Robust coding with spiking networks: a geometric perspective

Nuno Calaim[1†], **Florian Alexander Dehmelt**[1,2†], **Pedro J. Gonçalves**[3,4†], **Christian K. Machens**[1*]

[1]Champalimaud Research, Champalimaud Centre for the Unknown, Lisbon, Portugal; [2]Centre for Integrative Neuroscience, Tuebingen University Hospital, Tuebingen, Germany; [3]Max Planck Research Group Neural Systems Analysis, Center of Advanced European Studies and Research (caesar), Bonn, Germany; [4]Computational Neuroengineering, Department of Electrical and Computer Engineering, Technical University of Munich, Germany

Abstract The interactions of large groups of spiking neurons have been difficult to understand or visualise. Using simple geometric pictures, we here illustrate the spike-by-spike dynamics of networks based on efficient spike coding, and we highlight the conditions under which they can preserve their function against various perturbations. We show that their dynamics are confined to a small geometric object, a 'convex polytope', in an abstract error space. Changes in network parameters (such as number of neurons, dimensionality of the inputs, firing thresholds, synaptic weights, or transmission delays) can all be understood as deformations of this polytope. Using these insights, we show that the core functionality of these network models, just like their biological counterparts, is preserved as long as perturbations do not destroy the shape of the geometric object. We suggest that this single principle—efficient spike coding—may be key to understanding the robustness of neural systems at the circuit level.

Introduction

The dynamics of neural networks are usually analysed and understood by focusing on neurons' firing rates. The resulting network models have provided a host of intuitions about the types of computations that can be carried out with neural networks, from feedforward architectures to winner-take-all networks, associative memories, neural integrators, or working memory [1]. Despite these successes, it is not entirely clear that these network models are the 'right' way to explain the dynamics of neural circuits. Most neurons spike, and it has proven surprisingly difficult to translate results on rate networks into equivalent spiking neural networks when biological observations (such as irregular, asynchronous firing and low firing rates) are taken into account [2, 3].

A key hurdle is that we lack intuitions on how to think about communication with spikes at the network level. Many ideas of how to compute with spikes on the single-neuron level have been developed [4–11], but making these ideas work on the network level, while staying within realistic biological regimes, has often proven challenging. A crucial step forward was the development of balanced networks, which highlighted the conditions under which neural networks generate irregular and asynchronous spike trains [3, 12–16], as well as correlated fluctuations [17, 18]. While balance was initially just an implementational constraint imposed on neural circuitry, it was recently given a functional explanation in terms of efficient coding [19–21]. In these networks, which have been called 'spike coding networks' [2, 22], the dynamics of balancing were equated with self-correcting properties of the network. Interestingly, these networks showed themselves robust to perturbations such as neuron loss [19, 23].

We here show that these networks lend themselves to a geometric description that provides a host of insights

[†]These authors contributed equally to this work.

For correspondence: christian.machens@neuro.fchampalimaud.org (CKM)

about their spiking dynamics. In particular, the geometric view suggests a unifying principle for how neural circuits may have become robust to many perturbations encountered in nature (***Figure 1***). We use our geometric framework to study what happens when these systems are scaled up to realistic sizes (coding of hundreds of dimensions with thousands of neurons). We show how the geometric framework nicely illustrates what happens when neurons are destroyed, or when biophysical parameters such as synaptic strengths, spiking thresholds, or transmission delays, etc. are altered or perturbed from their optimal values. Finally, we illustrate how the framework can shed new light on optogenetic perturbation experiments, suggesting that neural circuits should be sensitive to small excitatory perturbations, yet insensitive to broad inhibitory perturbations.

In doing so, we both reproduce some previous findings (e.g. robustness to neuron loss [23]) and report new findings (e.g. on noise, delays, and optogenetics). Our key contribution here is to provide a geometric interpretation of spike coding networks (SCNs) and their robustness. This geometric framework allows us to visualise both the network's spiking dynamics and its various biophysical parameters in a lower-dimensional error space, thereby providing straightforward intuitions about how changes in the network's parameters affect the dynamics.

Results

Spike coding networks are based on the hypothesis that neural populations compute with analog quantities, such as membrane currents and voltages, and that they fire spikes only to encode and decode the 'signals' resulting from these computations [19–22, 24]. Their function is best illustrated in a network whose sole purpose is to encode a given set of time-varying input signals $\mathbf{x}(t) = \big(x_1(t), x_2(t), \dots, x_M(t)\big)$ into spike trains, such that one can reconstruct the signals using a linear readout (***Figure 2A***). We will focus exclusively on this simple autoencoder-network in order to highlight the mechanisms that make the network robust. As *SCN*s are capable of implementing more complex computations, we will show in the discussion how to transfer these insights to more general networks. We here replicate the derivations outlined in Boerlin et al. [19] and Barrett et al. [23], with some minor variations, and we show how to construct a geometric explanation of the network's spiking behavior.

The error bounding box

The architecture of spike coding networks is derived from two assumptions. The first assumption is that all signals can be decoded linearly from the network's spike trains. In other words, rather than specifying how (input) signals are mapped onto spike trains, we specify how spike trains are mapped into (output) signals (***Figure 2A***). In the 'linear readout' mapping, each spike train is convolved with an exponential filter, similar to the postsynaptic potentials generated in a single synapse. Then, the filtered spike trains are weighted and summed, similar to the passive summation in a dendritic tree. Formally, we write

$$\hat{\mathbf{x}}(t) = \sum_{k=1}^{N} \mathbf{D}_k r_k(t), \tag{1}$$

where $r_k(t)$ is the filtered spike train of the k-th neuron, N is the number of neurons, $\hat{\mathbf{x}}(t) = \big(\hat{x}_1(t), \hat{x}_2(t), \dots, \hat{x}_M(t)\big)$ is the vector of readouts, and $\mathbf{D}_k = (D_{1k}, D_{2k}, \dots, D_{Mk})$ is the decoding vector of the k-th neuron, whose individual elements contain the respective decoding weights.

To illustrate the geometrical consequences of this decoding mechanism, we imagine a network of five neurons that is encoding two signals. At a given point in time, we can illustrate both the input signals $\mathbf{x} = (x_1, x_2)$ and the readout produced by the network $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2)$, as two points in signal space (***Figure 2B***). Now let us imagine that one of the neurons, say neuron i, spikes. When that happens, the spike causes a jump in its filtered output spike train. In turn, and according to equation 1, the vector of readouts $\hat{\mathbf{x}}$ jumps in the direction $\mathbf{D}_i = (D_{1i}, D_{2i})$, as illustrated in ***Figure 2B***. Since the direction and magnitude of this jump are determined by the fixed readout weights, they are independent of the past spike history or the current values of the readouts. After this jump, and until another neuron fires, all components of the readout $\hat{\mathbf{x}}$ will decay. Geometrically, this decay corresponds to a movement of the readout towards the origin of the coordinate system.

The second assumption of *SCN*s is that a neuron spikes only when its spike moves the readout closer to the desired signal $\mathbf{x}$. For each neuron, this spike rule divides the whole signal space into two regions: a 'spike' half-space where the readout error decreases if the neuron spikes, and a 'no-spike' half-space where the readout error increases if the

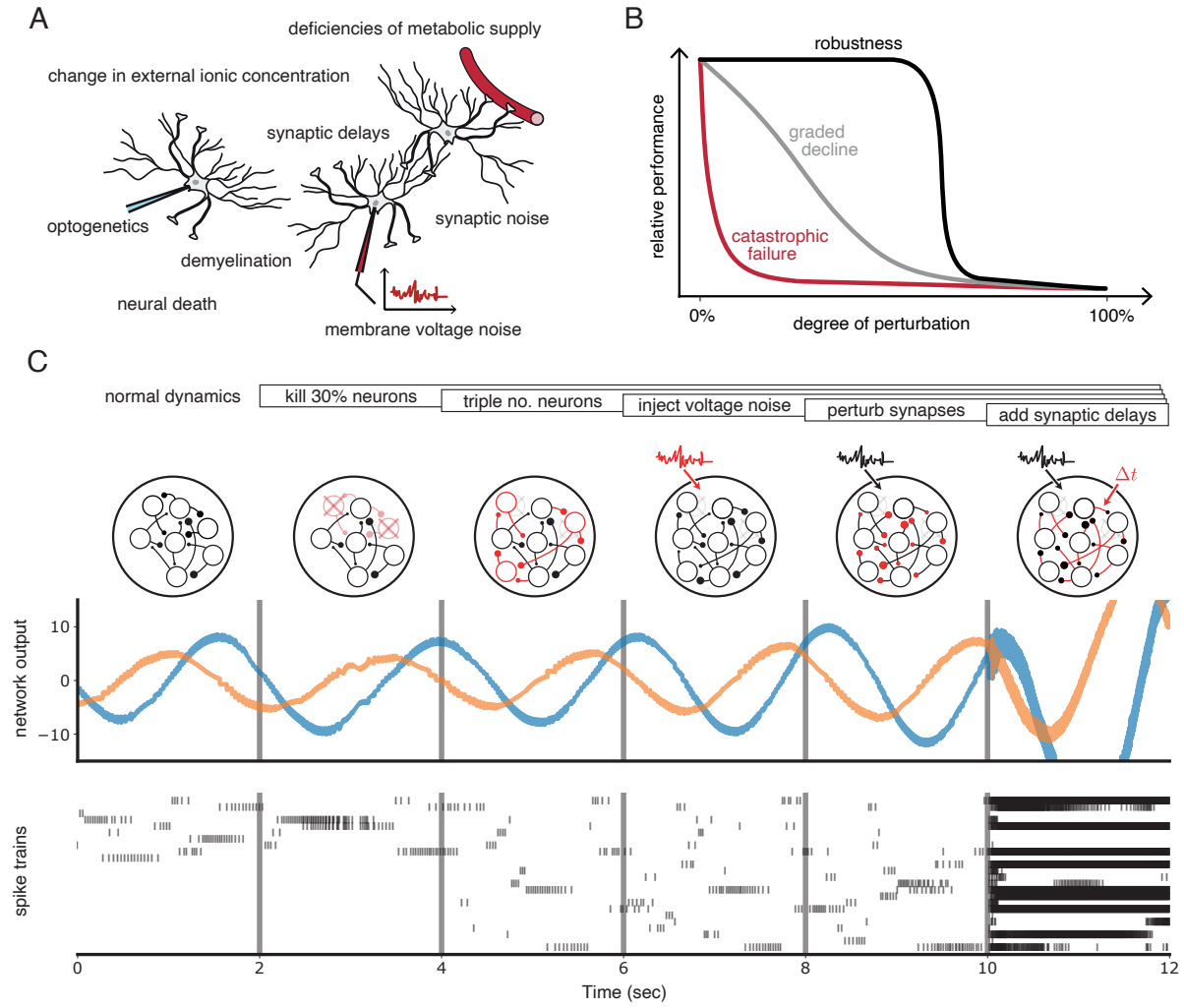

Figure 1: Neural systems are robust against a variety of perturbations. (**A**) Biological neural networks operate under multiple perturbations. (**B**) The degree of robustness of a system can fall into three regimes: 1. Catastrophic failure (red), when small changes in the conditions lead to quick loss of function for the system. 2. Gradual degradation (grey), when the system's performance is gradually lost when departing from optimal conditions. 3. Robust operation (black), when the network is able to maintain its function for a range of perturbations. (**C**) The output of a spike coding network, designed to generate a two-dimensional oscillation, is robust to several cumulative perturbations, breaking down only with the final introduction of synaptic delays. *Top*: Schematic of the various perturbations. Vertical lines indicate when a new perturbation is added. The standard deviation of the injected voltage noise is more than 5% of the neuronal threshold magnitude. The perturbation of all synaptic weights is random and limited to 5%. The synaptic delays are changed by 1 ms. *Middle*: Two-dimensional output, as decoded from the network activity. *Bottom*: Raster plot of the network's spike trains.

neuron spikes (***Figure 2B***). The boundary between these two half spaces is the neuron's spiking threshold, as seen in signal space. Consequently, the neuron's voltage V_i must be at threshold T_i, whenever the readout reaches this boundary, and the voltage must be below or above threshold on either side of it. We therefore identify the neuron's voltage with the geometric projection of the readout error onto the decoding vector of the neuron,

$$V_i = \mathbf{D}_i^\top (\mathbf{x} - \hat{\mathbf{x}}), \tag{2}$$

where, without loss of generality, we have assumed that $\mathbf{D}_i$ has unit length (see Material and Methods). The effect of this definition is illustrated in ***Figure 2E***, where the voltage increases or decreases with distance to the boundary. Accordingly, the voltage has a clear functional interpretation in terms of an error, given here by the distance of the readout to the neuron's boundary.

In addition to its functional interpretation, the voltage equation has a simple biophysical interpretation, as illustrated in ***Figure 2C***. Here, the two input signals, x_1 and x_2, get weighted by two synaptic weights, D_{1i} and D_{2i}, leading to two postsynaptic voltages that are then summed in the dendritic tree of neuron i. At the same time, the two readouts, $\hat{x}_1$ and $\hat{x}_2$, are fed back into the neuron via two exactly opposite synaptic weights, $-D_{1i}$ and $-D_{2i}$, thereby giving rise to the required subtraction. As a consequence, the neuron's voltage becomes the projection of the readout error, as prescribed above. When the neuron's voltage reaches the voltage threshold T_i, the neuron fires a spike, which changes the readout $\hat{\mathbf{x}}$. In turn, this change is fed back into the neuron's dendritic tree and leads to an effective reset of the voltage after a spike, as shown in ***Figure 2D***. Given that the decoding vectors are of length one, the optimal size of the threshold is given by $T_i = 1/2$ (see Material and Methods).

