} = 13, w_{EI} = 6, w_{IE} = 12, w_{II} = 3, \tau = 1, a_e = 1.3, a_i = 2, \theta_e = 4, \theta_i = 1.5$ for our analysis.

In the h_E-h_I subspace, we obtain a bifurcation diagram (fig. S19) that partitions the state space of the system into regions with three dynamically distinct behaviors: (i) asymptotically decaying dynamics, (ii) oscillatory decaying dynamics, and (iii) dynamics expressing sustained oscillations (or limit cycles). The system trajectories in each subspace behave qualitatively in the same way as the ones produced by a damped harmonic oscillator subject to different parameter and input configurations. The dynamics in region (i) are equivalent to the ones produced by a supercritically damped DHO, the ones of region (ii) correspond to a subcritically damped DHO, and the dynamics in region (iii) correspond to the case of a driven subcritically damped DHO (or a unit with velocity feedback; see Section 2.1).

The WC model allows for a rich repertoire of qualitatively different dynamics. However, it is difficult to tune its oscillatory properties, and in particular the frequencies of the oscillations (fig. S19), given that the model has 11 parameters. DHOs on the other hand have two easily interpretable parameters ω and γ, describing their oscillation frequency and decay behavior, respectively (fig. S1 A and B). Despite its simplicity, the DHO model (fig. S1) is able to qualitatively reproduce the different dynamics expressed by the WC model for the parameter values considered above (fig. S19).

2 Dynamics of DHO units

In our networks, one DHO unit represents a mesoscale region of the brain such as a (P)ING circuit or a cortical column, consisting of both an excitatory and an inhibitory population of neurons. Local recurrent connectivity within each mesoscale region results in feedback input of the circuit to itself (see Section 1). To model this feedback activity, we introduce two forms of feedback-connectivity for each DHO unit: a velocity and an amplitude feedback connection. Broadly speaking, the amplitude feedback connection allows the unit to adjust its preferred oscillation frequency, and the velocity feedback connection allows it to alter the damping behavior of its relaxation dynamics.

In the following, we will analytically study the effect of each of the feedback connections on unit dynamics and give an overview of how the interaction of the two feedback terms influences a unit's state space and relaxation dynamics.

2.1 Velocity-feedback DHO

In a network setting, the self-connectivity term in the velocity is given by the diagonal of the weight matrix $\mathbf{W}^{hh}$, which motivates the name $w = W_{ii}$ for the self-connection weight (see Equation 5 in Methods).

Here, we consider a single DHO unit with a velocity self-connection term w defined by

$$\ddot{x}(t) + 2\gamma\dot{x}(t) + \omega^2 x = \alpha \tanh(wy(t) + I(t)). \tag{S3}$$

As before, this ODE can then be rewritten as a two-dimensional system

$$\begin{aligned} f(x,y) &= \dot{x} = y \\ g(x,y) &= \dot{y} = -2\gamma y - \omega^2 x + \alpha \tanh(wy). \end{aligned} \tag{S4}$$

The eigenvalues of the Jacobian matrix are given by

$$\lambda_{1,2} = \frac{\alpha w - 2\gamma \pm \sqrt{(\alpha w - 2\gamma)^2 - 4\omega^2}}{2}. \tag{S5}$$

They cross the imaginary axis at the value

$$w_c = \frac{2\gamma}{\alpha}, \tag{S6}$$

which characterizes a Hopf bifurcation (fig. S20) [3].

In terms of system dynamics, this means that the unit starts expressing sustained oscillations if the velocity self-connection term w is greater than w_c.

2.2 Amplitude-feedback DHO

To model amplitude feedback, we introduce a second self-coupling term v (fig. S1E).

$$\ddot{x}(t) + 2\gamma\dot{x}(t) + \omega^2 x = \alpha \tanh(vx(t) + I(t)), \tag{S7}$$

where $I(t)$ denotes the external input to the unit at time t, which we will initially set to $I(t) = 0$.

We then rewrite the above second-order differential equation (S7) as a system of first order ODEs:

$$\begin{aligned} f(x,y) &= \dot{x} = y \\ g(x,y) &= \dot{y} = -2\gamma y - \omega^2 x + \alpha \tanh(vx). \end{aligned} \tag{S8}$$

We calculate the nullclines as

$$\dot{x} = 0 \Rightarrow y = 0, \tag{S9}$$

$$\dot{y} = 0 \Rightarrow y = \frac{\alpha \tanh(vx) - \omega^2 x}{2\gamma}. \tag{S10}$$

The intersections of the nullclines (fig. S21) defines the fixed points of the system that we denote by (x^*, y^*). As all fixed points of the system satisfy $y^* = 0$ (Equation S9), we will refer to the fixed point $(x^*, 0)$ as x^* from now on, unless otherwise stated.

Combining (S9) and (S10), we obtain the following transcendental equation

$$\tanh vx^* = \frac{\omega^2}{\alpha} x^*, \tag{S11}$$

which cannot be solved analytically. To obtain solutions, we compute the intersection between the nullclines numerically (fig. S21). This allows us to study the behavior of the fixed points of the system x^* for different values of v (fig. S22).

The bifurcation diagram (fig. S22) shows that the units undergo a pitchfork bifurcation at the critical point $v_c = \frac{\omega^2}{\alpha}$. This means that our units have different stationary behaviors for different intervals of the values of v: (i) Subcritical case: The unit has a single stable solution at (0,0) when $v \leq \frac{\omega^2}{\alpha}$. (ii) Supercritical case: Once we cross the bifurcation point $v = v_c$, the origin loses stability and two new stable solutions arise, given by (S11). In the latter case, the stationary solution depends on the initial conditions, see fig. S23.

Although the solutions of (S11) cannot be analytically obtained, we can calculate the bounds to the solutions. Consider $x' = x^* \frac{\omega^2}{\alpha}$. We can then rewrite (S11) as

$$\tanh v \frac{\alpha}{\omega^2} x' = x'. \tag{S12}$$

When $v \to \infty$ we have $x' \to \pm 1$, $x^* = \pm \frac{\alpha}{\omega^2}$ as the asymptotic values for the fixed points (fig. S21 black dots, fig. S22 dashed lines).

We now consider model dynamics in the driven regime, i.e. when $I(t) \neq 0$, defined by the ODE

$$\ddot{x} + 2\gamma\dot{x} + \omega^2 x = \tanh\left(vx + I\left(t\right)\right). \tag{S13}$$

In the following, we will analyze the dynamics of a single unit as defined above for different types of input.

Constant Input. First, we consider the case of constant input, $I(t) = b$. This results in a slightly different transcendental equation for the fixed points compared to the undriven case:

$$\alpha \tanh\left(vx^* + b\right) = \omega^2 x^*. \tag{S14}$$

Solving for v, we obtain

$$v = \frac{\tanh^{-1}\left(\frac{\omega^2}{\alpha} x^*\right) - b}{x^*}. \tag{S15}$$

The new parameter b breaks the symmetry of (S15), see fig. S24, top. Given a fixed $b \neq 0$, the system undergoes a saddle-node bifurcation as v passes x_m, the local minimum of (S15). We can calculate x_m, the local minimum and x_i, the inflection point, by setting $\partial v / \partial x^* = 0$ and obtain

$$b + \frac{\frac{\omega^2}{\alpha} x_m}{1 - \left(\frac{\omega^2}{\alpha} x_m\right)^2} - \tanh^{-1}\left(\frac{\omega^2}{\alpha} x_m\right) = 0 \tag{S16}$$

and

$$b - \frac{\frac{\omega^2}{\alpha} x_i}{1 - \left(\frac{\omega^2}{\alpha} x_i\right)^2} + \tanh^{-1}\left(\frac{\omega^2}{\alpha} x_i\right) = 0. \tag{S17}$$

as the two solutions.

This symmetry breaking creates two branches on which the fixed points of the system lie. One branch, which is stable, persists for all values of b, and the other branch, as mentioned above, is a saddle. To obtain the bifurcation point, we sequentially solve the equations (S16) and (S15) numerically. Equation (S16) lets us compute the local minimum $x^* = x_m$, which we insert into equation (S15) to obtain $v_m = v\left(x_m\right)$. The saddle-node bifurcation point is then defined by the tuple (x_m, v_m).

In fig. S24 we see that b influences the position of the saddle-node bifurcation. To understand how b can affect the location of the bifurcation point (x_m, v_m), we now fix v and consider the subspace (b, x^*).

We find that in the limits $v \to \pm\infty$ the fixed points converge to $x^* \to \pm\frac{\alpha}{\omega^2}$, respectively (fig. S25). We find that for $v < \frac{\omega^2}{\alpha}$ (subcritical unit) the position of the fixed point is sensitive to the external current amplitude b (fig. S25). For $v > \frac{\omega^2}{\alpha}$ (supercritical unit) we find that the regions with three fixed points are bounded by the local minima and maxima of $b(x^*)$. We can obtain this range by setting $\partial b/\partial x^* = 0$ in (S15) and get

$$x_{\text{min,max}} = \pm\frac{\sqrt{v - \frac{\omega^2}{\alpha}}}{\frac{\omega^2}{\alpha}\sqrt{v}}. \tag{S18}$$

Plugging these values into (S15), we get

$$b_{\pm} = \tanh^{-1}\left(\pm\frac{\sqrt{v - \frac{\omega^2}{\alpha}}}{\sqrt{v}}\right) \mp \frac{\sqrt{v}\sqrt{v - \frac{\omega^2}{\alpha}}}{\frac{\omega^2}{\alpha}}, \tag{S19}$$

for the amplitude interval $[b_-, b_+]$ for which there exist three fixed points (fig. S26, red lines). When we concatenate the two previous bifurcation diagrams, we obtain the full picture of the parameter subspace (v, b). The resulting bifurcation diagram (fig. S26) describes a cusp bifurcation, where the boundaries of the diagram are given by $b_-(v)$ and $b_+(v)$ (see Equation S19).

Harmonic Input. Given the oscillatory nature of the recurrent input that each unit in a HORN receives, we study here how a single unit subject to amplitude feedback responds to harmonic stimuli to better understand its dynamics. A harmonic input is defined as $I(t) = A\sin(\omega_i t)$, where ω_i denotes the input frequency. Inserting this input into (S7), we obtain

$$\ddot{x} + 2\gamma\dot{x} + \omega^2 x = \alpha\tanh\left(vx + A\sin(\omega_i t)\right). \tag{S20}$$

The periodic input introduces a time-dependent term to the equation, which makes it non-autonomous. The parameter space is now composed of: the damping ratio γ, the natural frequency ω, the local excitability α, the amplitude feedback v, the amplitude of the input A and the input frequency ω_i.

For this system, the dynamics of the units (fig. S24 bottom) chase fixed points that change their positions due to the harmonic input. We will call these points moving fixed points [4]. One way to picture this scenario is to consider that our bifurcation diagram is periodically and continuously transitioning between the upper panels of fig. S24. In this scenario, we can identify three different regions: In region 1, the system converges to a single moving fixed point (fig. S24, red). In region 2, two alternating saddle-node bifurcations dynamically create a continuous 3 to 1 fixed points regime (fig. S24, blue). The detachment of a stable branch due to its collapse to the unstable branch, that is, the saddle-node bifurcation, is known as a tipping point phenomenon [5]. In region 3, the system has 3 nonvanishing moving fixed points (fig. S24 green). The fixed point oscillating around 0 poses the unstable branch, whereas the other two pose the stable branches. Here, system dynamics can be captured by the single stable branch or it is juggled between the two stable branches. The three regions are delimited by the non-driven bifurcation point $v = \frac{\omega^2}{\alpha}$ (fig. S24, dashed dot line) and the position of the local minima (Equation S16 for $b = A$) $v\left(x_m(A)\right)$ (fig. S24, dashed line). With the above-described scenario, we gain understanding on the stationary behavior of the DHO units in the presence of harmonic inputs and how they converge to the fixed points of the system.