One neuron alone can only improve the readout along one specific direction in signal space and thus cannot correct the readout for all possible input signals (***Figure 2D***, arrow). In a network where each neuron contributes differently to the readout, the error will be corrected along different directions in signal space. A second neuron, say neuron j, is added in ***Figure 2F–H***. Following the logic above, its voltage is given by $V_j = \mathbf{D}_j^\top (\mathbf{x} - \hat{\mathbf{x}})$, and the respective voltage isoclines are shown in ***Figure 2H***. We see that the voltage of neuron j jumps when neuron i spikes. Mathematically, the size of this jump is simply given by the dot product of the two decoding vectors, $\mathbf{D}_j^\top \mathbf{D}_i$. Biophysically, such a jump could be caused by negative feedback through the readout units, but it could also arise through a direct synaptic connection between the two neurons, in which case $\Omega_{ji} = -\mathbf{D}_j^\top \mathbf{D}_i$ corresponds to the synaptic weight from neuron i to neuron j.

Finally, if we add three more neurons, and give them different sets of decoding weights, the network as a whole can restrict the readout to a bounded region in signal space (a polygon in two dimensions), as shown in ***Figure 2I–K***. We will call this bounded region the 'error bounding box' or simply the 'bounding box.' Its overall size determines the error tolerance of the network. To highlight the structure of this network, we can change Eq. 2 by inserting the definition of the readout, Eq. 1, to obtain

$$V_i = \mathbf{D}_i^\top \mathbf{x} - \sum_{k=1}^{N} \mathbf{D}_i^\top \mathbf{D}_k r_k. \tag{3}$$

Here, the term $\Omega_{ik} = -\mathbf{D}_i^\top \mathbf{D}_k$ can be interpreted as a lateral connection between neurons i and k in the network (***Figure 2I***). The diagonal elements of the respective connectivity matrix, Ω_{ii}, can be interpreted as the hyperpolarisation of the membrane voltage following a spike. Consequently, there is no need to compute the linear readout in a downstream layer, and then insert it via negative feedback back into the network. Rather, this negative feedback can be relayed through lateral connections and self-resets (***Figure 2I***; see also Material and Methods). While the connectivity of our network is symmetric, this assumption can be relaxed, as explained in the Material and Methods (see also Brendel et al. [25]).

As shown previously, the temporal derivative of the above equation yields a network of current-based, leaky integrate-and-fire neurons (see Material and Methods). We emphasize that there are two distinct situations that cause neurons to emit spikes. First, the readout always leaks towards the origin, and when it hits one of the boundaries, the appropriate neuron fires and resets the readout into the centre of the bounding box. Second, any change in the signal $\mathbf{x}$ causes a shift in the whole bounding box, since the signal is always at the centre of the box. A sudden shift may therefore cause the readout to fall outside of the box, in which case neurons whose boundaries have been crossed will fire to get the readout back into the box. We strongly encourage the reader to view ***Supplementary Video 1*** for an animation of the operation of SCNs.

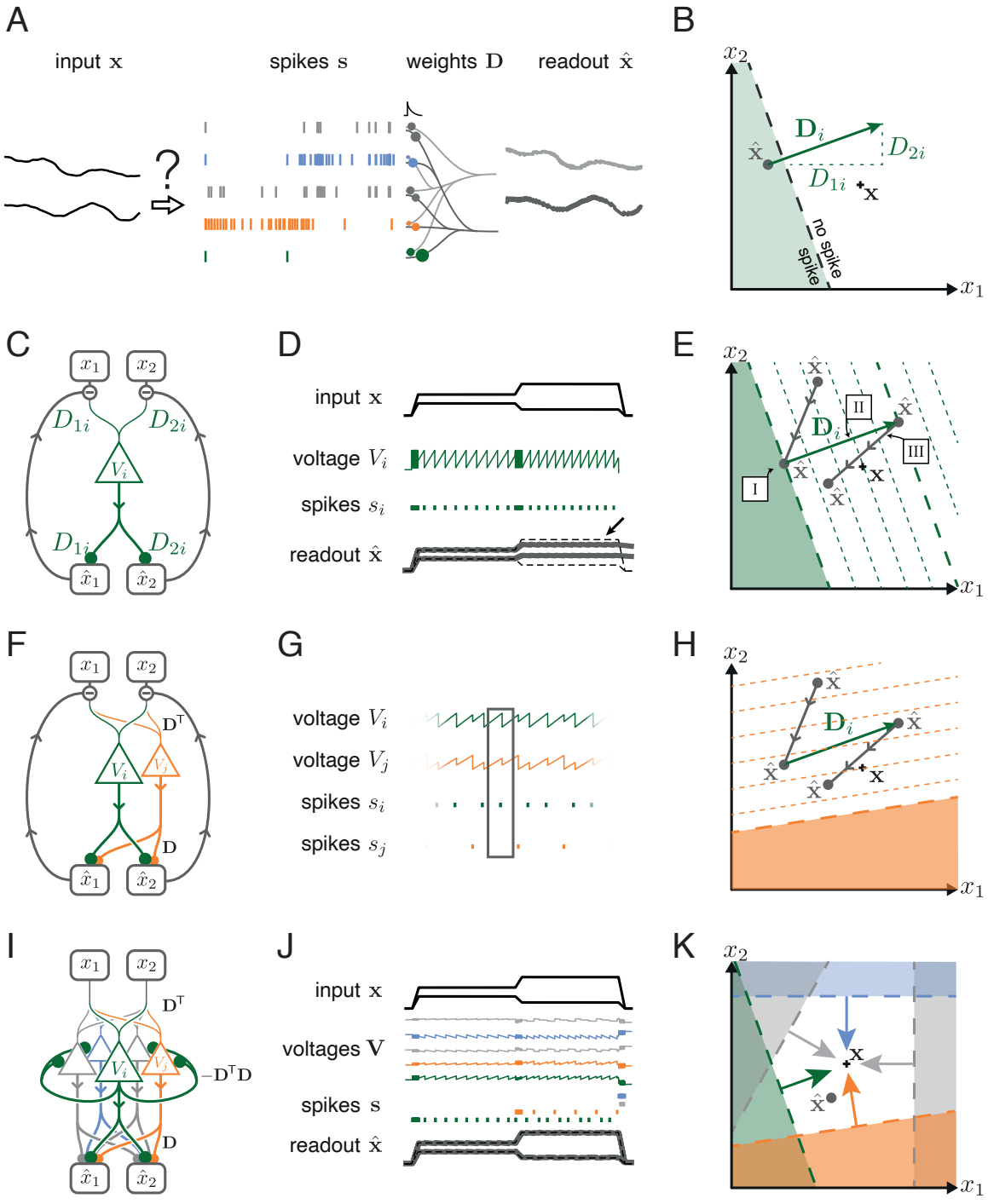

Figure 2: Spike coding networks (*SCN*s) operate by creating an error bounding box around the input signals. Here we construct a toy example with two inputs and five neurons. (**A**) The task of the network is to encode two input signals (black) into spike trains (coloured), such that the two signals can be reconstructed by filtering the spike trains postsynaptically (with an exponential kernel), and weighting and summing them with a decoding weight matrix **D**. (**B**) A neuron's spike moves the readout in a direction determined by its vector of decoding weights. When the readout is in the 'spike' region, then a spike from the neuron decreases the signal reconstruction error. Outside of this region ('no spike' region), a spike would increase the error and therefore be detrimental. *(continued on following page)*

Figure 2, continued: (**C**) Schematic diagram of one neuron. The neuron's voltage measures the difference between the weighted input signals and weighted readouts. (**D**) Simulation of one neuron tracking the inputs. As one neuron can only monitor a single error direction, the reconstructed signal does not correctly track the full two-dimensional signal (arrow). (**E**) Voltage of the neuron (green) and example trajectory of the readout (gray). The dashed green lines correspond to points in space for which neuron i has the same voltage (voltage isoclines). The example trajectory shows the decay of the readout until the threshold is reached (I), the jump caused by the firing of a spike (II), and the subsequent decay (III). (**F**) Same as C, but considering two different neurons. (**G**) Voltages and spikes of the two neurons. (**H**) Voltage of the orange neuron during the same example trajectory as in E. Note that the neuron's voltage jumps during the firing of the spike from the green neuron. (**I**) The negative feedback of the readout can be equivalently implemented through lateral connectivity with a weight matrix $\mathbf{\Omega} = -\mathbf{D}^T\mathbf{D}$. (**J**) Simulation of five neurons tracking the inputs. Neurons coordinate their spiking such that the readout units can reconstruct the input signals up to a precision given by the size of the error bounding box. (**K**) The network creates an error bounding box around $\mathbf{x}$. Whenever the network estimate $\hat{\mathbf{x}}$ hits an edge of the box, the corresponding neuron emits a spike pushing the readout estimate back inside the box (coloured arrows).

The geometry of the bounding box in higher dimensions

While the simple toy example in ***Figure 2*** is useful to illustrate some of the key features of SCNs, biological neural networks, and especially cortical networks, consist of thousands of neurons that are thought to represent hundreds of signals simultaneously. To get closer to the biological reality, we therefore need to study larger and more powerful networks. Many of the biological features of larger SCNs depend crucially on how the shape of the bounding box changes with the number of neurons N, and the dimensionality of the input signals M. For simplicity, we will assume that the decoding vectors of the neurons $\mathbf{D}_i$ are of unit length, but otherwise random, and that the thresholds of all neurons are the same (see Material and Methods for details on the parameter choices).

The number of input signals M determines the dimensionality of both the signal space and the corresponding bounding box. For a two-dimensional signal, the threshold of each neuron corresponds to a line, and the bounding box to a polygon, as illustrated in ***Figure 2*** and ***Figure 3A***, and in ***Supplementary Video 1***. For a three-dimensional signal, the threshold of each neuron corresponds to a plane, and the bounding box consequently to a polyhedron (***Figure 3A*** and ***Supplementary Video 1***). For higher-dimensional signals, though hard to visualise, bounding boxes are convex polytopes.

The number of neurons N corresponds to the number of sides of the bounding box, which are also known as 'faces' in three or more dimensions. When we increase the number of neurons (or randomly oriented faces), we are adding faces to the bounding box, which thereby changes its shape. As we keep adding neurons, the corresponding bounding box eventually approaches a hypersphere (a circle in two dimensions and a sphere in three dimensions), as shown in ***Figure 3B***, lower row. However, the number of neurons required to reach a decent approximation of the hypersphere grows exponentially with the number of dimensions, so that $N \sim 10^M$. Given the number of neurons in the human brain ($N \sim 10^{11}$), we could at most represent 11 signals under these circumstances.

To be able to encode higher-dimensional signals, we therefore need to introduce sub-exponential scaling. For simplicity, we will scale the number of neurons linearly with the number of signal dimensions, $N = \rho M$, where ρ defines the network redundancy. To characterize how the shape of the bounding boxes changes as we increase the dimensionality, we can compute the angles between neighbouring faces, $\gamma = \arccos(\mathbf{D}_i^\top \mathbf{D}_j)$. Since we assume random (and uncorrelated) decoding weights, their inner products, $\mathbf{D}_i^\top \mathbf{D}_j = -\Omega_{ij}$ will reach zero as the signal dimensionality grows, and the angles will approach 90° (***Figure 3D***). Accordingly, bounding boxes in high-dimensional spaces are more similar to hypercubes than hyperspheres (***Figure 3B–D***).

Although these results were obtained for random decoding vectors, the key insights hold for more structured decoding vectors as well. For instance, if we want to represent natural visual scenes, we may consider that the receptive fields of simple cells in V1 roughly correspond to the decoding vectors of our neurons [23, 26]. If we choose a set of (random) Gabor patches of size 13×13 for these decoding vectors, we again find that the corresponding bounding box is more similar to a hypercube than a hypersphere: for a given Gabor patch, almost all other Gabor patches are orthogonal, and only a select few are somewhat similar (***Figure 3E***).

As we show below, these properties of high-dimensional spaces will have a strong influence on how *SCN*s respond

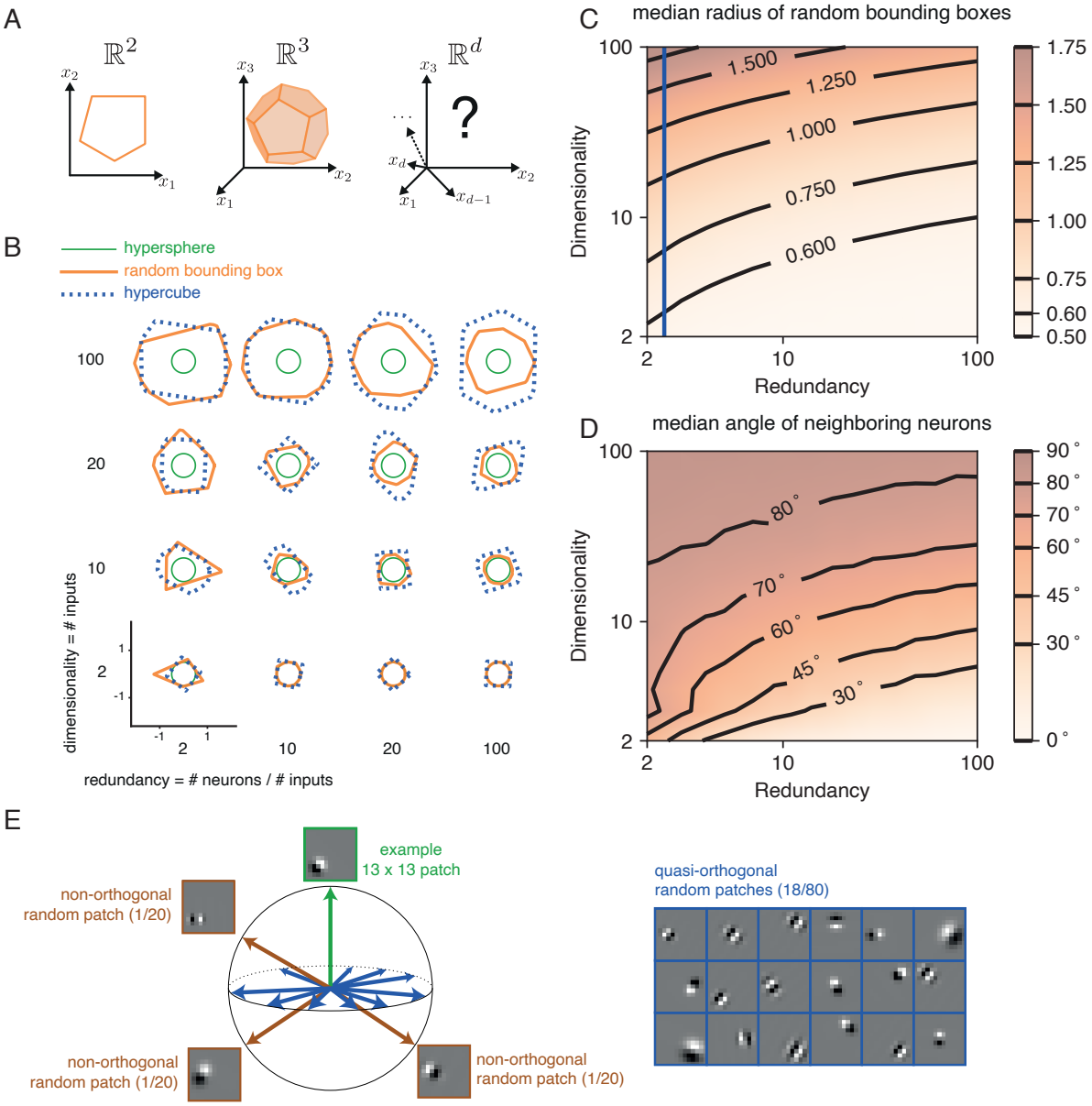

Figure 3: The geometry of the bounding box changes with input dimensionality and redundancy. (**A**) In *SCN*s tracking two-dimensional signals, the bounding box is geometrically depicted as a polygon with as many sides as the number of neurons. For three dimensional systems, the bounding box corresponds to a polyhedron. For four or more dimensions, the corresponding bounding boxes are mathematically described as convex polytopes, but their visualisation is hard. (**B**) Example two-dimensional cuts of bounding boxes (orange) for a given network size and space dimensionality (Material and Methods). Cuts for a hypersphere (green) and a hypercube (dashed blue) are shown for comparison. For low dimensionality, high redundancy bounding boxes are similar to hyperspheres whereas for high dimensionality they are more similar to hypercubes. (**C**) Median radius of bounding boxes as a function of dimensionality and redundancy. The blue line illustrates the average radius of a hypercube (thresholds of individual neurons are here set at T=0.5). (**D**) Median angle between neighbouring neurons, i.e., neurons that share an "edge" in the bounding box. Neighbouring neurons in high dimensional signal spaces are almost orthogonal to each other (**E**) Random 13x13 Gabor Patches representing the readout weights of neurons in a high dimensional space. Most Gabor patches are quasi-orthogonal to each other (angles within $90 \pm 5°$). Some neurons have overlapping receptive fields and non-orthogonal orientations.

to perturbations.

Baseline performance and spiking statistics

Before studying the network's response to perturbations, we will first establish several characteristics of the unperturbed networks, which will act as a baseline. See Material and Methods for a detailed explanation of how we chose to scale the networks with signal dimensionality and number of neurons.

The first characteristic is the performance. As explained above, the bounding box sets the limit of how far the output (or linear readout) is allowed to deviate from the inputs. To illustrate these limits in practice, we simulated a set of random, time-varying input signals, and then accumulated the decoding errors along each signal dimension into a large histogram, shown in ***Figure 4A***. As expected, and by design, the decoding errors stay roughly within the same range. Beyond that, we see two more subtle effects. First, the decoding errors for higher-dimensional networks are smaller than the decoding errors for lower-dimensional networks (***Figure 4B***). Since the input signals are chosen randomly from Gaussian distributions, the number of weak signals grows with dimensionality, leading to the slight shift in the distribution. Second, the decoding errors for more redundant networks are slightly smaller than those for less redundant networks. Since more redundant networks are slightly closer to a hypersphere, they provide a somewhat tighter bound of the errors for fixed neuronal thresholds. We emphasize that these comparisons are done under the assumption that the length of the decoding vectors is normalised to one. Obviously, the performance of any bounding box can be adjusted to a desired level by simply changing this normalisation factor.

The second characteristic are the firing rates. We find that if the network receives a constant stimulus, then the distribution of firing rates is long-tailed (and roughly log-normal, see ***Figure 4C***), as has been observed in many brain areas [27], and can be found in randomly connected networks [28]. Beyond that, we see that the median firing rates of the networks drop with increasing redundancy (***Figure 4D***). Since an increase in redundancy corresponds to the addition of faces in the bounding box, the individual faces (or neurons) need to cover less overall space, and thereby get hit less often, so that overall firing rates decrease.

The third characteristic that we will study are coefficients of variation (CVs) which serve to measure the irregularity of spike trains (see Material and Methods). For lower redundancies, $\rho < 4$, we find low CVs, and for higher redundancies, $\rho > 4$, we find CVs close to one, which corresponds to random firing, similar to Poisson spike trains (***Figure 4E,F***). When the network has fewer neurons, it has less degeneracy and the number of spiking patterns, that can approximate the signal, decreases. As a consequence, the spike patterns of individual neurons become more predictable and more regular.

Neural death and birth

We will now use these geometric intuitions to study the robustness of *SCN*s to different types of perturbations. We will start with neuronal loss or death. Throughout an organism's life, cells, including neurons, can undergo the process of cell death or apoptosis if they are damaged or unfit [29], a process that is usually promoted in diseased states [30–32]. Biological tissue, including nervous tissue, is often resilient against this type of perturbation.

Previous work has shown that *SCN*s are remarkably robust to the removal of neurons [23]. When too many neurons have died, *SCN*s cross a 'recovery boundary' after which functionality declines rapidly. By studying the network's behavior through the lens of the bounding box, we can provide a simple and intuitive explanation for these results. Geometrically, the death of a neuron is equivalent to the removal of its corresponding face from the bounding box (***Supplementary Video 2***, ***Figure 5A***). However, if the network is sufficiently redundant, then the removal of a single neuron has only a minor impact on the shape of the bounding box. Since this shape determines the network's error tolerance, the network will continue to encode the input signals correctly, despite the loss of a neuron. Naturally, the higher the redundancy ρ, the higher the resilience of the network to random neural death. However, any *SCN*, independent of the redundancy, loses its functionality when the bounding box breaks open on one side (***Supplementary Video 2***, and ***Figure 5A***, last panel). Such an opening occurs when a complete set of similarly tuned neurons has been eliminated.

In addition to neural death, many neural circuits are subject to neural birth, i.e. neurogenesis, both in developing and adult animals. If we imagine that a single neuron is added to the network, and if we further imagine that its synapses have been properly adjusted (e.g. through silent learning with voltage-dependent plasticity [25]), then adding that neuron corresponds to adding an extra face to the bounding box (***Figure 5B***). Adding neurons thereby increases

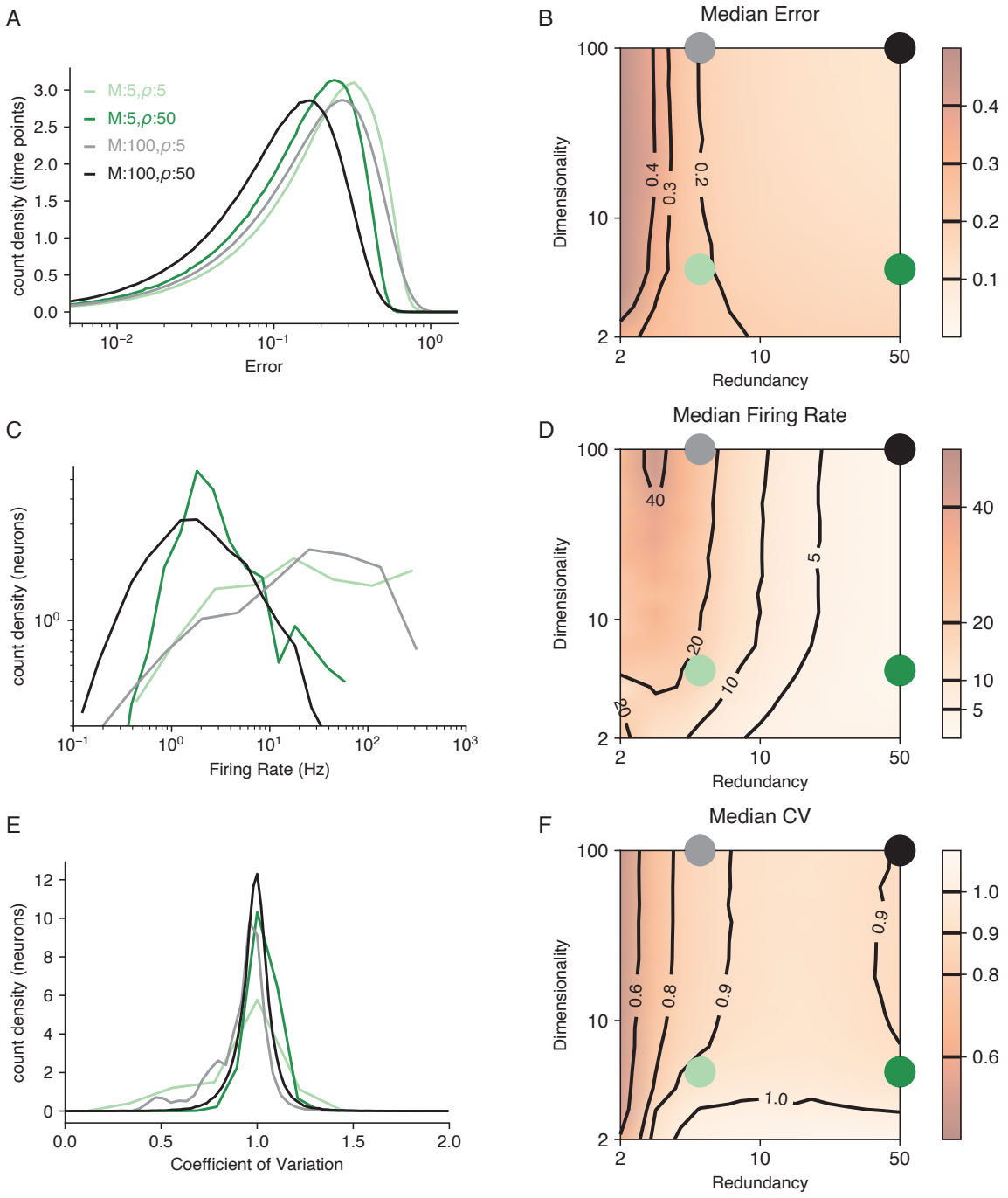

Figure 4: Errors, firing rates, and CVs as a function of network redundancy and input dimensionality. (**A**, **C** and **E**) Four different example networks were simulated with the same fixed input (and slow-varying input noise, see Material and Methods) on multiple trials and its resulting distribution plotted. Colours match the dots in the subsequent panels. (**B**, **D** and **F**) Median of these distributions as a function of redundancy and input dimensionality. Even for small network sizes, CVs are already close to one, corresponding to Poisson spike trains.

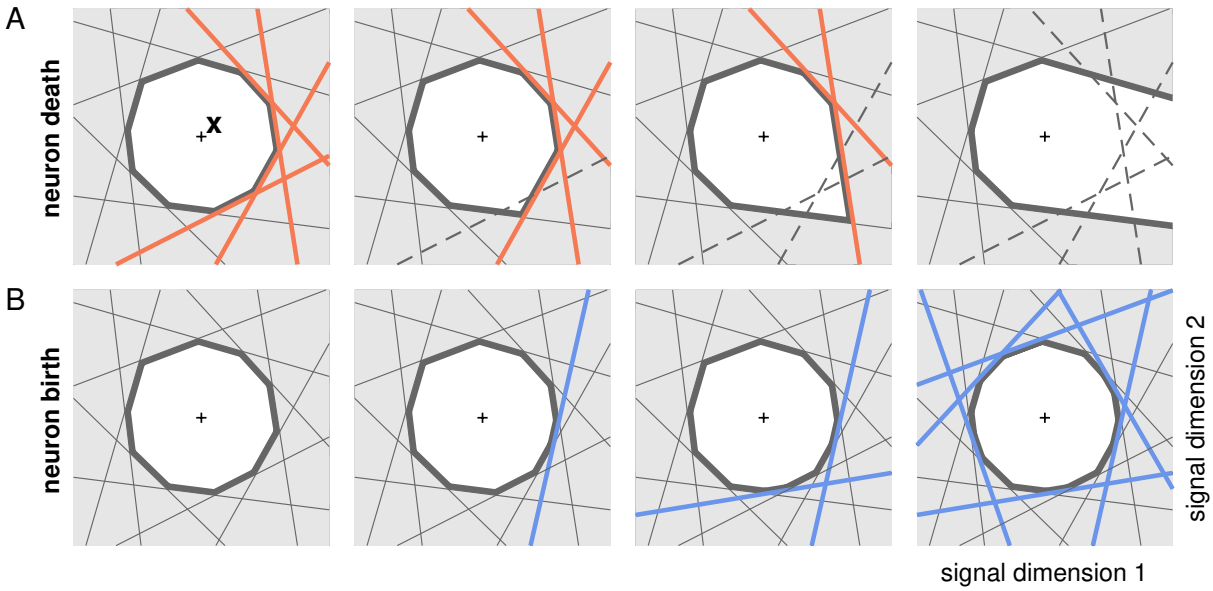

Figure 5: The effect of neural death or birth. Neuron death (birth) is geometrically equivalent to removing (adding) the corresponding bounds on the bounding box. (**A**) Left, bounding box with four neurons highlighted. Middle, when two of the highlighted neurons are eliminated, the bounding box remains closed, and the error remains bounded. Right, when all four neurons are eliminated, the bounding box breaks open, and the error is no longer bounded in the respective direction. (**B**) Bounding box, followed by the addition of random neurons. Additional neurons only marginally reduce box size and, accordingly, the maximum coding error

the redundancy ρ of the system and improves the system's robustness. In turn, subtracting neurons decreases redundancy ρ, and brings the system closer to the recovery boundary.