Given the non-autonomous nature of the ODE, the initial conditions of the system are defined by the triple (x_0, y_0, t_0), where t denotes time. The absence of dependence of stationary behavior on initial conditions can result in two different types of attractiveness in nonautonomous systems, namely, forward and pullback attractiveness [6]. For the DHO units in our networks, such a dependency could imply two different modes of computation. In the first case, the computation relies on the initial state of the unit. In the other case, the computation exploits the stationary state of the unit.

Using simulated single unit dynamics, we numerically investigate the types of attractiveness of the DHO units for different parameter values and initial conditions. Since our input is harmonic, any initial

condition (x_0, y_0, t_0) can be expressed as $(x_0, y_0, t_0 + k\frac{2\pi}{\omega_i})$, with $k \in \mathbb{Z}$ due to the periodicity of the input. Without loss of generality, we restrict our analysis to the time window $[0, \frac{2\pi}{\omega_i}]$.

We start by inspecting pullback attractiveness. From fig. S27 we observe that subcritical and critical units seem to display pullback attraction, where for supercritical units this does not always seem to be the case.

To investigate forward attractiveness, we inspect the influence of different initial positions x_0 on system dynamics. From fig. S28 we conclude that subcritical and critical units seem to display forward attraction, while this does not always seem to be the case for supercritical units.

We conclude that all of our units possess converging trajectories and that the unit trajectories for the subcritical and critical units are not strongly dependent on the initial states of the units. This corroborates the idea that structurally different inputs result in the segregation of HORN system trajectories and that signals with similar characteristics result in similar evoked dynamics.

2.3 Interaction of amplitude and velocity feedback

The ODE for a DHO unit subject to both velocity and amplitude feedback is

$$\ddot{x}(t) + 2\gamma\dot{x}(t) + \omega^2 x(t) = \alpha \tanh\left(vx(t) + w\dot{x}(t)\right)), \tag{S21}$$

where v and w denote the amplitude and velocity feedback parameter, respectively.

Writing the above second order ODE as a system of first order ODEs, we get

$$\begin{aligned} \dot{x}(t) &= y(t) \\ \dot{y}(t) &= \alpha \tanh\left(vx(t) + wy(t)\right) - 2\gamma y(t) - \omega^2 x(t). \end{aligned} \tag{S22}$$

The entries of the Jacobian of this system are

$$\begin{aligned} a &= \frac{\partial \dot{x}}{\partial x} = 0, \\ b &= \frac{\partial \dot{x}}{\partial y} = 1, \\ c &= \frac{\partial \dot{y}}{\partial x} = -\omega^2 + v\alpha \operatorname{sech}^2\left(vx + wy\right), \\ d &= \frac{\partial \dot{y}}{\partial y} = -2\gamma + w\alpha \operatorname{sech}^2\left(vx + wy\right). \end{aligned} \tag{S23}$$

An in-depth analytic treatment of this system with both types of feedback is beyond the scope of the present work, but we performed numerical simulations to show how the different types of feedback interact (fig. S29). In particular, w allows the unit to adjust its damping (leading to a Hopf bifurcation that allows for sustained oscillations above a critical value) and v the natural frequency ω (leading to a pitchfork bifurcation at a critical value). The superposition of the two types of feedback results in a combination of both bifurcation scenarios, leading to complex dynamics (fig. S29, bottom right panel).

2.4 Resonance

Each DHO unit performs a nonlinear gain modulation of its input depending on its natural frequency ω and the input frequency ω_i (S7). The amplitude feedback parameter v influences the peak position of the gain curves (fig. S30). The velocity self-connection term w does not influence the gain curve.

Supplementary Figures

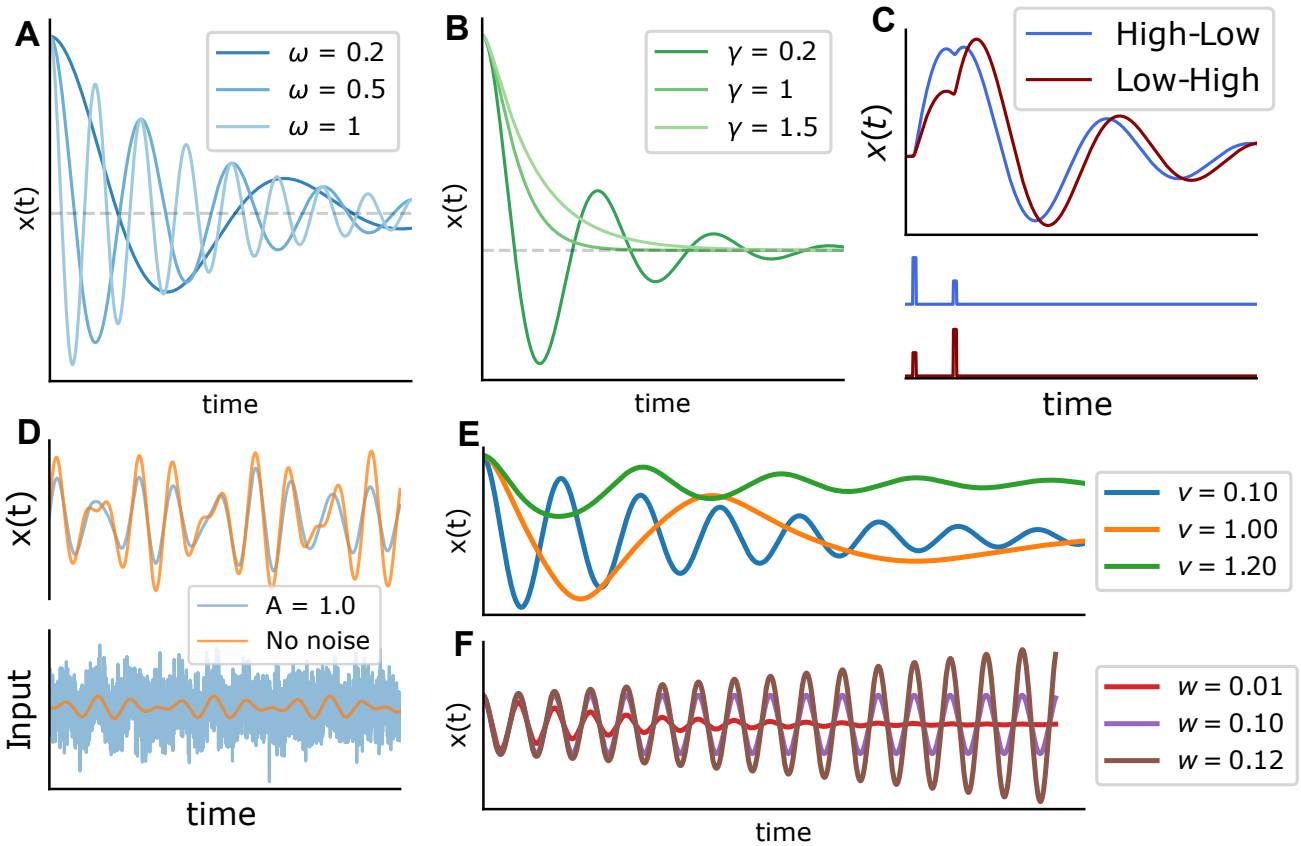

fig. S1. Relaxation dynamics of a single DHO unit ($\omega = 1$, $\gamma = 0.05$, $\alpha = 1$). A. Effect of the natural frequency parameter ω. B. Effect of the damping parameter γ. C. Sequence sensitivity of a DHO. The sequence order of two delta pulses with different amplitudes gets encoded in the oscillation phase. D. DHO unit response (top) to two different input signals (bottom). Input signals ans response color coded. Orange: input without noise, blue: input with additive white Gaussian noise. Note the similar amplitude trajectory $x(t)$ in both cases. E. Effect of the amplitude feedback parameter v on the relaxation dynamics of a DHO unit (initial condition $(x_0, y_0) = (1, 0)$). The critical value of v is $v_c = 1$. Note the emergence of two stable attractors for $v > v_c$. Relaxation dynamics to one of the attractors shown (green line). For $v < v_c$, only one attractor exists. F. Effect of the velocity feedback parameter w on the relaxation dynamics of a DHO unit (initial condition as in E). The critical value for w is $w_c = 0.1$. Note the emergence of a limit cycle for $w > w_c$ (Hopf bifurcation).

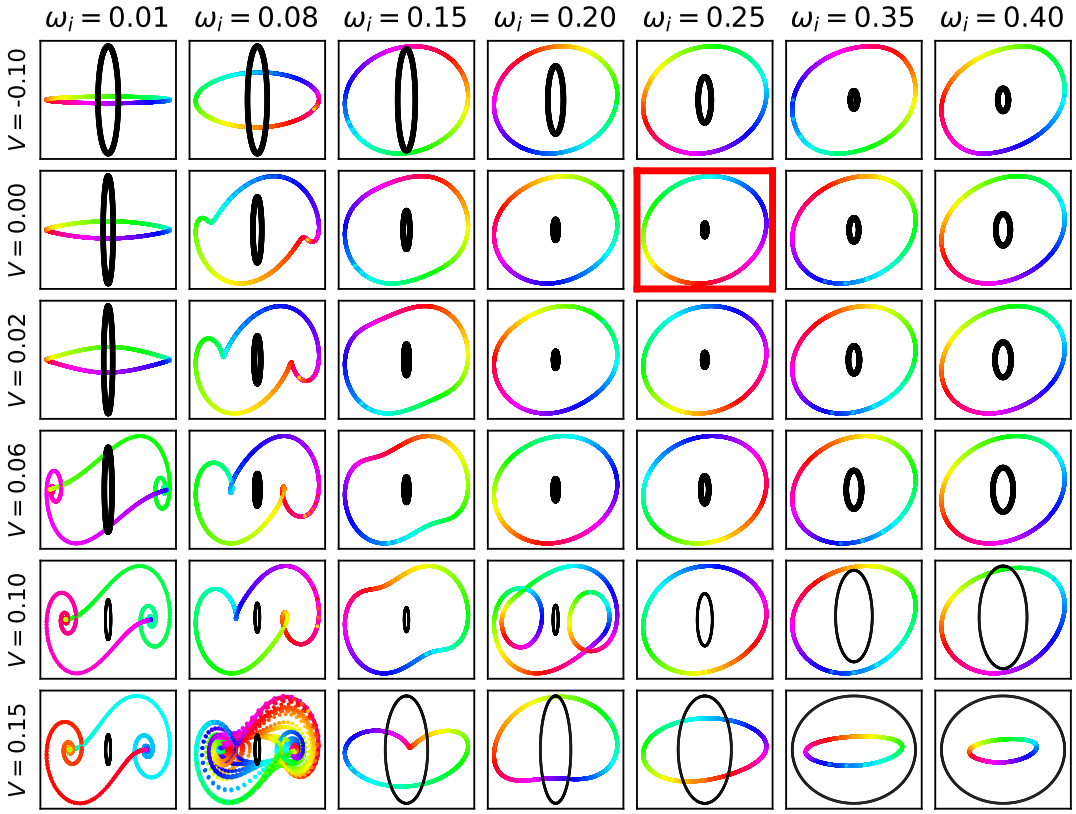

fig. S2. Dynamics of a single DHO unit with $\omega = 0.25$, $\gamma = 0.05$, $\alpha = 1$ driven by a harmonic input with an amplitude of $A = 1$ for different input frequencies ω_i and self-connection strengths V. Each panel shows the phase space of the stationary dynamics of the DHO unit (colored curve, color-coded according to the input phase), due to the driving harmonic input (black curve, unit circle). The panel corresponding to $\omega_i = 0.25$, $v = 0$ (red frame) shows the case of a simple damped harmonic oscillator without feedback.