The death or birth of random neurons therefore simply correspond to a change in the overall redundancy of the network. Consequently, to understand how network performance and statistics change, we can simply look at ***Figure 4B,D,F***, and observe what happens when we change the redundancy. We observe that changing the redundancy over a broad range has negligible effects on the performance (***Figure 4B***). However, decreasing redundancy (neuron death) leads to higher firing rates (***Figure 4D***) and lower CVs (***Figure 4F***).

Thresholds

Biological systems should also be robust against the mistuning of any of their components. We will now show that many types of parameter mistuning can be understood as deformations of the bounding box. We order the exposition by the complexity of the effects, and start with the simplest effect, caused by perturbations in the neuronal spiking threshold. While the actual spiking threshold of a cell depends on both conductances and reversal potentials, we will treat it here as a simple parameter.

Since a neuron's voltage is a projection of the coding error, its spiking threshold sets its error tolerance (***Figure 2B***). Consequently, an increase of a neuron's spiking threshold will push the corresponding face of the bounding box outwards, (***Figure 6A***, first panel). For a sufficiently redundant system, increasing the threshold of a single neuron will have a very minor effect on the shape of the bounding box. In fact, a large increase of the threshold eventually entails an effective loss of that neuron to the circuit, which we studied above (***Figure 5***). If all thresholds increase, then the bounding box becomes wider, which increases the error tolerance of the system.

On the other hand, a decrease in a neuron's spiking threshold will push the corresponding face inwards, thereby shrinking the bounding box from one side (***Figure 6A***, second panel). As a consequence, the corresponding neuron will take up the load of all of the neurons that are now hidden, firing more and more. Eventually, the neuron's spikes may reset the readout beyond the boundary of the opposite side, thereby crossing the threshold(s) of one (or more) neurons on the opposite side, and causing them to spike. In turn, these double-threshold crossings can lead to fast

firing of oppositely tuned neurons (***Figure 6A***, third panel), which has previously been termed the 'ping-pong' effect [19]. While the system may remain functional in that case (the readout could still be correct), it will generate a lot more spikes than necessary. If the thresholds of many neurons are decreased, then the sudden surge of energetic needs could lead to system failure in real, biological systems. We will therefore generally assume that ping-pong denotes system failure.

The onset of system failure will depend on the initial, 'default' threshold values. Throughout this manuscript, we will therefore often consider two scenarios, a 'narrow box' with a threshold between $T = 0.50$ and $T = 0.55$ and a 'wide box' with a threshold between $T = 0.7$ and $T = 1.5$ (***Figure 6–Figure Supplement 1***). In addition to added protection against catastrophic failure, a wide box (e.g. with $T = 1$) can be mistuned in its threshold parameters by up to 50% of their value without affecting the network's functionality. In contrast, in a narrow box (e.g. with $T = 0.55$), threshold parameters must be tuned to within 10% of their optimal value to keep the network in its functional range.

Voltage noise

Biological systems are constantly subject to noise at multiple levels, e.g. sensory transduction noise, ion channel noise [33] or 'background' synaptic activity [34, 35]. Here we study the impact of such noise by injecting small, random currents into each neuron. Due to the voltage leak, the white-noise current becomes (coloured) voltage noise, which we can add to the voltage equation, ***Equation 3***. This voltage noise changes how close the voltage of a neuron is to its spiking threshold. With regard to spike generation, these voltage fluctuations are thus equivalent to fluctuations of the threshold (see Material and Methods). In the above section, we have already shown that changes to a neuron's threshold correspond to movements of the corresponding face in the bounding box. Accordingly, fluctuations of the thresholds are equivalent to independent, random movements of all of the faces of the bounding box around their unperturbed positions (see ***Supplementary Video 2***).

For networks with low redundancy ρ, small voltage fluctuations cause only minor deformations of the bounding box – here, 'small' is measured relative to a neuron's operating regime, from reset to threshold. In turn, the error tolerance remains roughly the same, and network performance is not affected (***Figure 6D***). Even if voltage noise is very small, however, it can still have a dramatic effect on the spike trains of individual neurons. When trials are repeated, these spike trains can show high trial-to-trial variability (***Figure 6F***). Even small levels of voltage noise get therefore amplified at the level of spike trains, but not at the level of readouts, as previously observed in Boerlin et al. [19].

For networks with high redundancy, ρ, small voltage fluctuations can cause a fatal collapse of the system. The key reason is that the effective size of the bounding box is not determined by the unperturbed positions of the thresholds, but by the position of the thresholds that have moved furthest into the box. As more and more neurons are added, the likelihood that some of them have very decreased thresholds increases, and the effective size of the bounding box shrinks (***Figure 6B***, left three panels). In turn, the probability that the network moves into an 'epileptic seizure' (due to the 'ping-pong' effect) increases as well. Ultimately, random movement of the bounds may cause a collapse of the box, in which case neurons fire uncontrollably (***Figure 6C***, second and third panels). While the readouts may still be contained under such 'epileptic seizures' (***Figure 6D***), the excessive number of spikes fired (***Figure 6E***) come at high metabolic costs and would be detrimental to biological systems.

To avoid this failure mode, one can simply increase the size of the bounding box for a fixed redundancy (***Figure 6B***, right panel). Such a 'wide box' will be more resilient towards noise (***Figure 6C***, right panel, ***Figure 6D–F***). However, no matter how wide the box, there will always be a level of redundancy at which the system collapses. In this system, more redundancy does therefore not necessarily lead to higher robustness.

The described effects of noise on *SCN*s are independent of the signal dimensionality (***Figure 6–Figure Supplement 1***). Unsurprisingly, higher noise levels increase the variability of single neuron spiking, an effect which is amplified for larger networks (***Figure 6–Figure Supplement 2***). When these variable but long interspike intervals are mixed with rapid bursts of short-interval ping-pong spikes, the overall coefficient of variation strongly increases. (***Figure 6–Figure Supplement 1***).

Resets

Next, we will study perturbations of a neuron's reset potential, i.e., the voltage reached directly after a spike. In *SCN*s, this voltage should ideally be $V_{i,\text{reset}} = T_i - \mathbf{D}_i^\top \mathbf{D}_i$ (see Material and Methods). A decrease of this default reset voltage can be interpreted as a quadratic cost on neural firing [19], which distributes spiking across similarly tuned neurons.

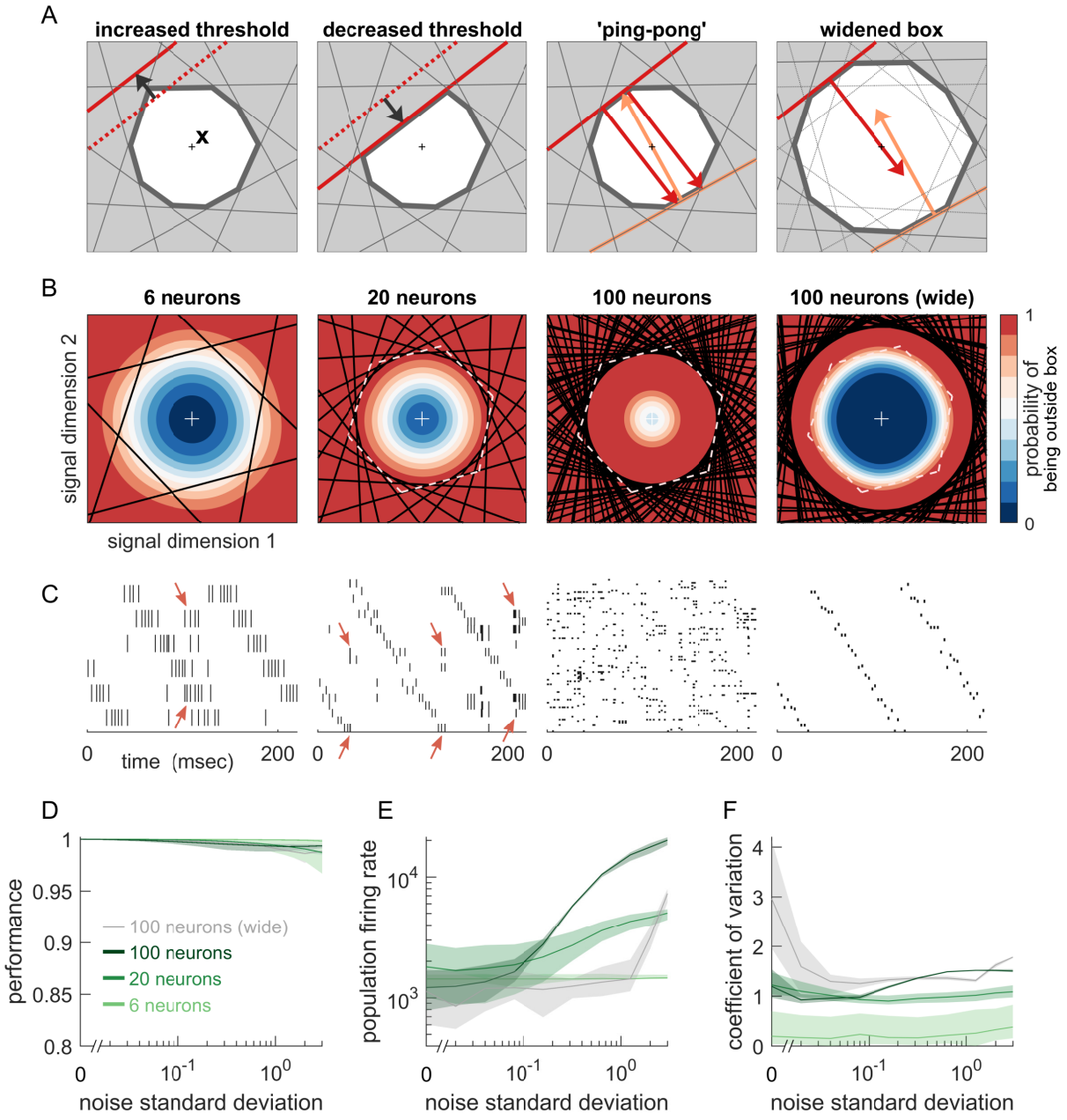

Figure 6: Voltage or threshold noise induces fluctuations of the bounding box shape. (**A**) (left) If a neuron's threshold increases beyond its default value, its respective boundary moves outwards. (centre left) If the threshold decreases below its default value, the boundary moves inward (centre right). A shrunk box can trigger a spike that pushes the readout beyond the boundary of an oppositely tuned neuron, leading to a compensatory spike. Fired in rapid succession, a barrage of mutually opposed spikes may follow: the 'ping-pong' effect. (right) If the default values of all thresholds are increased, the box becomes wider and more robust against ping-pong. *(continued on following page)*

Figure 6, continued: (**B**) Voltage noise can be visualised as jittery movement of all thresholds. Instead of a rigid box defining a permanent, unambiguous boundary between the spike and no-spike zones, any point in signal space now has a non-zero probability of falling outside the box, shown in colour. Black lines represent the thresholds of individual neurons in the absence of noise. (left) At low redundancy, most points within the default box retain a low probability of exclusion. (centre left, centre right) As redundancy increases, this low-probability volume disappears, increasing the likelihood of ping-pong spikes. (right) Networks with an expanded bounding box retain a large low-probability volume even at high redundancy. Dashed white lines show 6-neuron bounding box for comparison. (**C**) Raster plots for the networks in (B) when tracking two oscillatory signals. See also ***Figure 6–Figure Supplement 2***. Arrows highlight examples of ping-pong spiking (left two panels). Ping-pong becomes dominant in the more redundant network in the third panel, but is not highlighted. Ping-pong spiking can be eliminated by widening the box (fourth panel). (**D**) When noise level increases, performance (relative to a network without noise, see Material and Methods) drops only slightly. (**E**) The ping-pong effect causes numerous unnecessary spikes for higher levels of noise, with more redundant networks affected more strongly. (**F**) CVs are largely unaffected by increases of the noise level. Note that an expanded box with low noise level shows a bimodal distribution of single-neuron interspike intervals (intervals within an up state and intervals between two up states – see (C)), leading to particularly large CVs. (**D-F**) In each case, networks with an expanded box retain healthy dynamics until much higher noise levels. Lines show medians across random equidistant networks, and outlines represent interquartile ranges. All colours as in (D). (**A-F**) Thresholds are 0.7 for the 'wide' box and 0.5 otherwise.

Figure 6–Figure supplement 1. Robustness to noise for different signal dimensionalities.

Figure 6–Figure supplement 2. Spike trains and decoded signals with voltage noise.

Biophysically, when the neuron resets to a voltage above (below) this ideal reset potential, then its post-spike voltage is temporarily closer (further) from threshold. In terms of the neuron's spiking output, a change in its reset voltage is therefore equivalent to a (temporary) change in its threshold. As before, we can therefore illustrate perturbations of the voltage resets by their equivalent effect on the thresholds (and thereby the bounding box) of the network.

The effect on the bounding box is shown in ***Figure 7A***. Here, we see that a reset voltage below the optimal reset will initially lead to a push of the neuron's threshold outwards. However, because of the voltage leak, the threshold will then decay back to its normal position. The opposite effect holds for a reset voltage above the optimal reset. ***Supplementary Video 2*** illustrates this effect in a system with a two-dimensional input.

Synaptic noise

Synapses have been shown to have multiple sources of variability [33], such as a variable number of neurotransmitters in a vesicle or the diffusion process of vesicles in the synaptic cleft. Such noise sources can lead to spontaneous or variable postsynaptic currents during synaptic transmission. In order to study these perturbations, we will first study the mistuning of a single synapse from its optimal value, $\Omega_{ij} = -\mathbf{D}_i^\top \mathbf{D}_j$. If the respective synapse becomes too small, then the induced voltage jump in the postsynaptic neuron will be too small. For an inhibitory synapse, the postsynaptic neuron will therefore remain closer to threshold than it should have. As before, we can illustrate this perturbation as an inward move of the respective threshold (***Figure 7B***). Accordingly, each mistuning of a synapse causes a temporary change in the threshold of the postsynaptic neuron whenever a presynaptic spike arrives. When all synapses in the network are randomly mistuned, then each spike fired will cause a random, but transient deformation of the bounding box (see ***Supplementary Video 2***).

Given these geometric insights, we see that small, but random perturbations of all the synapses in the network have a similar effect to the voltage noise we studied above, albeit on short time scales. If perturbations target only inhibitory or excitatory synapses, however, the deformations of the bounding box are no longer random. Specifically, strengthening inhibitory synapses or weakening excitatory synapses leads to a temporary widening of the box after a spike (***Figure 7C***), whereas weakening inhibitory synapses or strengthening excitatory synapses leads to a temporary shrinking of the box after a spike (***Figure 7D***).

In biological systems, we would furthermore expect that the size of possible perturbations scales with the strength of the synapses [33], so that weak synapses, e.g., can only be perturbed by small amounts. In other words, synaptic noise should be multiplicative and not additive. Accordingly, perturbations of stronger synapses lead to greater box

deformations. These stronger synapses are precisely the ones connecting neurons with similar (opposite) readout weights, since closely (oppositely) aligned neurons have stronger inhibitory (excitatory) synapses. In contrast, weak synapses, which happen between neurons with approximately orthogonal readout weights, are less impacted by such synaptic perturbations. Therefore, *SCN*s encoding higher dimensional signals, for which readout weights tend to be orthogonal (***Figure 3C***), are in principle more robust to random synaptic weights scaling. However, for fixed redundancy and increasing dimensionality, *SCN*s have linearly increasing neurons, and quadratically more synapses. Overall we found that these two opposite effects, i.e., signal dimensionality and number of neurons, cancel out and thus conclude that signal dimensionality does not qualitatively change the network response to such synaptic perturbations (***Figure 7E,F***). On the other hand, when signal dimensionality is fixed and network size is increased, the system becomes more susceptible to synaptic mistuning: similar to the impact of voltage noise in the performance of *SCN*s, the most mistuned synapses will dominate the dynamics, and lead to a collapse of the bounding box (***Figure 7G,H***). We note that if the size of the bounding box is increased, this effect can be alleviated and the network becomes more resilient to synaptic mistuning (***Figure 7G,H***).

We also considered other types of synaptic perturbations (***Figure 7–Figure Supplement 1***) such as time-varying synaptic noise, sparsification of the connectivity matrix, and temporary synaptic failure (see Material and Methods). Despite minor differences in their response properties as a function of signal dimensionality, we found a strong agreement on how *SCN*s respond to all types of synaptic perturbations as a function of redundancy. In these cases and as before, networks with more neurons (and consequently more synapses) are typically more vulnerable to these perturbations.

Synaptic delays

While the propagation of an action potential within a neuron and the subsequent synaptic transmission are fast, they are not instantaneous. Rather, lateral excitation and inhibition in biological neural networks may incur delays on the order of milliseconds. Like many other network models, *SCN*s do not by default take these delays into account, and instead assume nearly instantaneous exchange of information (within one simulation time step).

When we relax this assumption, the voltages of the neurons no longer reflect an accurate estimate of the collective coding error, but instead an imperfect estimate based on outdated information. When different neurons have identical decoding vectors, delays can lead to the firing of uninformed spikes that decrease the coding error erroneously (***Figure 8A,B***). With multiple identically tuned neurons, the delayed arrival of lateral inhibition from a single spike can enable many uninformed spikes at once. Once the resulting lateral excitation arrives at neurons with opposite tuning to those originally spiking, they may then react with a similar 'ping-pong' barrage of compensatory spikes, all but the first of which will be uninformed [36, 37]. We note that the only effect of refractory periods, rather than compensating for synaptic delays, is to cap the maximum number of uninformed spikes by limiting the firing rate of each neuron.

The picture becomes more complicated when neurons are not identically tuned (***Figure 8C–F***). ***Figure 8C*** shows the dynamics surrounding a single spike fired in a network with delayed synaptic transmission. When a neuron fires, it resets its own voltage immediately, but neither a hypothetical readout unit, nor the other neurons in the network are aware of the spike. From the point of view of the network, the voltage of the firing neuron is therefore temporarily too low (or its threshold temporarily too high), which we can visualise as an outward jump of its boundary (***Figure 8C***, second and third panels). When the spike finally arrives, the readout and voltages of all affected neurons are updated, and the voltage of the firing neuron agrees again with the network state, which we can visualise as the boundary coming back to its default position (***Figure 8C***, fourth panel).

Whether such a delayed spike is detrimental to network performance depends on the shape of the bounding box. In ***Figure 8C***, the delayed spike is not harmful, since the firing neuron is almost orthogonally tuned to its neighbouring neurons. The situation is different when the firing neuron is more similarly tuned to a neighbouring neuron (***Figure 8D***). In this case, during the delay from the firing of a spike to its arrival to postsynaptic neurons, a second neuron might cross its threshold, so that its boundary also retracts from its default position (***Figure 8D***, third panel). Eventually, the two spikes reach their postsynaptic neurons, the readout is updated, and the bounding box retracts to its original shape (***Figure 8D***, fourth panel). The readout can then overshoot and cross an opposite boundary, triggering further compensatory spikes, which again leads to 'ping-pong' spiking. In highly redundant networks, this scenario is essentially unavoidable.

To understand how the dimensionality of the bounding box interacts with synaptic delays, we first note that the

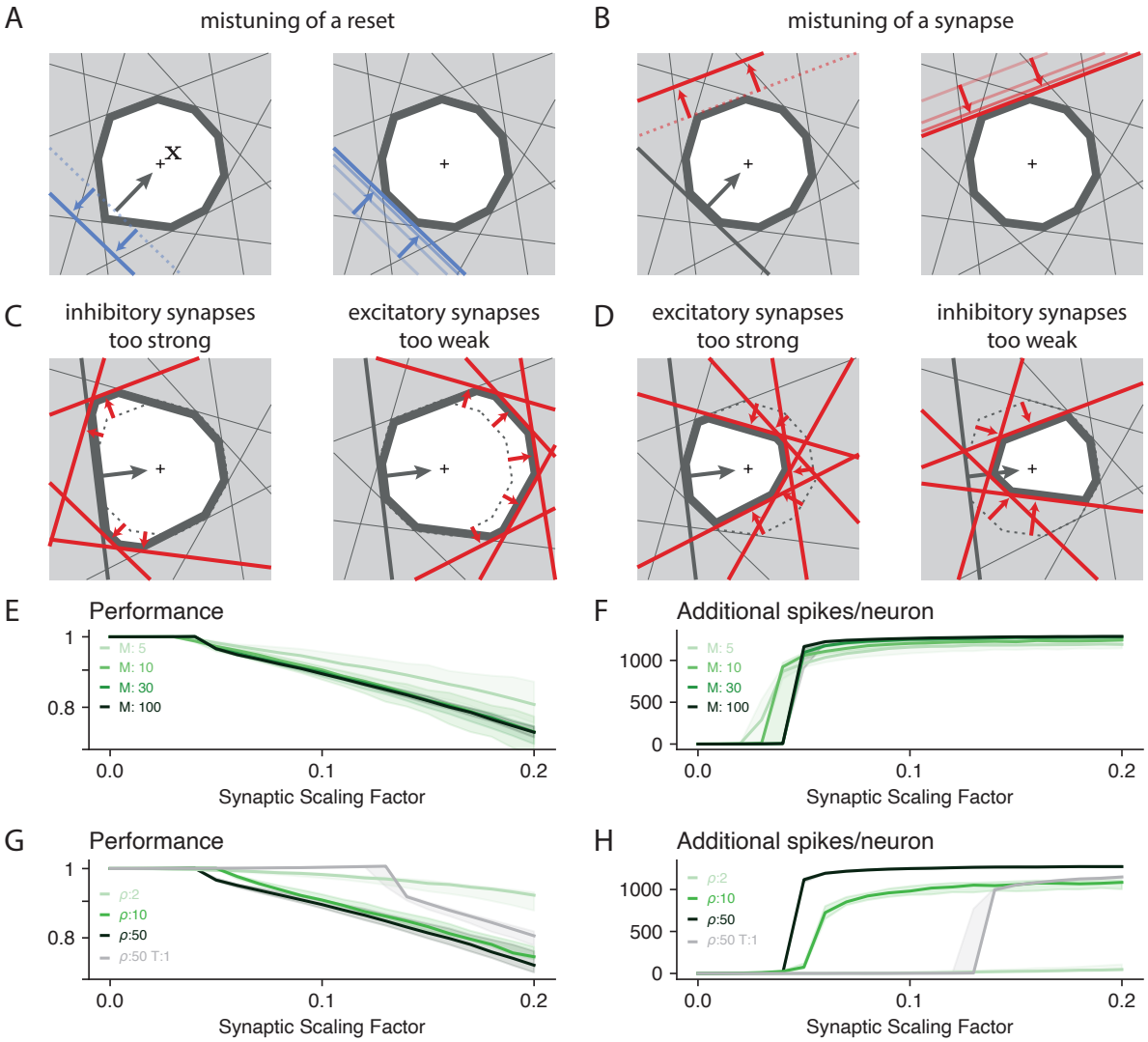

Figure 7: Resilience of networks to mistuning of resets or synaptic scaling. (**A**) Temporary bounding box deformation caused by a mistuned reset. The deformation appears after a spike of the affected neuron and decays away with the time constant of the voltage leak. (**B**) Temporary bounding box deformation caused by a mistuned synapse. The deformation appears after a spike of the presynaptic neuron and decays away with the same time constant. (**C**) Weakening excitatory synapses and potentiating inhibitory synapses cause a temporary expansion of the box after a spike, thus making the system less prone to firing instabilities. (**D**) The converse manipulations cause a temporary contraction of the box, potentially leading to uncontrolled firing. (**E-F**) Networks tracking higher-dimensional signals withstand slightly stronger synaptic mistuning before entering 'ping-pong' (F), but *SCN* performance (relative to an unperturbed *SCN*, see Material and Methods) degrades similarly across dimensionality (E). (**G-H**) Higher redundancy networks are more sensitive to synaptic mistuning. This extra sensitivity can be counteracted when the box is made wider.

Figure 7–Figure supplement 1. Robustness of *SCN*s for different types of synaptic noise.

angles of neighbouring neurons become more and more orthogonal as the number of signal dimensions is increased (***Figure 3D***). Accordingly, as we increase dimensionality, we should find ourselves more often in the scenario of ***Figure 3C***. Numerically, we find that, for very short delays (<0.1 msec), *SCN*s retain good performance since uninformed spikes are rare and ping-pong mostly absent (***Figure 8G***). With biologically plausible delays (1msec), however, *SCN*s suffer from drastic reductions in performance due to ubiquitous ping-pong. Accordingly, the bounding boxes are too tight to observe any beneficial effects of dimensionality.

As in the above described scenario of a noisy network (***Figure 6A***, fourth panel), widening the box can prevent networks from showing ping-pong (***Figure 8E***). Therefore, we determined the minimum box size required to avoid ping-pong for any given combination of dimensionality and redundancy (***Supplementary Algorithm 3***, ***Figure 8–Figure Supplement 2A***). However, given the potentially large number of neurons participating in the initial 'ping', delayed networks require significantly larger boxes to avoid ping-pong. While they prevent ping-pong, wider boxes automatically reduce coding accuracy, even when the readout is rescaled (***Figure 8G***; ***Figure 8–Figure Supplement 1***).

An alternative to simply widening the box is to eliminate excitatory connections between direct and near antipodes. In this case, the bounds of a neuron's disconnected antipodes expand whenever it fires a spike, and only temporarily (***Figure 8F***). Just as wide boxes, these networks are less likely to initiate ping-pong. However, since their widening is local and only temporary, their performance is less affected and almost reaches baseline for higher-dimensional systems (***Figure 8H***), even for biologically plausible delays (1-2 msec). The rapid increase in firing due to the concomitant ping-pong effect can thus be avoided as well (see also ***Figure 8–Figure Supplement 1***).

Predictions on optogenetic perturbations

Finally, we investigate the effects that optogenetic perturbations would have on *SCN*s. We simulate optogenetic perturbations of *SCN*s with an extra current term on the perturbed neurons (see Material and Methods). The effect of these currents can again be understood as a change in the voltage threshold of each perturbed neuron: an inhibitory current injection leads to an increase of the voltage threshold, and an excitatory current injection to a threshold decrease. Geometrically, this is equivalent to a targeted movement of the perturbed bounds: inhibitory currents shift the respective bounds away from the centre of the box (***Figure 9A***), whereas excitatory currents have the opposite effect (***Figure 9B***).

It is plausible to assume that excitation and inhibition of a given neural system should have opposite effects, e.g. unilateral excitation of motor areas can lead to biases toward contralateral movements whereas inhibiting the same area would cause an ipsilateral bias [38]. In *SCN*s, though inhibition and excitation induce an opposite movement of the bound, the network response is not necessarily opposite. Indeed, in high redundancy *SCN*s, partial network inhibition is in general a silent perturbation and leads to no change in the readout, as unperturbed neurons can compensate for the perturbation by increasing their firing rates (***Figure 9A***). However, partial excitation almost always induces a bias on the readout, with excited neurons deforming the bounding box (***Figure 9B***). As the redundancy of the networks decreases, this effect becomes less pronounced. We note that in some conditions (e.g. for *SCN*s operating with a tight box) partial excitation does not induce a bias in the readout, but instead drives the system into the ping-pong regime (***Figure 9–Figure Supplement 1A,B***).

We furthermore predict an apparent paradoxical effect observed in electrophysiological recordings [39], where directly inhibited neurons may in fact become more active during the perturbation. While such an effect can be attributed to some type of disinhibition, the bounding box adds a geometric perspective: some of the inhibited neurons may have their bounds contribute with a larger surface of the box during the perturbation (***Figure 9–Figure Supplement 1C***), and thus have higher firing rates (***Figure 9–Figure Supplement 1D***).