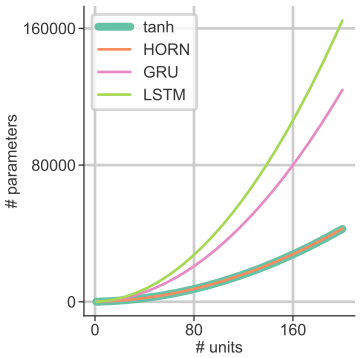

fig. S3. Number of trainable system parameters as a function of network size for different RNN architectures and sMNIST or pMNIST classification (input signal dimensionality 1, readout signal dimensionality 10).

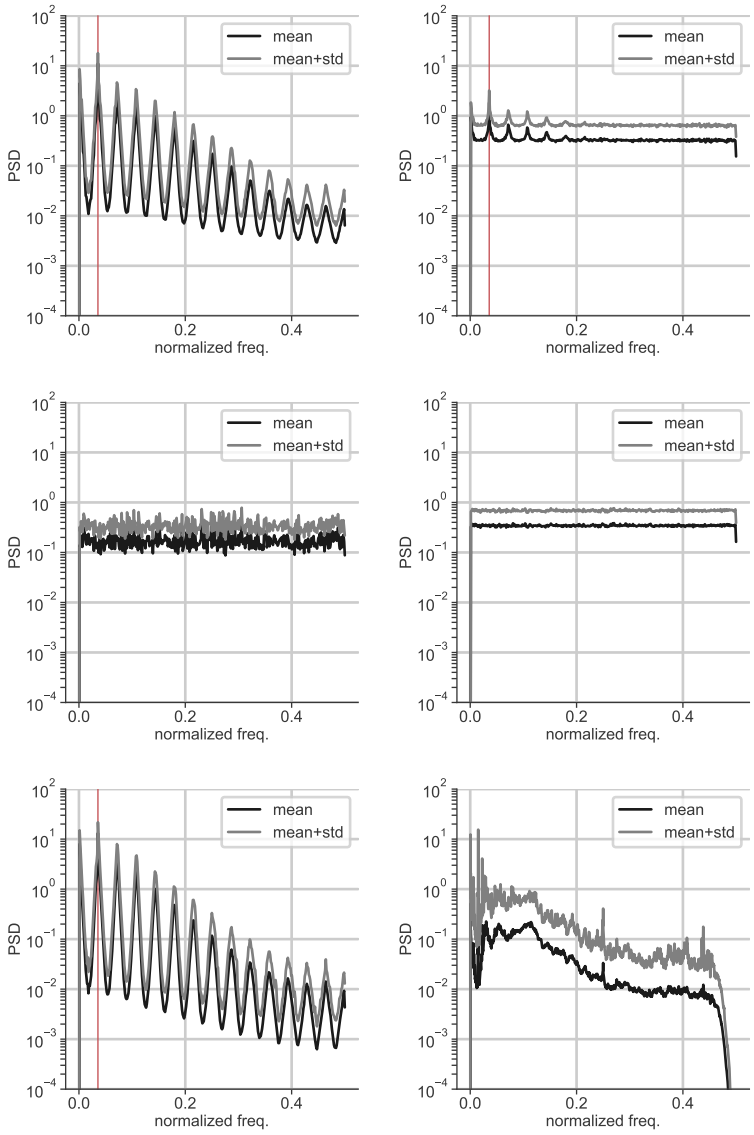

fig. S4. Power spectral densities for groups of samples from different data sets. Plots show mean and std.-dev. in PSD computed over 1000 samples (`scipy.periodogram` with `fs=1.0`). Vertical axes show PSD, horizontal axes show normalized frequency f. The red line indicates the line frequency of MNIST ($2\pi/28$). From top left to bottom right: sMNIST, sMNIST corrupted by $N(0,1)$ additive white Gaussian noise, pMNIST, pMNIST corrupted by $N(0,1)$ additive white Gaussian noise, EMNIST, Spoken Digits Data Set.

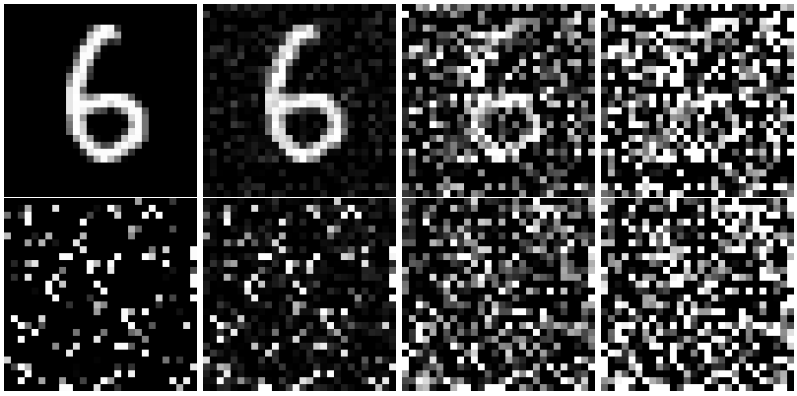

fig. S5. Example digit from MNIST data set in unshuffled (top row) and shuffled form (bottom row). Columns show different levels of additive white Gaussian noise applied per pixel with $\sigma^2 = 0, 0.1, 0.5, 1.0$ (from left to right). Stimuli are clipped to values in $[0, 1]$ after noise application.

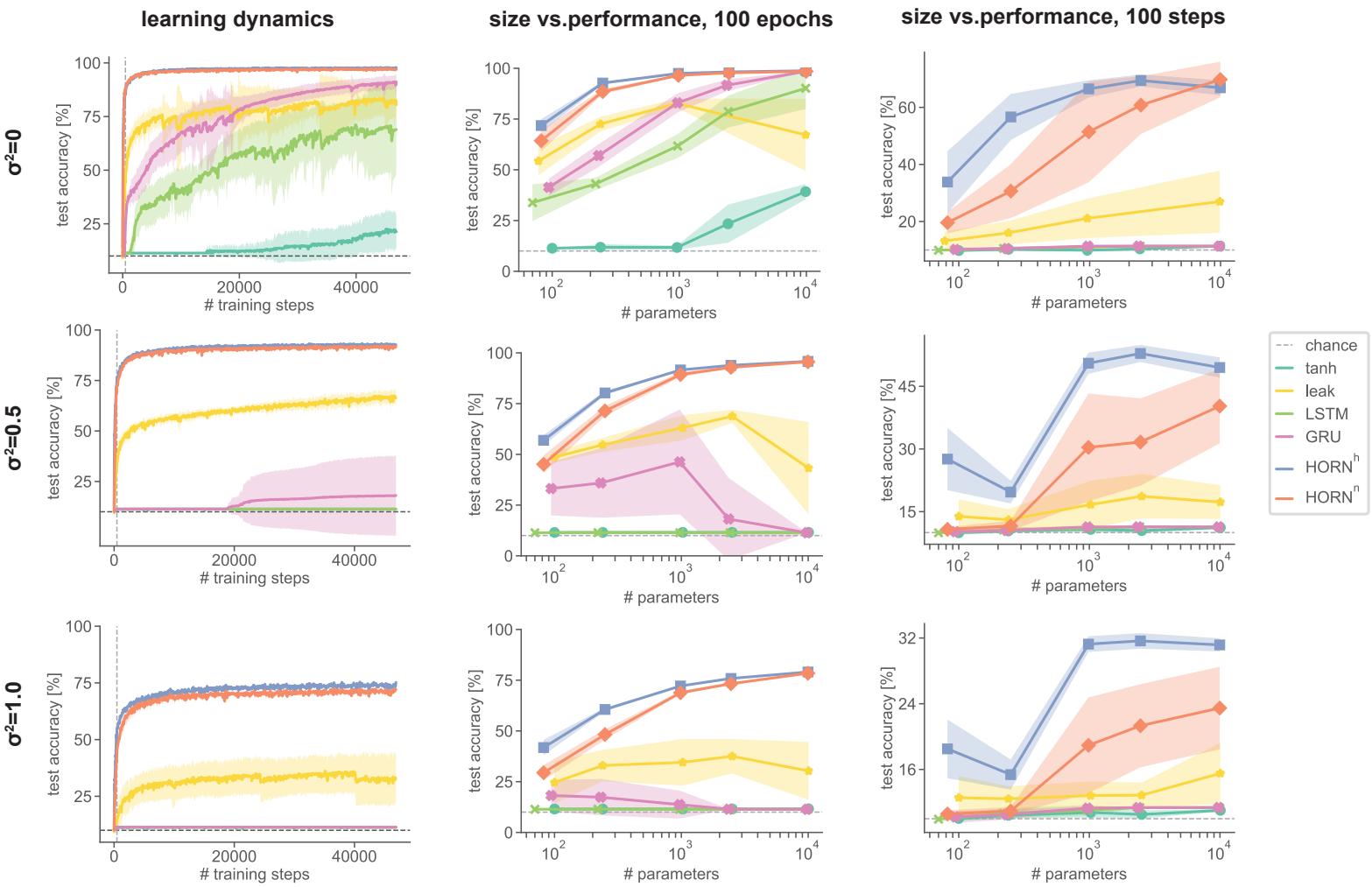

fig. S6. Learning dynamics and performance for different RNN architectures on sMNIST. Left column: Learning dynamics for a network with 2500 trainable parameters. Middle and right columns: Maximal accuracy after 100 learning epochs and after 100 BPTT update steps as a function of the number of trainable parameters, respectively. The rows show different noise levels σ^2. Lines show mean over 10 network instances with random weight initialization, shaded areas standard deviation. Vertical dashed lines mark the end of the first learning epoch.

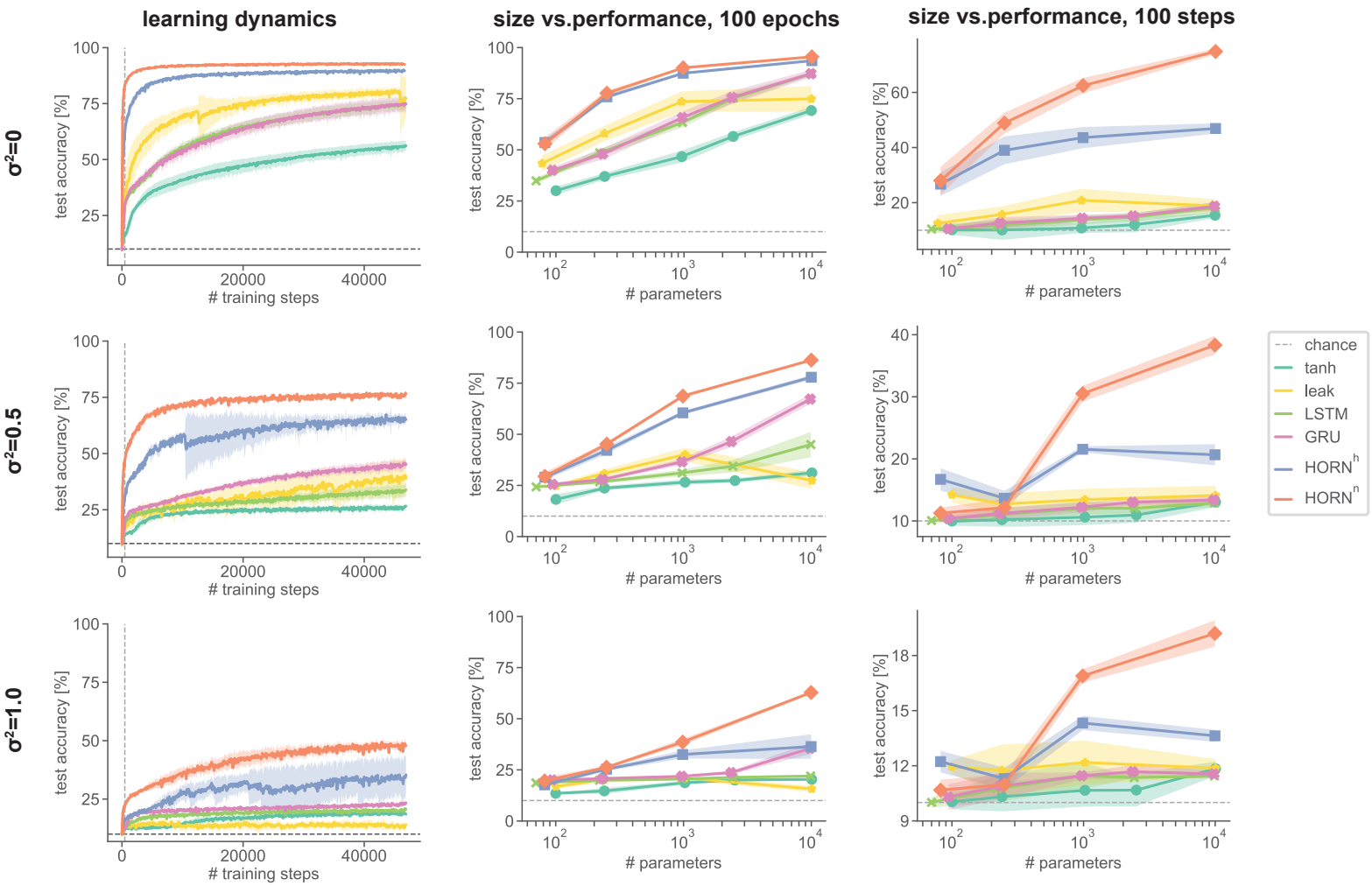

fig. S7. Learning dynamics and performance for different RNN architectures on pMNIST. Panels as in fig. S6.

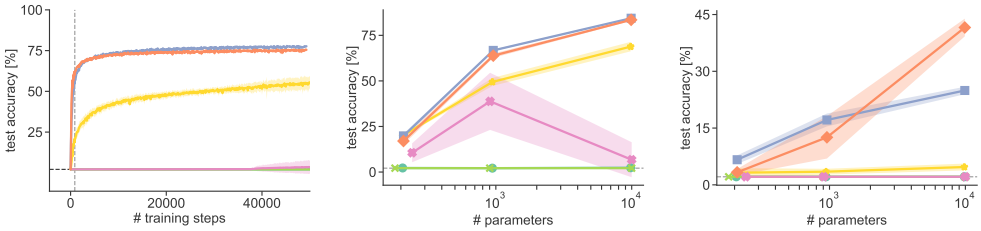

fig. S8. Performance of different RNN architectures on serialized EMNIST (balanced) data set. Left column: Learning dynamics for a network with 2500 trainable parameters. Middle and right columns: Maximal accuracy after 100 learning epochs and after 100 BPTT update steps as a function of the number of trainable parameters, respectively. Lines show mean over 10 network instances with random weight initialization, shaded areas standard deviation. Colors as in fig S6.

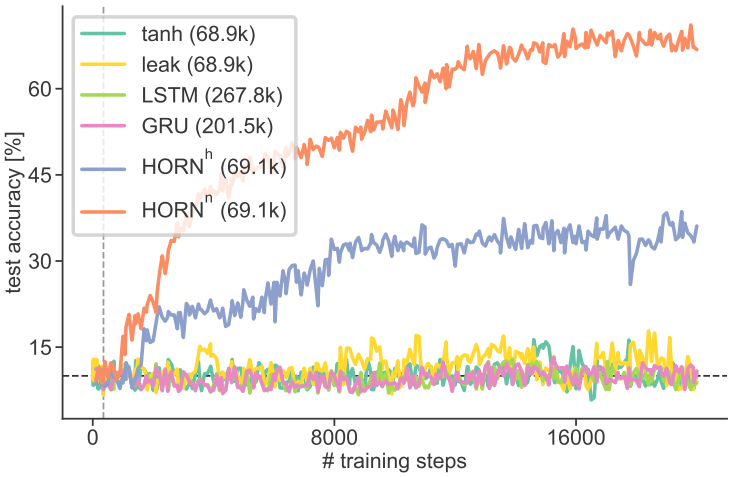

fig. S9. Classification accuracy as a function of training steps for different RNN architectures when trained on SDDS for 50 epochs (batch size 64). All networks had 256 units (number of trainable parameters in parentheses). HORN parameters as in Table S1.

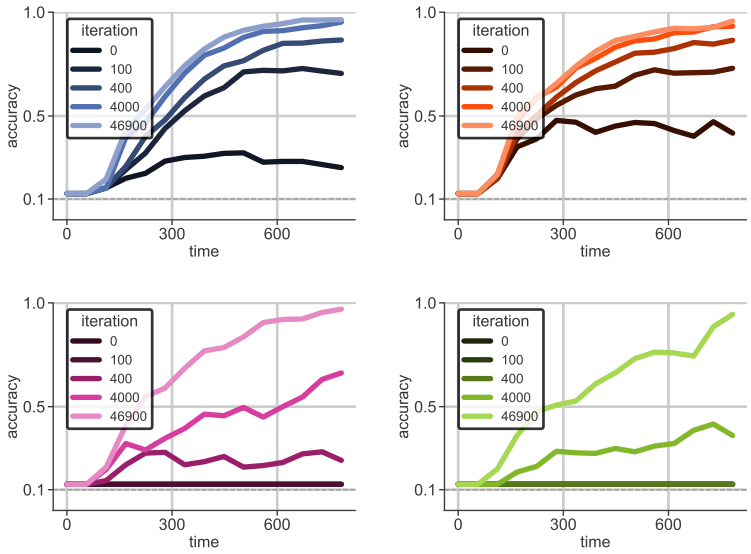

fig. S10. Dynamics of stimulus decodability by linear SVM from RNN activity vector for different architectures for different network instances during training on sMNIST for 100 epochs, from top left to bottom right HORN^h, HORN^n, LSTM, GRU. Color lightness indicates number of BPTT training iterations.

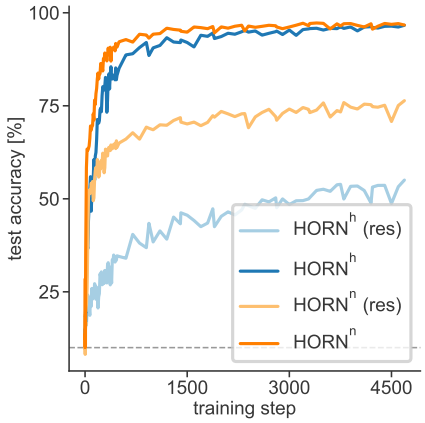

fig. S11. Learning dynamics of full and reservoir HORNs of size 128 during 10 training epochs on sMNIST. For reservoir HORNs, only input and readout layers are trained with BPTT.

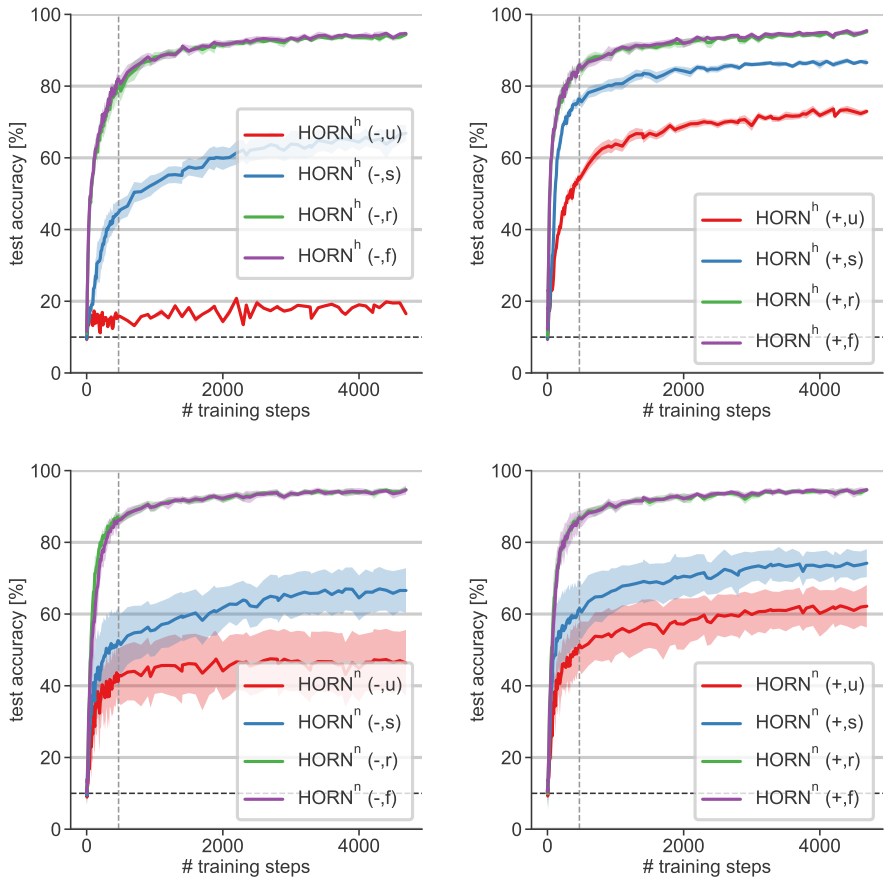

fig. S12. Learning dynamics of HORN networks (32 units) trained on sMNIST for 10 epochs under different connectivity constraints. Lines show mean for 10 random weight initializations, shaded areas standard deviation. Network label in parentheses indicates network connectivity pattern: "-" or "+" indicates presence of apmplitude feedback connection, "u" fully unconnected network of isolated units ($\mathbf{W}^{hh} = 0$), "i" indicates network of isolated units with self-connections in y state variable ($\mathbf{W}^{hh} = \mathrm{diag}(w_{11}, \ldots, w_{nn})$), "r" indicates fully connected network without velocity self-connections ($\mathbf{W}^{hh}_{ii} = 0$), "f" indicates fully connected network with velocity self-connections. Top row: homogeneous HORN, bottom row: hetergeneous HORN. Left column: no apmlitude feedback-connections in x, right column: amplitude feedback connections present.