Discussion

In this study, we characterise the functioning of spike coding networks under normal conditions and under a diversity of perturbations, using a simple, geometric visualisation, the bounding box. The bounding box delimits the error that an *SCN* tolerates in approximating a set of input signals, and its geometry is found to be largely determined by the properties of downstream decoders. The bounding box allows us to visualise and thus understand the dynamics of a spike coding network, including the firing of every single spike. We showed how various perturbations of the network, including neuron loss, changes in threshold or resets potentials, changes in synaptic weights, or increases in synaptic delays, can be mapped onto shape deformations of this bounding box. As long as the box stays intact, the

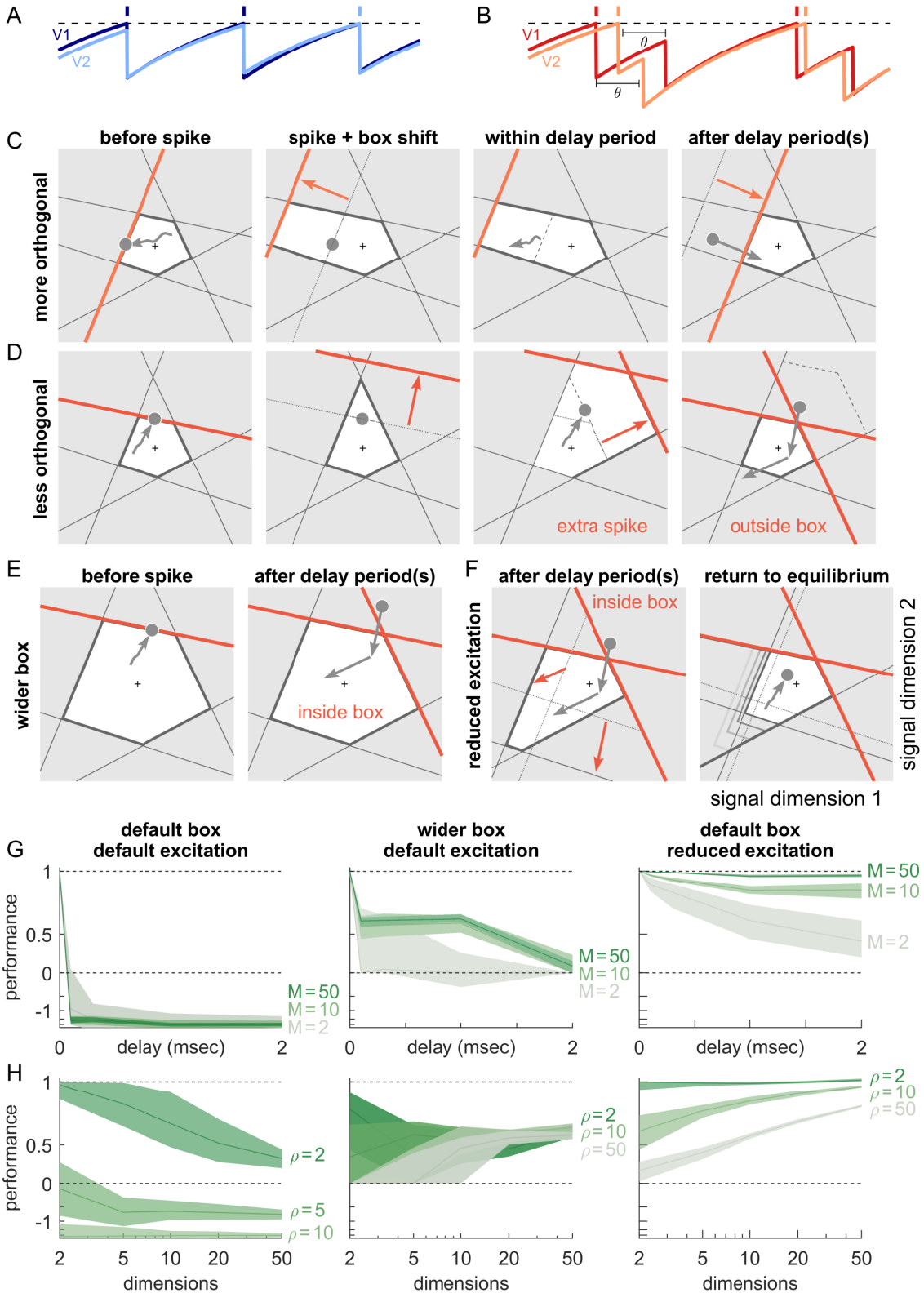

Figure 8: Synaptic transmission delays cause uninformed spikes, but high-dimensional low-excitation networks are less affected. (**A**) In an undelayed *SCN*, when membrane potentials V_1 and V_2 of two identically tuned neurons approach firing threshold (dashed), the first neuron to cross it will spike and instantly inhibit the second.

(continued on following page)

Figure 8, continued: (**B**) If recurrent connections are instead withheld for a delay θ, the second neuron may reach its own threshold before receiving this inhibition, emitting an 'uninformed' spike. (**C**) Readout dynamics in a delayed SCN that encodes a two-dimensional input. After the spike of the orange neuron, but before its postsynaptic arrival, the bounding box temporarily expands due to the retraction of the bound of the spiking neuron. (**D**) For less orthogonal pairs of neurons, the retraction of the boundary of a spiking neuron may expose the boundary of a similarly tuned neuron, leading to a suboptimally fired spike, and increasing the likelihood of 'ping-pong'. (**E**) Wider boxes or (**F**) removing excitation between nearly opposite decoders are two effective strategies to avoid 'ping-pong'. (**C-F**) Readout shown as grey circles and arrows, bounds of spiking neurons as coloured lines, and the resulting shift of other bounds as coloured arrows. (**G**) Redundancy $\rho = 10$. In default *SCN*s, performance (relative to the undelayed default network, see Material and Methods) drops sharply as delays increase (left). Preventing 'ping-pong' by either widening the box (centre) or removing the largest excitatory connections (right) restores robustness to biologically plausible delays, but performance remains lower at high redundancy. (**H**) Synaptic delay $\theta = 1$ msec. The detrimental effects of delays are eliminated in higher-dimensional *SCN*s when the box is widened (centre) or when the largest excitatory connections are removed (right). (G,H) Note the exponential scaling of the y axis. Left panel of (H) shows lower redundancies than elsewhere; more redundant *SCN*s have lower performance.

Figure 8–Figure supplement 1. Single trials with normal or wide boxes, and full or reduced connectivity (20 dimensions).

Figure 8–Figure supplement 2. Box size and reduced excitation to avoid ping-pong in delayed *SCN*s.

network's performance is essentially unaffected, in that downstream readouts of the network's outputs will not notice the perturbation. Our study therefore sheds light into the remarkable robustness of *SCN*s and provides potential links to the observed robustness of biological neural networks.

Robustness of spike coding networks

Robustness, i.e., the ability to maintain functionality despite perturbations, is a key property of biological systems, ranging from molecular signalling pathways to whole ecosystems. Several overarching principles have been identified that allow systems to be robust [40–43]. These include (1) negative feedback, to correct perturbations and recover functionality; (2) heterogeneity of components, to avoid common modes of failure; and (3) modularity or 'bow-tie' architectures, to create alternative pathways or solutions in the case of a perturbation. (4) Furthermore, making a system robust against certain perturbations almost always involves a tradeoff, in that the system becomes fragile against other perturbations.

These core themes can also be found in *SCN*s. (1) Negative feedback exists through extensive lateral connectivity (or, alternatively, through actual feedback of the readout, as in ***Figure 2F***), and is precisely tuned such that it automatically corrects any perturbations. (2) Individual neurons are heterogeneous and thereby allow the system (as visualised by the bounding box) to be more robust against the loss of components than if neurons were simply duplicated. (3) Since neuron space is always larger than signal space, there are many alternative neural codes ('alternative pathways') that give rise to the same linear readout, thus embodying a bow-tie architecture whose core is the signalling space. (4) Furthermore, the networks are fragile against any perturbation that leads to a shrinking of the box. Paradoxically, this fragility may become more relevant if a system becomes more redundant. These four themes may relate the robustness of the networks studied here to the more general topic of tissue robustness [41].

Given these properties, *SCN*s act like robust modules, in that they self-correct perturbations instead of passing them on, so that downstream networks remain unaffected. These observations remain correct even if we move beyond the simple autoencoder networks that we have studied here. Indeed, if we embed the networks with a set of slower connections to perform linear or non-linear computations [19, 44, 45], the robustness remains the same, as illustrated in ***Figure 1***, which relies on slow connections to generate the oscillations. These extensions work because the mechanisms underlying the encoding of the signals into spike trains are decoupled from the mechanisms that generate the dynamics of the signals (or readouts).

Fragility of spike coding networks

Despite their strong robustness, *SCN*s are also surprisingly fragile against any perturbations that cause an effective shrinking of the box, and thereby lead to a ping-pong effect. These problems can be ameliorated by widening the box,

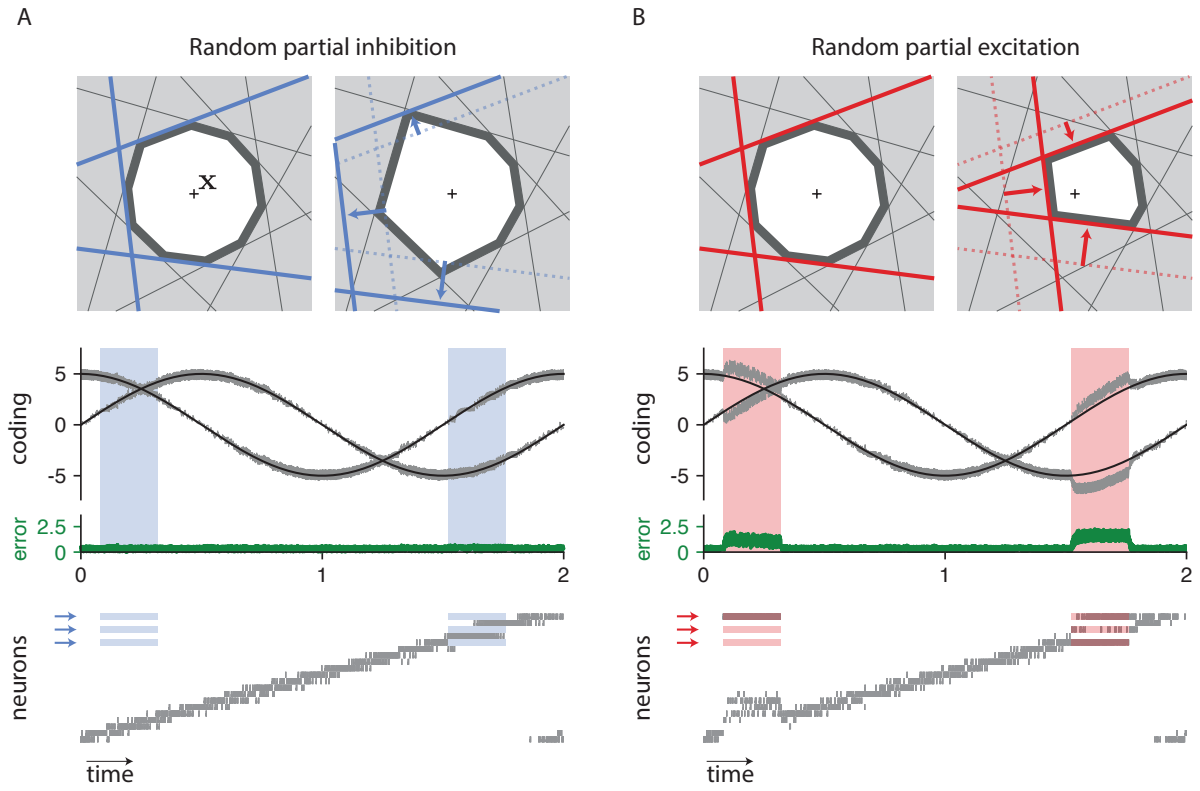

Figure 9: Predictions of *SCN*s response to optogenetic perturbations. (**A**) (Upper) Box deformation caused by inhibitory perturbation. Inhibited neurons move their bounds away from the centre of the box. (centre) Signal (black), linear readout (gray), and decoding error (green). Periods of inhibitory perturbations are highlighted in blue. A partial inhibitory perturbation does not induce any coding error. (Lower) Spike raster of the network. Arrows indicate the perturbed neurons. (**B**) (Upper) Box deformation caused by excitatory perturbation. Activated neurons move their bounds closer to the centre of the box. (centre) Signal (black), linear readout (gray), and decoding error (green). Periods of excitatory perturbations are highlighted in red. During both perturbation periods, the excitatory perturbations cause a readout error. (Lower) Spike raster of the network. Arrows indicate the perturbed neurons. Note that for both simulations in (A) and (B), we perturb the same neurons, at the same times and injecting a similar (but opposite) current. (Threshold $T = 1.55$)

Figure 9–Figure supplement 1. Simulations of random partial inhibition and excitation with tighter box (thresholds of 0.55), and paradoxical effect of optogenetic inhibition.

but this 'ad hoc' fix does not truly eliminate the underlying problem, which can re-appear, e.g., with higher redundancy (***Figure 6***). We believe that this shortcoming may point to a crucial mismatch between *SCN*s and real neural circuits. Interestingly, the ping-pong effect can be eliminated by cutting some excitatory connections which effectively 'opens' the bounding box temporarily in certain directions (***Figure 8F***). The elimination of excitatory connections breaks the symmetric treatment of excitatory and inhibitory connections that is otherwise a given in *SCN*s. Indeed, this symmetric treatment leads to neurons that both excite and inhibit their neighbors, thereby violating Dale's law. Future work will need to reconsider these issues which seem to be tightly connected. (We note that Boerlin et al. [19] developed *SCN*s that obey Dale's law, but did so without fixing the issue of the ping-pong effect.)