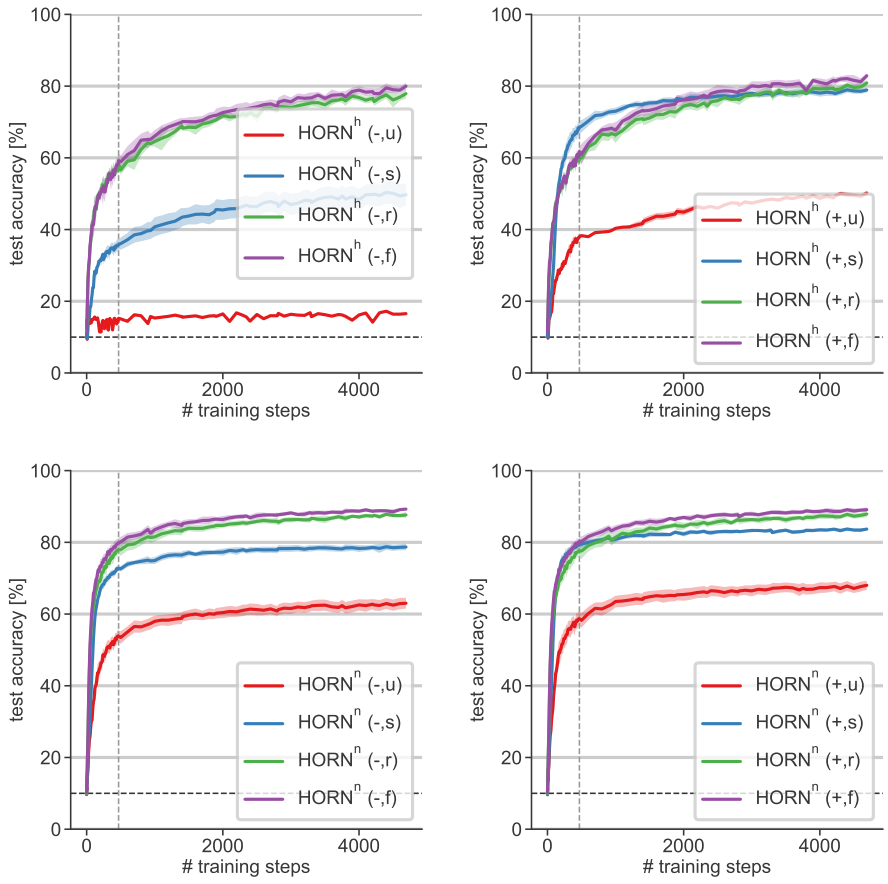

fig. S13. Learning dynamics of HORN networks (32 units) as in fig. S12 trained on pMNIST for 10 epochs.

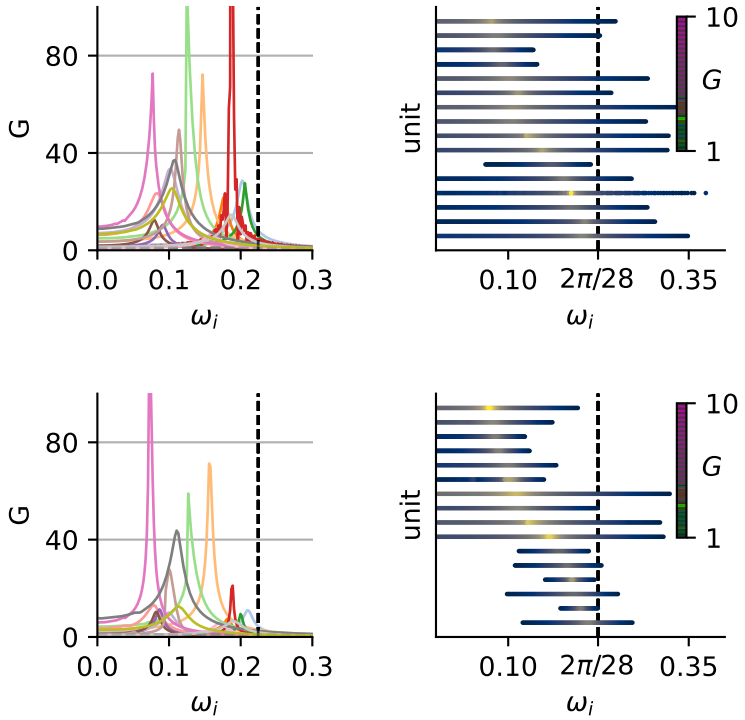

fig. S14. Intrinsic receptive fields (gain function G) of units in a heterogeneous HORN network (16 units) before (top row) and afer learning on sMNIST (bottom row) for 10 epochs.

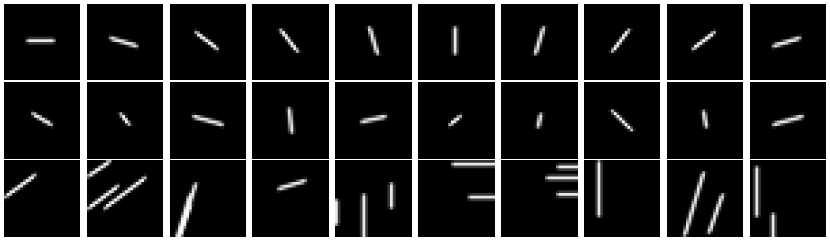

fig. S15. Samples of Line Segments data set. Top row: Angles present in LSDS(10,1,10,10,c). Middle row: Samples from LSDS(32,5,11,c) (referred to as LSDSa). Bottom row: Samples from LSDS(10,3,8,24,r) (referrred to as LSDSb).

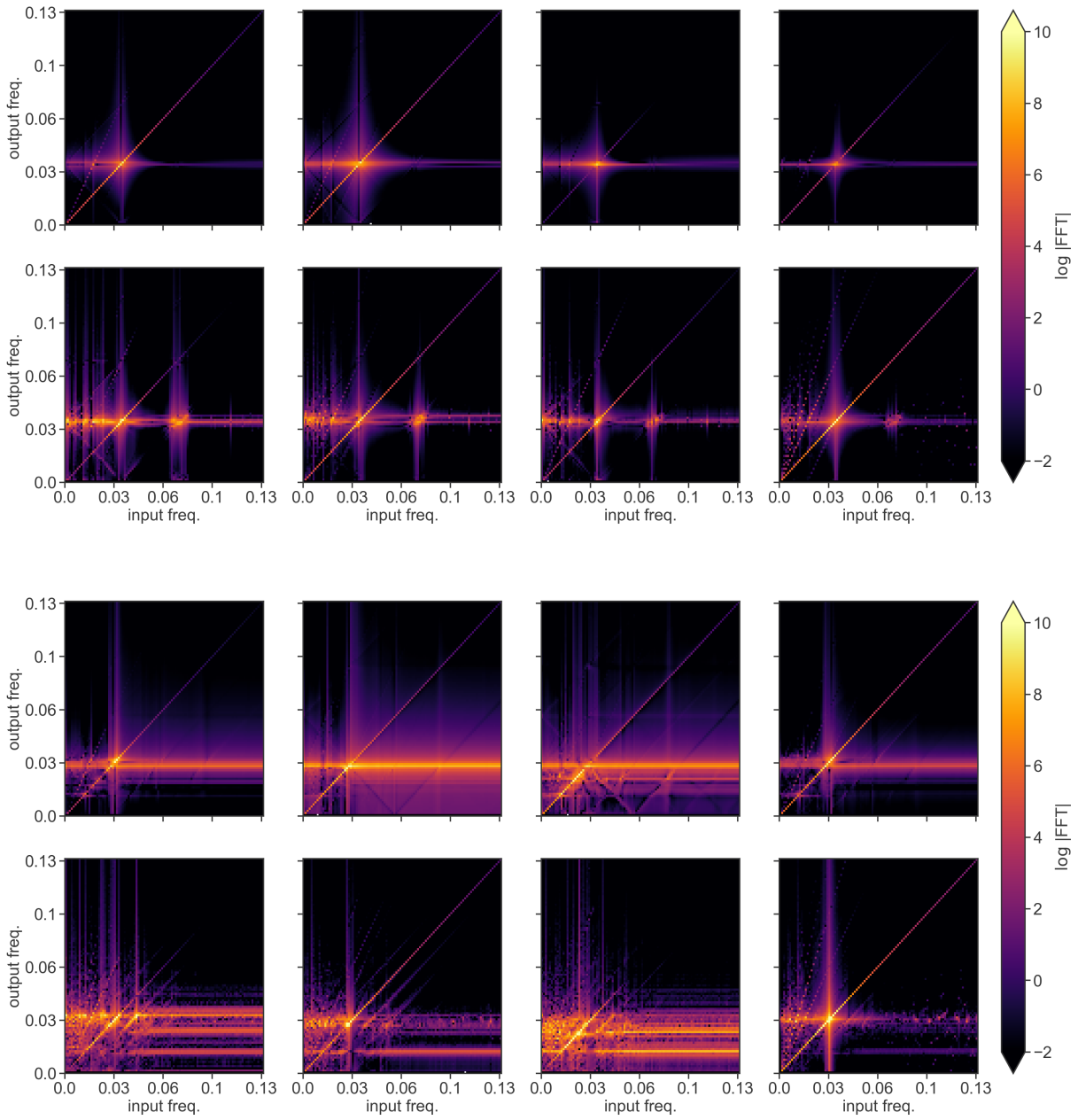

fig. S16. Resonance behavior of DHO units in HORN networks (16 units) in response to sinusoidal input stimulation before and after training on sMNIST for 100 epochs. Top panel shows homogeneous HORN, bottom panel shows heterogeneous HORN. Each panel shows four units from the network in an untrained state (first row) and after training (second row). Each matrix shows the logarithm of the power spectral density (PSD) of a unit's activity across frequency bands (output freq., vertical axis) as a function of the input frequency of the sinusoidal stimulus (input freq., horizontal axis). Color scale on the right, PSD values clipped to a range between 10^{-2} and 10^{10} to enhance visibility. Note that the bright horizontal lines correspond to the natural frequencies of the units, vertical lines indicate resonance at input frequencies around the units natural frequency, and bright diagonal lines indicate entrainment. Units in HORNs develop complex resonance behavior (bottom rows in both panels).

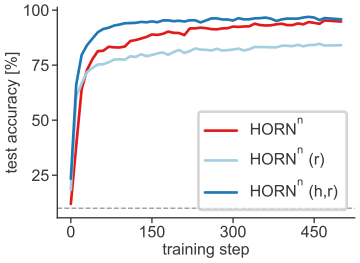

fig. S17. Learning dynamics of HORN^n network (93 units) when trained on sMNIST from scratch (red line), or with pre-training on LSDSb data set under different training conditions: (i) Only readout weights trained (light blue), and (ii) readout and recurrent weights trained (dark blue).

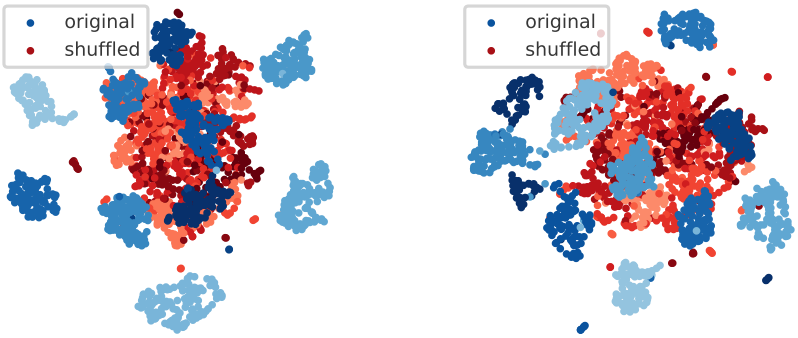

fig. S18. UMAP projection of homogeneous (left, 93 units) and heterogeneous HORN (right, 93 units) network state at the roadout time ($t = 784$) after training on sMNIST for 10 epochs. Activity elicited by sMNIST stimuli shown in blue shades, acitivity elicited by pMNIST stimuli shown in red shades. MNIST digit class coded in color lightness in both cases.