Structural robustness of neural networks

Historically, the study of robustness in neural networks has received relatively little attention, perhaps because classical models of neural networks can show a diversity of dynamics and functions, making it hard to define general principles of robustness. A key focus has been the robustness of attractors of the network dynamics, defined as the ability of a system to remain in the same attractor landscape despite perturbations. For instance, several neural systems seem to work like continuous attractors, such as the oculomotor integrator and the head direction system, which show patterns of activity at a continuum of levels and with long timescales [46, 47]. Such continuous attractors are structurally unstable, in that even small perturbations of the parameters or small amounts of noise lead to rapid dynamic drifts [46, 47]. Paradoxically, this fragility to perturbations is not observed in biological neural networks.

In order to achieve the required robustness, several biophysical mechanisms have been proposed to enhance continuous attractors models, e.g. bistability at the somatic level [48] or dendritic level [49]. More recent work proposed network-level mechanisms based on derivative feedback, in order to solve the problem of robustness for continuous attractor networks [50]. In our work, the problem disappears in some sense, because perturbations such as neuron loss, noise, or tuning of synapses are compensated on the level of spiking, as mediated by the fast, lateral connections. In turn, attractor dynamics can be implemented with a second, slower set of synaptic connections [19, 45], which effectively act on the level of the underlying estimated signals or readouts. Consequently, only perturbations that disturb the linear readout can impact the attractor dynamics. Interestingly, SCNs that implement continuous attractors are mathematically very similar to those that use derivative feedback [19, 50].

Models of neural networks implementing point attractors, such as the Hopfield model [51], are typically considered structurally robust, meaning that perturbations up to certain magnitudes of their parameters and the introduction of dynamics noise do not disrupt the attractor. We note, however, that perturbations in these networks lead to changes in neurons' firing rates, which may still cause changes in putative downstream linear readouts. From the point-of-view of linear readouts, perturbations are therefore not really compensated within attractor networks. This picture changes only when the readout is taken to be a classifier; only the combined system of attractor network and classifier readout can then be seen as a 'robust module', i.e., a module that keeps problems to itself, rather than spreading them to all who listen.

Similar observations apply to studies of the robustness of deep networks against various perturbations such as the loss of neurons [52, 53]. In these cases, the network's robustness is evaluated with respect to the output of a final classification step, such as the identification of an object. Indeed, a lot of work has been dedicated to make this final output robust to small perturbations, especially perturbations applied to the inputs [25, 54–56]. Based on the arguments above, we similarly expect that the problem of making a graded output robust will be harder and fundamentally different.

Robustness in the brain

The advent of optogenetic methods has led to a recent surge of perturbation studies, and a renewed interest in the robustness of brain circuits and possible compensatory mechanisms [57]. A few recent studies have found examples of instantaneous compensation against perturbations. For instance, the dynamics of premotor cortex activity in mice has been shown to be robust to unilateral (but not bilateral) silencing, suggesting a mechanism of redundancy across hemispheres that ensures such robustness [58]. The hippocampus has been shown to be robust against the elimination of place cells, through an immediate compensation in the remaining circuitry [59]. In monkey area MT, optogenetic inhibition had only a small and transient effect on the psychophysical performance in a motion

discrimination task [39]. Whether these observations can or cannot be explained with the mechanisms that we propose remains to be explored.

Of course, mechanisms of robustness exist at many levels, and we have only addressed the network level in our work. A canonical example of robustness through cellular mechanisms can be found in the crustacean stomatogastric system which is robust to temperature fluctuations [60, 61]. Here, the activity-dependent regulation of channel expression at the single cell level has been proposed as a mechanism to ensure the conservation of the firing patterns of the respective neurons and the proper network functioning under temperature perturbations [60].

Insights on spiking networks

Apart from these insights on robustness, our work also provides a framework to understand a large class of spiking networks. Spiking networks have traditionally been quite hard to understand, except for special cases [3, 62, 63]. Classical work on spiking networks has largely focused on understanding the dynamics of spiking networks, either in synchronous [64–67] or asynchronous regimes [12–15, 68], while paying less attention to functional implications. When a functional neural network is required, the default fallback have been neural networks based on firing rate units, as in the recent deep learning boom. In turn, functionality in spiking networks is usually imposed by transferring insights from rate networks. This strategy has led to spiking networks with neurons that fire regularly [69] or to spiking networks in which irregular firing is considered non-coding noise [e.g. 70].

Here, we have studied networks based on efficient spike coding, and we have shown how their dynamics can be understood within a lower-dimensional signal space, which is tightly linked to linear readouts. Since (low-dimensional) linear readouts are a ubiquitous finding in recordings from neural populations, we may speculate that our signal space is roughly equivalent to the latent subspaces discovered by linear projections of neural activities, as, e.g., obtained through dimensionality reduction methods [71]. This link between a space of neural activities and a space of (latent) signals is common to all network models based on low-rank connectivities [19, 46, 69, 72]. We believe that the link we made here—which visualises the spiking activity inside the signal space in a direct way—may provide useful insights into the functioning of spiking networks in the brain, and may well be expanded beyond the confines of the current study.

Methods and Materials

Spike coding networks and bounding box

Mathematically, *SCN*s can be derived from a single objective function that quantifies coding accuracy. Step-by-step derivation for the autoencoder networks can be found in Barrett et al. [23]; networks that additionally involve a set of slow connections are derived in Boerlin et al. [19]. Here, we focus on the autoencoder networks which contain all the crucial elements needed to understand the spiking dynamics of the networks. Instead of starting with an objective function, we take a slightly different perspective in our derivation here. This perspective ties more directly into our geometric interpretations, and also allows us to include the more general class of spike coding networks found after learning the recurrent connections [25].

In short, we assume that a network of N neurons encodes an M-dimensional input signal $\mathbf{x}(t)$, in its spike trains $\mathbf{s}(t)$, such that the signal can be read out from the filtered spike trains,

$$\hat{\mathbf{x}}(t) = \mathbf{D}\mathbf{r}(t) \tag{4}$$

$$\dot{\mathbf{r}}(t) = -\lambda\mathbf{r}(t) + \mathbf{s}(t). \tag{5}$$

Here, $\mathbf{x}(t)$ is the linear readout or signal estimate, the $M \times N$ matrix $\mathbf{D}$ contains the decoding weights (and each column corresponds to a decoding vector $\mathbf{D}_i$), the filtered spike trains are represented by $\mathbf{r}(t)$, and λ determines the filtering time constant.

The key idea of *SCN*s is to derive a spiking rule that bounds the difference between the input signal $\mathbf{x}$, and the linear readout $\hat{\mathbf{x}}$,

$$\|\mathbf{x} - \hat{\mathbf{x}}\| < T, \tag{6}$$

where $\|\cdot\|$ denotes the Euclidean distance or L2 norm and T determines the maximally allowed difference. In *SCN*s, we approximate this bound (which defines a hypersphere) by a set of linear bounds or inequalities, one for each

neuron i,

$$\mathbf{D}_i^\mathsf{T}(\mathbf{x} - \hat{\mathbf{x}}) < T. \tag{7}$$

For simplicity, we assume that the decoding vectors $\mathbf{D}_i$ have unit norm. Each inequality defines a half-space of solutions for the readout $\hat{\mathbf{x}}$. For properly chosen $\mathbf{D}_i$, the intersection of all of these half-spaces is non-empty and bounded, and thus forms the interior of the bounding box. Geometrically, the equations define a polytope $B = \{\hat{\mathbf{x}} \in \mathbb{R}^M | \mathbf{D}^\mathsf{T}(\mathbf{x} - \hat{\mathbf{x}}) < \mathbf{T}\}$. If the thresholds are chosen sufficiently large, then crossing a bound and firing a spike keeps the readout inside the bounding box.

The dynamics of the autoencoder *SCN*s are obtained by identifying the left-hand side of the above equation with the neuron's voltage, V_i, and then taking the temporal derivative [19, 23]. If we also add some noise to the resulting equations, we obtain,

$$\dot{\mathbf{V}} = -\lambda\mathbf{V} + \mathbf{D}^\mathsf{T}(\lambda\mathbf{x}(t) + \dot{\mathbf{x}}(t)) - \mathbf{D}^\mathsf{T}\mathbf{D}\mathbf{s}(t) + \sigma_V\eta(t), \tag{8}$$

which describes a network of leaky integrate-and-fire neurons. The first term on the right-hand side is the leak, the second term corresponds to the feedforward input signals to the network, the third term captures the fast recurrent connectivity, with synaptic weights $\Omega_{ij} = -\mathbf{D}_i^\mathsf{T}\mathbf{D}_j$, and the fourth term is added white current noise with standard deviation σ_V. When the voltage V_i reaches the threshold T, the self-connection $\Omega_{ii} = -\mathbf{D}_i^\mathsf{T}\mathbf{D}_i$ causes a reset of the voltage to $V_{\text{reset}} = T + \Omega_{ii}$. For biological plausibility, we also consider a small refractory period of $\tau_{\text{ref}} = 2$ms for each neuron. We implemented this refractory period by simply omitting any spikes coming from the saturated neuron during this period.

Generalisation of the bounding box

There are two straightforward generalisations of the bounding box, as described so far. One generalisation is to allow neurons to have different thresholds, in which case, the bounding box can take more elliptical shapes. The second generalisation consists in decoupling the orientation of a neuron's face from the direction of the readout jump, which can be achieved by choosing a voltage $V_i = \mathbf{F}_i(\mathbf{x} - \hat{\mathbf{x}})$, where $\mathbf{F}_i$ denotes the norm vector of a bounding box face. In contrast, the readout jumps in the direction $\mathbf{D}_i$, and a non-orthogonal jump with respect to the face requires $\mathbf{D}_i \neq \mathbf{F}_i$. Indeed, for elliptically shaped bounding boxes, non-orthogonal jumps of the readout can sometimes be advantageous. The more general dynamical equation for *SCN*s is then given by

$$\dot{\mathbf{V}} = -\lambda\mathbf{V} + \mathbf{F}(\lambda\mathbf{x}(t) + \dot{\mathbf{x}}(t)) - \mathbf{F}\mathbf{D}\mathbf{s}(t) + \sigma_V\boldsymbol{\eta}(t), \tag{9}$$

and was first described in Brendel et al. [25]. As shown here, this generalisation has a bounding box interpretation as well. For simplicity, however, we have chosen to present the bounding box with symmetric connectivities in the main text.

Readout biases and corrections

When one of the neurons fires, its spike changes the readout, which jumps into the bounding box. In previous work, these jumps were generally taken to reach the opposing face of the bounding box, because the neurons' thresholds were linked with the length of the jumps through the equation $T_i = \|\mathbf{D}_i\|^2/2$ [19, 23]. This setting creates a tight error bounding box around $\mathbf{x}$, and guarantees that the average readout matches the input signal.

When the jumps are significantly shorter than the average bounding box width, the average readout will be biased away from the input signal. However, this bias can be corrected by rescaling the readout.

$$\hat{\mathbf{x}} = \left(\frac{\langle\|\mathbf{D}\mathbf{r}\|\rangle + T - \frac{1}{2}}{\langle\|\mathbf{D}\mathbf{r}\|\rangle}\right)\mathbf{D}\mathbf{r}, \tag{10}$$

where the angular brackets denote the time-averaged estimate strength. Note that this new scaling factor was analytically derived for *SCN*s shaped like hyperspheres (i.e. in the limit of an infinite number of neurons N) and assuming a constant stimulus. In cases where both of these assumptions are violated, we empirically found that we can apply a correction to the readout using a similar scaling as in ***Equation 10*** where $\langle\|\mathbf{D}\mathbf{r}\|\rangle \approx \mathbf{D}\mathbf{r}(t)$. We use this correction in all our simulations that involve increased thresholds ($T > 0.5$).

In ***Figure 1***, we used networks that involve an extra set of slow recurrent connections [19]. In this case, we are additionally required to scale the slow recurrent connectivity matrix Ω_{slow} with the same scaling factor as the corrected readout in ***Equation 10***:

$$\Omega_{\text{slow}} = \left(\frac{\langle \|\mathbf{Dr}\| \rangle + T - \frac{1}{2}}{\langle \|\mathbf{Dr}\| \rangle} \right) \mathbf{D}^{\mathsf{T}} \left(\mathbf{A} + \lambda \mathbf{I} \right) \mathbf{D}. \tag{11}$$

Geometry of high-dimensional bounding boxes

The dimensionality of the bounding box is determined by the dimensionality M of the input signal. Throughout the illustrations in the Results section, we mostly used two-dimensional bounding boxes for graphical convenience. In order to illustrate some properties of higher-dimensional error bounding boxes (***Figure 3***), we compared their behaviour against that of hyperspheres and hypercubes. We defined the equivalent hypersphere as

$$\{\mathbf{p} \in \mathbb{R}^M : \|\mathbf{p}\|_2 \leq T\},$$

and the equivalent hypercube as

$$\{\mathbf{p} \in \mathbb{R}^M : \|\mathbf{p}\|_\infty \leq T\},$$

where $\|\mathbf{p}\|_2 = \sqrt{p_1^2 + \ldots + p_n^2}$ and $\|\mathbf{p}\|_\infty = \max_i |p_i|$. In practice, we chose the smallest box size, $T = 0.5$ (***Figure 3***).

For a first comparison, we took the intersection between the border of the M-dimensional polytope B and a random two-dimensional plane containing the centre of the polytope. We computed such intersections numerically by first choosing two random and orthogonal directions u and v in the full space defining the two-dimensional plane. Then for each $\theta \in [0, 2\pi]$, we defined a ray in the two-dimensional plane, $w(\rho) = \rho \cos(\theta) u + \rho \sin(\theta) v$, and then plotted

$$\rho(\theta) = \underset{\rho > 0, w(\rho) \in B}{\arg\max} \; w(\rho).$$

For a second comparison, we found the distribution of angles between neighbouring neurons by first randomly choosing one neuron, and then moving along the surface of the M-polytope in a random direction, until we found a point that belongs to the face of a different neuron. We then computed the angle between the decoding weights of those two neurons.