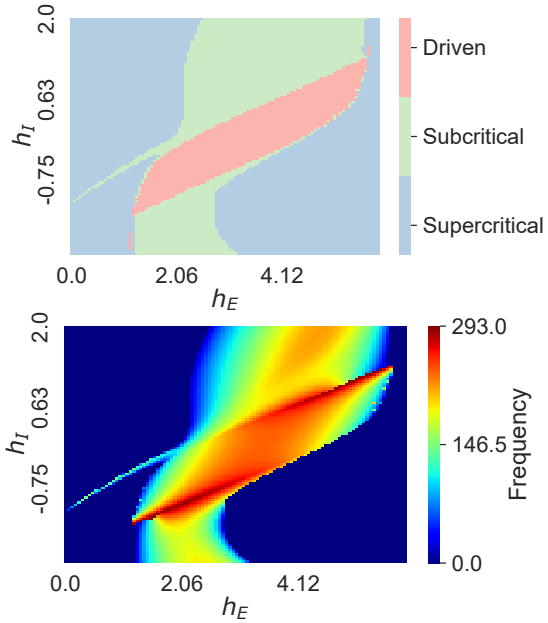

fig. S19. Wilson-Cowan dynamics in h_E-h_I subspace. Top: Bifurcation diagram, blue: supercritically damped harmonic oscillations, green: subcritically damped harmonic oscillations, red: driven subcritically damped harmonic oscillations. Bottom: Oscillation frequencies (frequency units arbitrary).

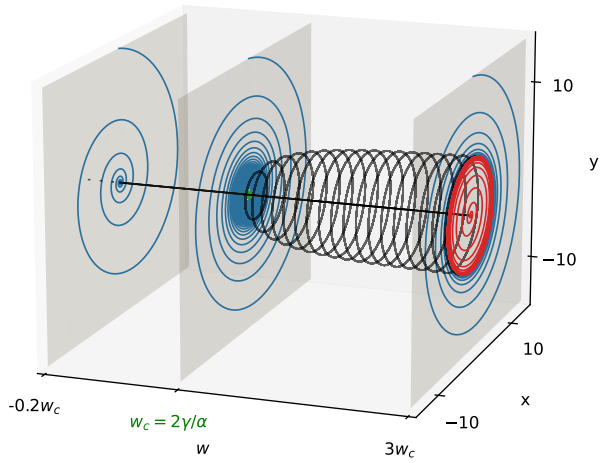

fig. S20. Hopf bifurcation of a DHO unit ($\omega = 1$, $\alpha = 1$ and $\gamma = 0.1$) with velocity feedback term w at a critical value $w_c = 2\gamma/\alpha = 0.2$. Blue lines: Unit relaxation dynamics from the initial condition $(x_0, y_0) = (0, 15)$ for different values of w. Green dot: Hopf bifurcation point at $w = w_c$. Black circles: limit cycle solutions with a radius depending on w. Red line: Relaxation dynamics from the initial condition $(x_0, y_0) = (0, 0.1)$ for $w = 0.6$. Black line: $(x, y) = (0, 0)$.

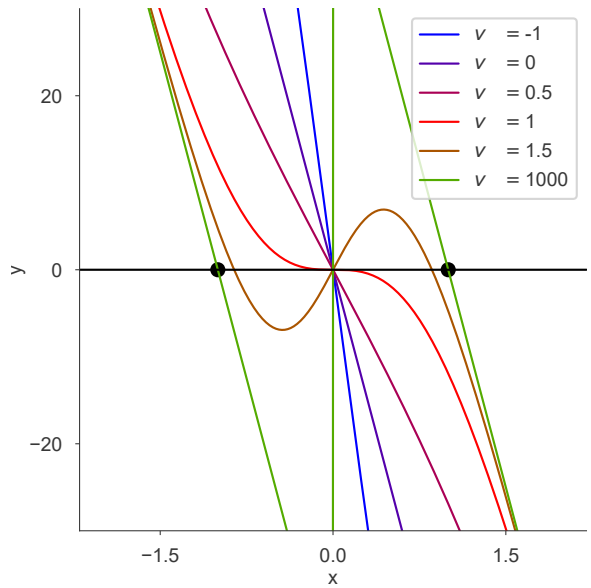

fig. S21. Intersections between the nullclines of DHO system dynamics. Black curve: $\dot{x} = 0$ (Equation S10). Other curves: $\dot{y} = 0$ (Equation S9, $\alpha = \omega = 1$) for different values of amplitude feedback parameter v (color coded). Black dots marks the points $(\pm\frac{\alpha}{\omega^2}, 0)$.

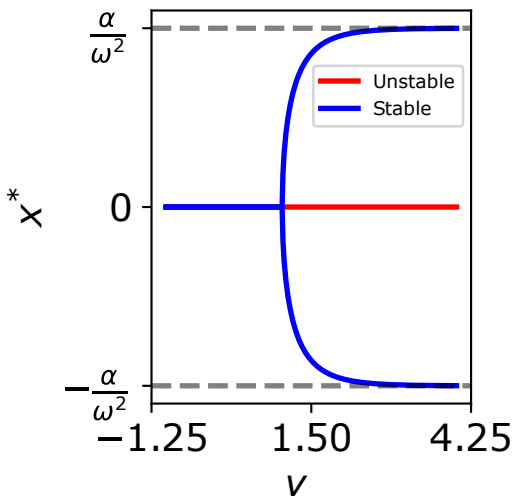

fig. S22. Bifurcation diagram of DHO unit ($\alpha = \omega^2 = 1$) dynamics under changes in the amplitude feedback parameter v in (v, x^*)-subspace. Pitchfork bifurcation point at $v_c = \frac{\omega^2}{\alpha}$.

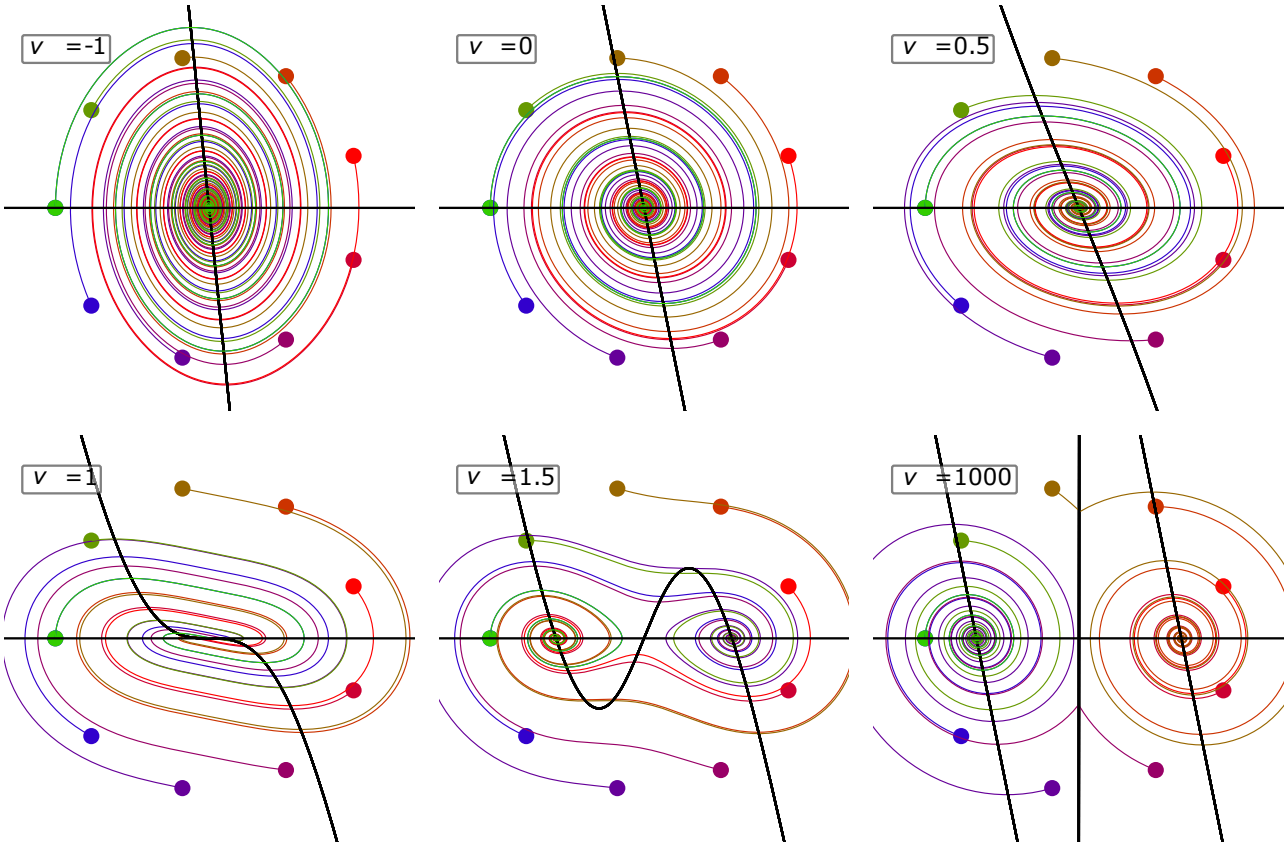

fig. S23. Phase portraits of relaxation dynamics of DHO unit ($\omega = 1$, $\alpha = 1$, $\gamma = 0.1$) for different values of the amplitude feedback parameter v. Different initial conditions color coded.

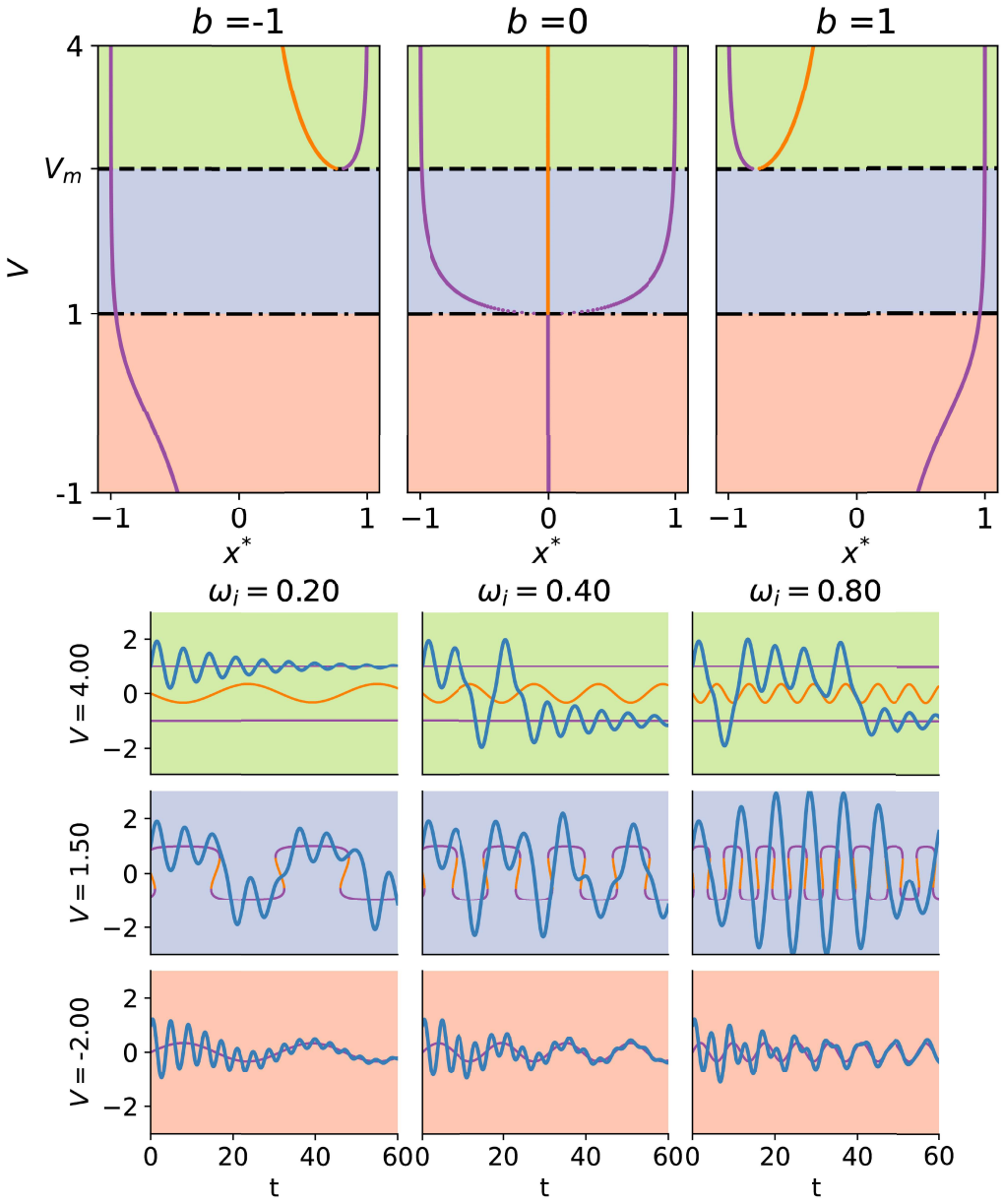

fig. S24. Top: Bifurcation diagrams of DHO unit dynamics for constant external input. Panels show bifurcation diagrams for different values of the amplitude b of a constant external input. Dashed line: Local minima for $v = v_m$ (Equation S16). Dash-dotted line: Pitchfork bifurcation point at $v = v_c = 1$. Orange curve: one unstable branch, purple curves: two stable branches. Regions color code qualitative system dynamics for harmonic input. Green: 3 fixed points, blue: 3 or 1 fixed point. Red: 1 fixed point. Bottom: DHO fixed point and unit dynamics for harmonic input. Panels show cases for different values of harmonic input frequency ω_i and amplitude feedback values v. Purple: stable branch, orange: unstable branch, blue: unit trajectory. Background colors indicate corresponding values of v.

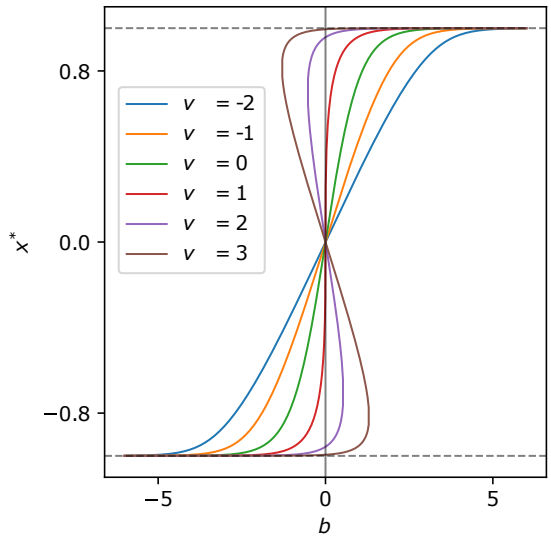

fig. S25. Position of DHO unit's ($\omega = 1$, $\alpha = 1$, $\gamma = 0.01$) fixed points for a constant input of amplitude b and different values of the amplitude feedback parameter v (color coded).

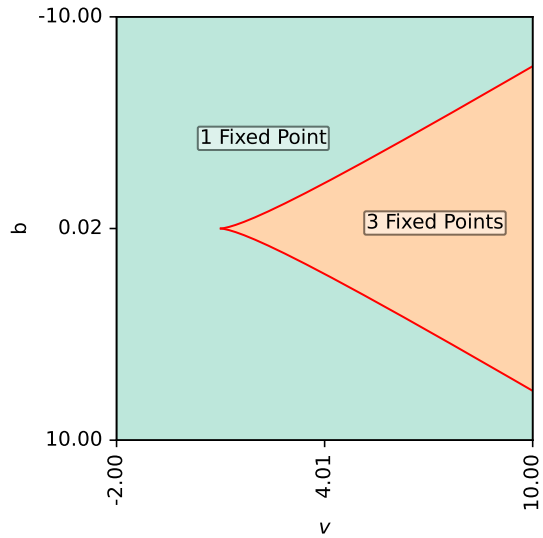

fig. S26. Cusp bifurcation of the dynamics of a DHO unit ($\omega = 1$, $\alpha = 1$, $\gamma = 0.01$) subject to an external constant input of amplitude b with amplitude feedback strength v, shown in (v, b)-space.

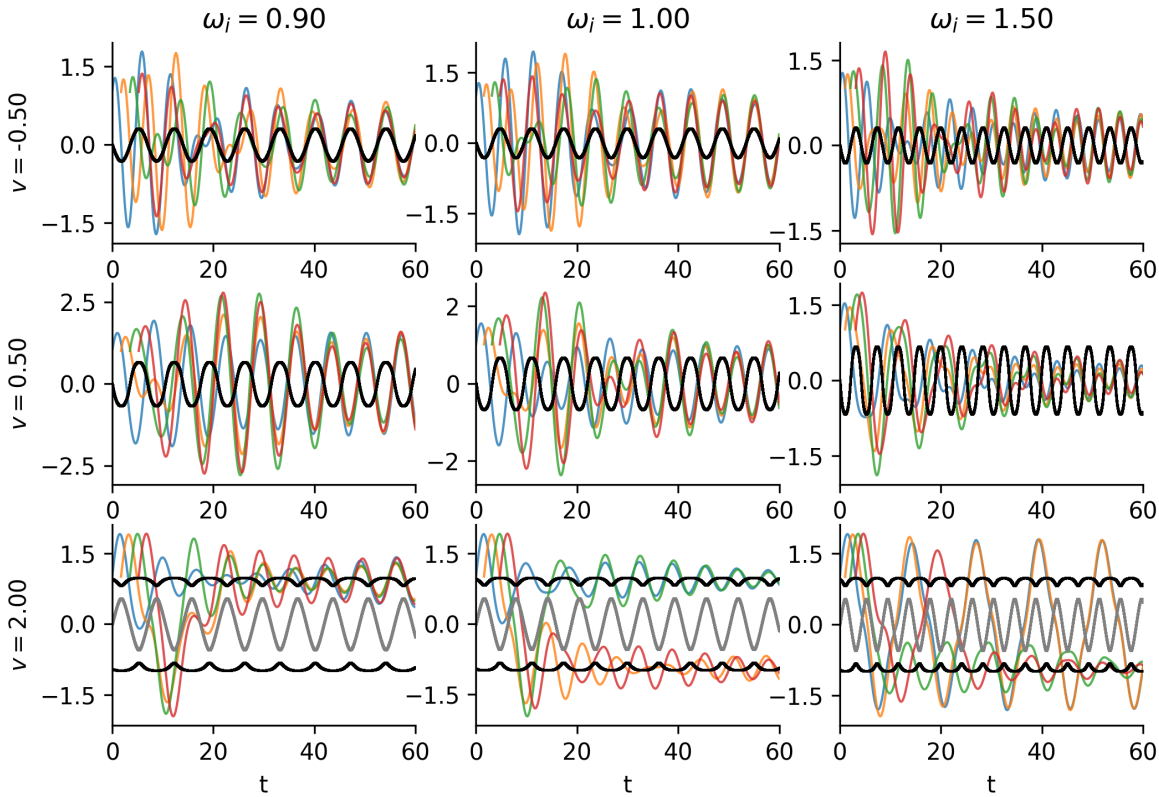

fig. S27. DHO unit ($\omega = 1$, $\gamma = 0.05$, $\alpha = 1$) relaxation dynamics dependence on the values of initial time t_0. Panels show cases for different values of harmonic input frequency ω_i and amplitude feedback values v. Blue: $t_0 = 0$, orange: $t_0 = \frac{\pi}{2\omega_i}$, green: $t_0 = \frac{\pi}{\omega_i}$, red: $t_0 = \frac{3\pi}{2\omega_i}$, black: stable moving fixed point, gray: unstable moving fixed point.

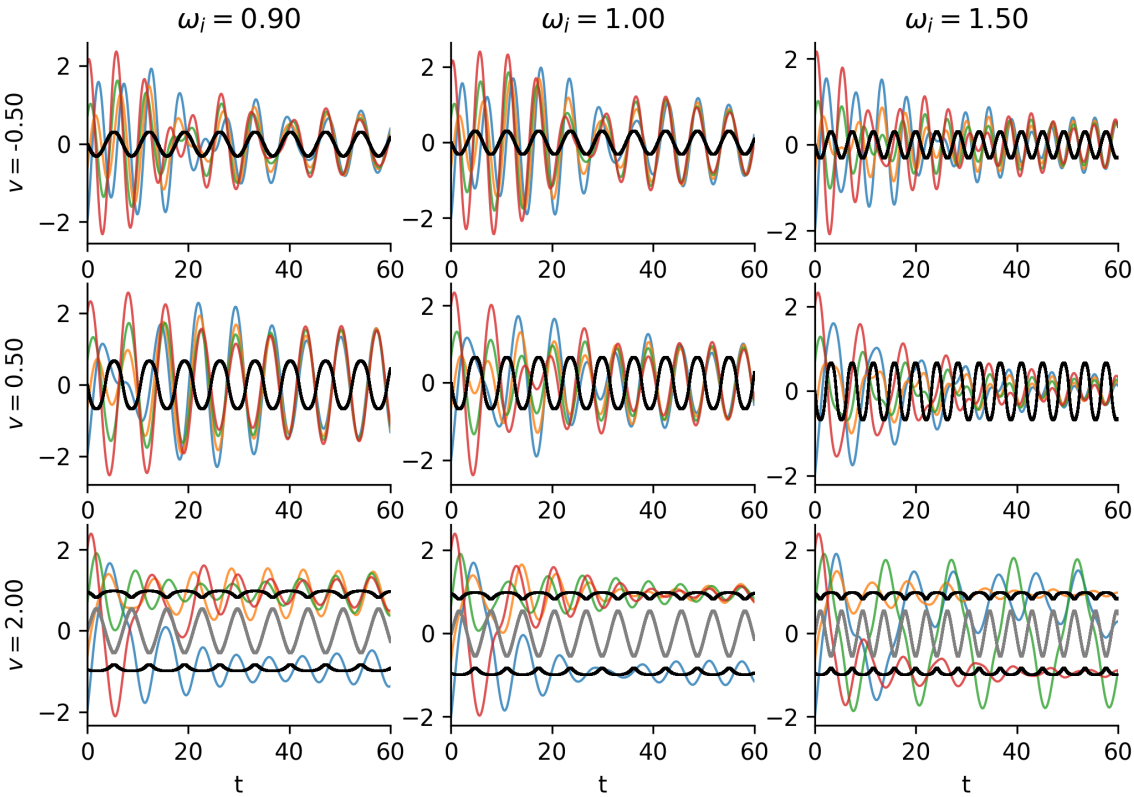

fig. S28. DHO unit ($\omega = 1$,$\gamma = 0.05$, $\alpha = 1$) relaxation dynamics dependence on the values of initial amplitude x_0. Panels show cases for different values of harmonic input frequency ω_i and amplitude feedback values v. Blue: $x_0 = -2$, orange: $x_0 = -1$, green: $x_0 = 1$, red: $x_0 = 2$, black: stable moving fixed point, gray: unstable moving fixed point.

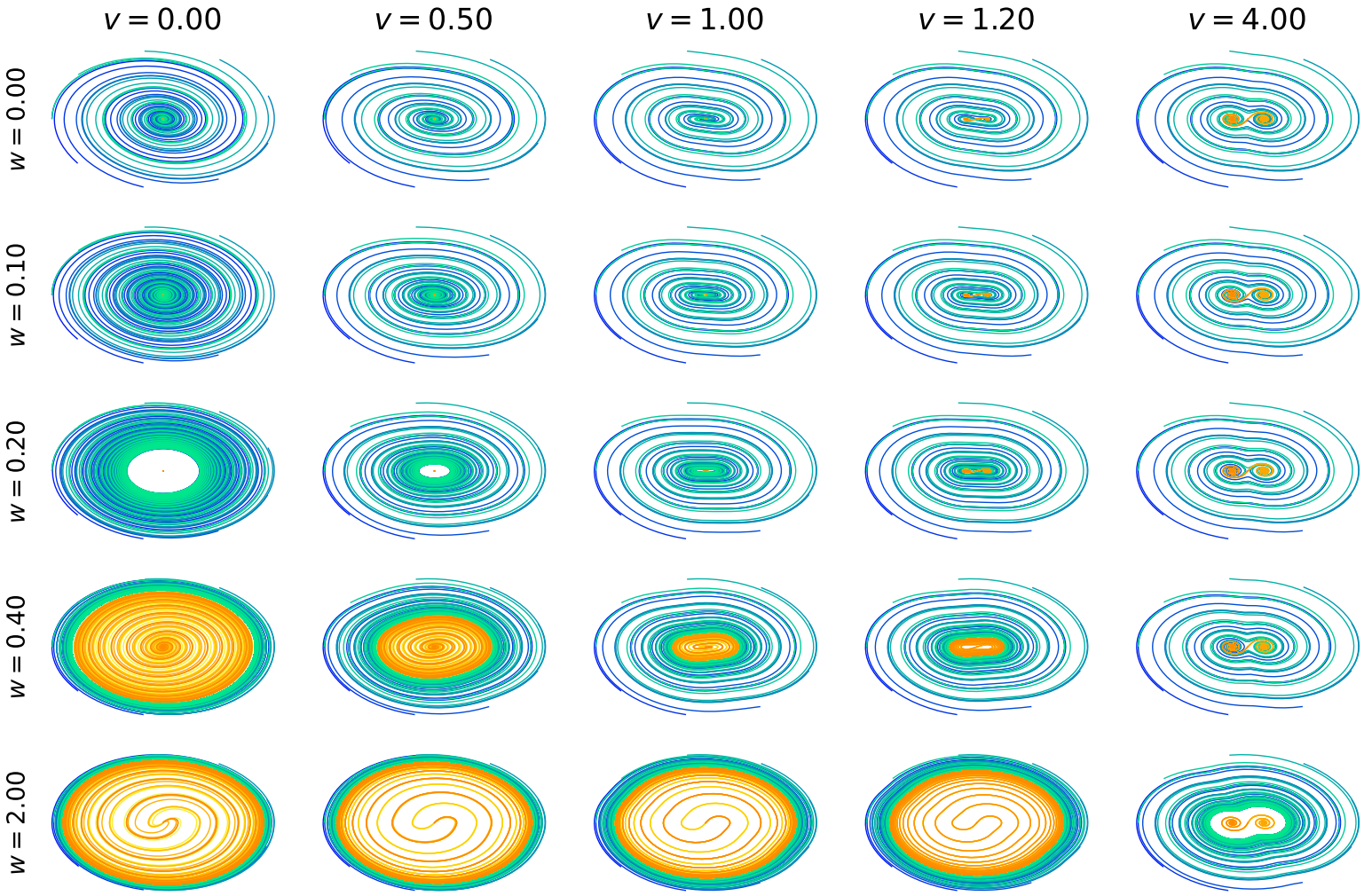

fig. S29. DHO unit ($\omega = 1$,$\gamma = 0.05$, $\alpha = 1$) relaxation dynamics for different values of the amplitude feedback parameter v and the velocity feedback parameter w. Panels show system trajectories for different initial conditions x_0 (color coded). Blue: $x_0 = -2$, orange: $x_0 = -1$, green: $x_0 = 1$, red: $x_0 = 2$, black: stable moving fixed point, gray: unstable moving fixed point.

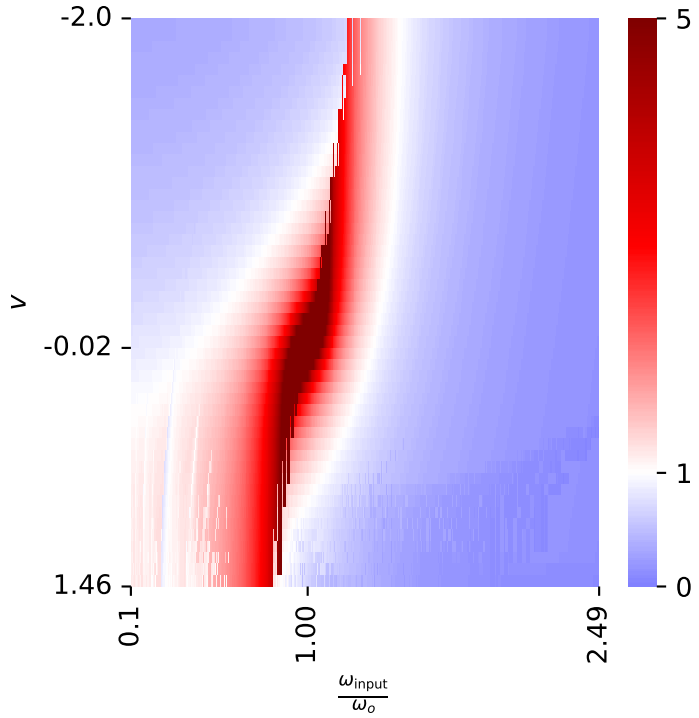

fig. S30. Dependence of the gain function G of a DHO unit ($\gamma = 0.1$, $\omega = 1$, $\alpha = 1$) on the value of the amplitude feedback parameter v. Gain color coded. Red shades indicate regions of resonance ($G > 1$), blue shades indicate amplitude depression ($G < 1$).

Table 1: Parameter values for studied RNN architectures on different data sets. Columns ω_p, γ, α show parameter values for HORN networks. First value in each column specifies parameter value for homogeneous HORN, values in parentheses denote value ranges for heterogeneous HORN. Parameter values of the natural frequency are given in period units of the input, and $\omega = 2\pi/\omega_p$. Columns h, a show parameters for leak RNN.

Dataset	$\boldsymbol{\omega_p}$	$\boldsymbol{\gamma}$	$\boldsymbol{\alpha}$	$\boldsymbol{h}$	$\boldsymbol{a}$
sMNIST	28 ($[28, 85]$)	10^{-2} ($[10^{-4}, 2 \cdot 10^{-2}]$)	$4 \cdot 10^{-2}$ ($[10^{-2}, 1.6 \cdot 10^{-1}]$)	0.2	0.2
pMNIST	7 ($[6, 22]$)	$5 \cdot 10^{-2}$ ($[1.5 \cdot 10^{-2}, 9 \cdot 10^{-2}]$)	$2.5 \cdot 10^{-1}$ ($[9 \cdot 10^{-2}, 3.6 \cdot 10^{-1}]$)	0.5	0.3
EMNIST	28 ($[30, 85]$)	$5 \cdot 10^{-2}$ ($[10^{-3}, 5 \cdot 10^{-2}]$)	10^{-2} ($[2.5 \cdot 10^{-2}, 4 \cdot 10^{-2}]$)	0.1	0.2
LSDS	28 ($[28, 85]$)	10^{-2} ($[10^{-4}, 2 \cdot 10^{-2}]$)	$4 \cdot 10^{-2}$ ($[10^{-2}, 1.6 \cdot 10^{-1}]$)	0.2	0.2
SDDS	63 ($[42, 126]$)	$5 \cdot 10^{-4}$ ($[5 \cdot 10^{-5}, 1.5 \cdot 10^{-3}]$)	10^{-2} ($[2.5 \cdot 10^{-3}, 2.5 \cdot 10^{-2}]$)	0.1	0.2
MNIST	18	10^{-3}	10^{-2}	–	–

Movie S1. Temporal evolution of the activity of the units of a 196 node HORN^h trained on geometrically organized MNIST input for 10 example digits. Units are arranged on a 14×14 matrix according to the geometric location of the nodes' 2×2 receptive fields in the 28×28 stimulus matrix. Each trial lasts 150 time steps, and the stimulus is presented during the first time step and then switched off. Node amplitudes are color-coded (color scale $[-0.1, 1.0]$ for first 50 time steps and $[-2.0, 2.0]$ for the remaining time steps to enhance visibility). Current time step t and stimulus identity shown at the top, asterisk marks stimulus presentation.

Movie S2. Temporal evolution of the phases of the units of a 196 node HORN^h trained on geometrically organized MNIST input for 10 example digits. Node phases $\phi \in [-\pi, \pi]$ are color-coded. Trials and unit placement correspond to the ones in Movie S1.

References

[1] Jack D. Cowan, Jeremy Neuman, and Wim van Drongelen. Wilson–Cowan Equations for Neocortical Dynamics. *The Journal of Mathematical Neuroscience*, 6(1):1, December 2016.

[2] Alberto Pérez-Cervera, Tere M. Seara, and Gemma Huguet. Phase-locked states in oscillating neural networks and their role in neural communication. *Communications in Nonlinear Science and Numerical Simulation*, 80:104992, 2020.

[3] Y. Kuznetsov. *Elements of applied bifurcation theory*. Applied mathematical sciences. Springer New York, 2008.

[4] Jeremiah H. Li, Felix X.-F. Ye, Hong Qian, and Sui Huang. Time-dependent saddle-node bifurcation: Breaking time and the point of no return in a non-autonomous model of critical transitions. *Physica D: Nonlinear Phenomena*, 395:7–14, August 2019. Publisher: Elsevier BV.

[5] Jielin Zhu, Rachel Kuske, and Thomas Erneux. Tipping points near a delayed saddle node bifurcation with periodic forcing. *Siam Journal On Applied Dynamical Systems*, 14(4):2030–2068, 2015. Publisher: Society for Industrial & Applied Mathematics (SIAM).

[6] P.E. Kloeden and M. Rasmussen. *Nonautonomous dynamical systems*. Mathematical surveys and monographs. American Mathematical Society, 2011. tex.lccn: 2011020550.