Finally, we illustrated a high-dimensional bounding box with a set of Gabor patches. These were defined as

$$g(x, y; \lambda, \theta, \sigma, \gamma) = \exp\left(-\frac{\tilde{x}^2 + \gamma^2 \tilde{y}^2}{2\sigma^2} \right) \cos\left(2\pi \frac{\tilde{x}}{\lambda} + \frac{\pi}{2} \right), \tag{12}$$

where $\tilde{x} = x \cos\theta + y \sin\theta$ and $\tilde{y} = -x \sin\theta + y \cos\theta$. For our purposes, we randomly chose the Gabor parameters: λ, the wavelength of the sinusoidal stripe pattern, was sampled uniformly from $\{3, 5, 10\}$ Hz; θ, the orientation of the stripes, was sampled uniformly in $[0, 2\pi]$; σ, the standard deviation of the Gaussian envelope, was sampled uniformly from $\{1, 1.5\}$; γ, the spatial aspect ratio, was sampled uniformly from $\{1, 1.5\}$.

Finally we randomly centred the resulting Gabor patch in one of 9 different locations on the 13×13 grid. We computed the angle (in the 169-dimensional space) between the Gabor patches and found that roughly 80% of the neurons are quasi-orthogonal (their angle falls between 85 and 95 degrees) to a given example patch.

Parameter choices

Spike coding networks depend on several parameters:

1. The number of neurons in the network, N.
2. The number of signals fed into the network, M, also called the dimensionality of the signal.
3. The $M \times N$ matrix of decoding weights, D_{ik}, where each column $\mathbf{D}_k$, corresponds to the decoding weights of one neuron.
4. The inverse time constant of the exponential decay of the readout, λ.
5. The threshold (or error tolerances) of the neurons, T.
6. The refractory period, τ_{ref}.
7. The current noise, σ_V.

Table 1. *SCN* parameter values.

	Variable (Unit)	baseline value	value range
N	network size		[2, 5000]
M	signal dimensions		[1, 100]
ρ	network redundancy $\frac{N}{M}$		[2, 50]
$\|\|\mathbf{D}_i\|\|_2$	decoder norms	1	
$\frac{1}{\lambda}$	decoder time constant (ms)	10	
T_i	threshold (a.u.)	0.55	[0.5, 1.55*]
$t_{\max}$	trial duration (s)	5	
Δt	simulation time step (ms)	0.1	[0.01 0.1]
σ_x	standard deviation of each signal component	3	
η_x	signal noise	0.5	
τ_{ref}	refractory period (ms)	2	[0, 10]
$V_{i,\text{reset}}$	reset (a.u.)	1.014	[1, 1.5]
σ_V	current noise (a.u.)	0.5	[0, 3]
δ_Ω	synaptic scaling/noise	0	[0, 0.2]
p_s	sparsity factor	0	[0, 0.4]
p_f	synaptic failure	0	[0, 0.1]
θ	recurrent delay (ms)	0	[0, 2]
p_{opto}	optogenetic inhibition	0	[-0.05, 0]
p_{opto}	optogenetic excitation	0	[0, 0.05]

*To counteract synaptic delays as in ***Figure 8***, thresholds $T > 1.55$ were also used (***Figure 8–Figure Supplement 2***).

These parameters fully define both the dynamics and architecture – in terms of feedforward and recurrent connectivity – of SCNs, as well as the geometry of the bounding box. We did a parameter sweep to narrow down the range of parameters that matches key observational constraints, such as low median firing rates, as found in cortex [27, 73] (***Figure 4C***), and coefficients of variation of interspike intervals close to one for each neuron, corresponding to Poisson-like spike statistics (***Figure 4E***). ***Table 1*** displays the range of parameters used to simulate baseline and perturbed networks.

Input signal

We used two different types of inputs throughout our simulations. The results shown in ***Figure 1C***, ***Figure 6C*** and ***Figure 9*** are for a circular, 2-dimensional signal,

$$\mathbf{x}(t) = (a\sin(\omega t), a\cos(\omega t))^{\mathsf{T}}, \tag{13}$$

with constant amplitude a and constant frequency ω.

For all other figure panels, the input signal is smooth but random: for each trial, we sample a single point in input space from an M-dimensional Gaussian distribution,

$$\mathbf{x}_0 \sim \mathcal{N}\left(\mathbf{0}, \sigma_x^2\mathbf{I}\right). \tag{14}$$

The input signal ramps linearly from zero to this point $\mathbf{x}_0$ during the first 400ms. For the rest of the trial, the input to the neurons is set to slowly vary around this chosen value for each dimension of $\mathbf{x}$: to generate the slow variability, we sample from an M-dimensional Gaussian distribution as many times as time steps in the rest of the trial; we then twice-filter the samples with a moving average window of 1s for each dimension of $\mathbf{x}$, and for each dimension of $\mathbf{x}$ and across time, we normalise the individual slow variabilities to not exceed $\eta_x = 0.5$ in magnitude. This procedure was supposed to mimic experimental trial-to-trial noise.

Metrics and network benchmarking

To compare the behaviour of *SCN*s under baseline conditions to those under the different perturbations, we need reliable measures of both coding accuracy and firing statistics. Below, we describe the measures used in this study.

Random seeds

For each simulated trial, we generate a new *SCN* with a different random distribution of decoding weights, random input signal, and random voltage noise. We initialise the random number generator with a different seed before each trial. If a single trial with a perturbation is compared to a single unperturbed trial (see **Network Performance**), each such pair shares a seed and thus has identical decoders, connectivity, input and noise unless explicitly affected by the perturbation.

Distributions of firing rates and coefficients of variation

We measured the time-averaged firing rate for a given neuron by dividing the total number of spikes by the total duration of a trial. The coefficient of variation (CV) of a single spike train is computed as the ratio of the standard deviation of the interspike intervals (ISI) to their mean

$$\mathrm{CV} = \frac{\sigma_{\mathrm{ISI}}}{\mu_{\mathrm{ISI}}}. \tag{15}$$

We recorded the full distributions of both the firing rates and CVs for a given network, pooling across neurons and different trials.

Network Performance

We defined coding error as the mismatch between the encoded signal $\mathbf{x}(t)$ and spike-based reconstruction $\hat{\mathbf{x}}(t)$. Note that this quantity is computed at every time step t, and separately for each dimension m of signal space.

$$\epsilon\,(t, m) = |x_m(t) - \hat{x}_m(t)|\,, \qquad m \in [1, M]. \tag{16}$$

One straightforward approach when calculating the error is computing its L2 (or euclidean) norm at every time step. However, since this quantity scales with the signal dimensionality M, it is not suited for analysing how the distribution of the errors incurred by *SCN*s varies across different signal dimensionalities. Therefore, in ***Figure 4***, we pool over all dimensions and all time steps of all trials and obtain a characteristic distribution of errors for each network.

When our aim was to simply compare the relative network performance with and without the different perturbations, we opted to use the most straightforward error metric, i.e. the average (in time) of the L2 norm of the coding error

$$E_{\mathrm{trial}} = \langle \|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\| \rangle_t. \tag{17}$$

This value was then compared to the error of a dead network (i.e. where $\hat{\mathbf{x}}(t) = 0$) and a reference one using the formula

$$P = \frac{E_{\mathrm{trial}} - E_{\mathrm{dead}}}{E_{\mathrm{reference}} - E_{\mathrm{dead}}}, \tag{18}$$

where the reference network is the one without any perturbation, and $E_{\mathrm{dead}} = \langle \|\mathbf{x}(t)\| \rangle_t$.

Benchmarking

To fully compare the behaviour of *SCN*s under baseline conditions to those under the different perturbations, we adopt the following benchmarking procedure: each trial with a perturbation is compared to an otherwise identical trial without perturbation. For each such pair of trials, N random decoding weights are drawn from an M-dimensional standard normal distribution,

$$\mathbf{D}_j \sim \mathcal{N}(0, \mathbf{I}), \tag{19}$$

and then normalised,

$$\mathbf{D}_j \leftarrow \mathbf{D}_j / \|\mathbf{D}_j\|_2. \tag{20}$$

such that each neuronal decoding vector is of length 1. We then apply our M-dimensional input signal $\mathbf{x}$ as described above. Coding error and spike statistics are recorded for each trial.

This procedure is repeated multiple times ($N_{\mathrm{trials}} \geq 20$), each repetition with a different random seed, resulting in different network connectivity, inputs (with the exception of ***Figure 4A,C,E***, where a single network and input are used), and injected current noise. Then, for each perturbed trial, we use the same trial seed to control for the trial randomisation.

We choose this benchmarking procedure to sample input space in an unbiased way. Even though the rate of change of the input is constrained to be small on the time scale of a single trial, we sample a large part of the input space from trial to trial. This ensures that network performance is not accidentally dominated by a perfect match, or mismatch, between the fixed decoding weights and a given random input. Particularly bad mismatches may still lead to high decoding errors, but because our error measure considers the median response, these extremes do not bias our benchmarking procedure.

Number of simulations

Figure 1C shows a single trial. The distributions in ***Figure 4*** are across 820 trials of 100s (500 trials for $M = 5$ and $\rho = 5$, 200 for $M = 5$ and $\rho = 50$, 100 for $M = 100$ and $\rho = 5$ and 20 for $M = 100$ and $\rho = 50$). ***Figure 4B,D,F*** each show a total of 29,400 trials. ***Figure 6D-F*** show 16,830 pairs of trials, ***Figure 6–Figure Supplement 1*** shows 4,996 pairs, and ***Figure 6–Figure Supplement 2*** shows 1 perturbed trial per row. Each data panel of both ***Figure 7*** and ***Figure 7–Figure Supplement 1*** consists of 840 trials. ***Figure 8G-H*** show 18,000 pairs of trials, or 200 pairs per data point, and ***Figure 8–Figure Supplement 1*** shows 1 perturbed trial per row. ***Figure 9*** and ***Figure 9–Figure Supplement 1*** show 1 perturbed trial per row.

Perturbations

Here, we formalise all the perturbations addressed in this study.

Voltage noise

We implement voltage noise as an extra random current on the voltage dynamics. This term could be added to the voltage itself, but since an *SCN* is a type of leaky integrate-and-fire network, spike generation depends only on the difference between voltage and threshold,

$$V_j(t) \geq T_j. \tag{21}$$

As we focus on spike times instead of subthreshold activity, we can thus move the voltage noise term from one side of ***Equation 21*** to the other, and include it in the definition of the threshold instead. In either case, the extra current follows a Wiener process scaled by σ_V which denotes the standard deviation of the noise process with Gaussian increments (see ***Equation 8***). In the absence of recurrence,

$$dV_j(t) = -\lambda V_j(t)\, dt + \nu(t)\sqrt{dt}, \qquad \nu \sim \mathcal{N}_M(0, \sigma_V). \tag{22}$$

*SCN*s leaky integration with time constant λ biases the random walk of the thresholds back towards their default values, so for stationary input, the thresholds follow an Ornstein-Uhlenbeck process. Note that if we had instead applied noise to the voltages themselves, these would perform a random walk biased towards their equilibrium potential.

Synaptic perturbations

In this study, we investigated four different ways to induce perturbations at the synaptic level.

1. Synaptic scaling (***Figure 7***): we perturb synapses between different neurons ($i \neq j$) by a multiplicative noise term

$$\Omega_{i,j} \leftarrow \Omega_{i,j} * (1 - \delta_\Omega)^{u_{i,j}}, \tag{23}$$

   where $u_{i,j} \sim \mathcal{U}(-1, 1)$. Here, the parameter δ_Ω is the maximum weight change in percentage of each synapse.
2. Synaptic noise (***Figure 7–Figure Supplement 1***): we add a time-varying multiplicative noise term to all synapses between different neurons

$$\Omega_{i,j}(t) \leftarrow \Omega_{i,j} * (1 - \delta_\Omega)^{u_{i,j}}, \tag{24}$$

   where $u_{i,j} \sim \mathcal{U}(-1, 1)$, is drawn at every time step. Note that we opted for a multiplicative noise term to avoid a single synapse to undergo the biologically unrealistic change from inhibitory to excitatory.
3. Sparsity (***Figure 7–Figure Supplement 1***): we remove synapses between neurons by setting their connection weights to 0. We specifically target the fraction p_s of synapses that are weakest in absolute value.
4. Synaptic failure (***Figure 7–Figure Supplement 1***): all synapses have their original value but fail with probability p_f, independently of each other.

Synaptic delays

We implement delayed recurrent connections with the same constant delay length $\theta \geq 0$ for all pairs of neurons. Regardless of whether or not lateral excitation and inhibition are delayed in this way, the self-reset of a neuron onto itself remains instantaneous. ***Equation 3*** thus becomes

$$V_i = \mathbf{D}_i^\top \mathbf{x} - \sum_{k=1}^{N} \mathbf{D}_i^\top \mathbf{D}_k \left(r_k(t) \cdot (1 - \delta_{ik}) + r_k(t - \theta) \cdot \delta_{ik} \right), \tag{25}$$

where δ_{ik} is Kronecker's delta. We assume that the decoder readout is equally delayed.

Optogenetic perturbations

We simulate optogenetic perturbations in *SCN*s by incorporating an external additive current $\mathbf{p}(t)$ in their voltage dynamics

$$\dot{\mathbf{V}} = -\lambda \mathbf{V} + \mathbf{D}^\mathsf{T}(\dot{\mathbf{x}} + \lambda \mathbf{x}) + \mathbf{p}, \tag{26}$$

where $\mathbf{p}(t)$ is a vector function of size N capturing the temporal evolution of the perturbation for each neuron. In our simulations, we used simple square functions and set $p_i(t) = p_{\text{opto}}$ for the duration of the perturbation, and $p_i(t) = 0$ otherwise. For the unperturbed neurons, we set $p_i(t) = 0$ for the entirety of the simulation.

Note that adding a current $\mathbf{p}(t)$ to the voltage dynamics is equivalent to a transient change in the neuronal thresholds, similar to our previous transfer of voltage noise to the thresholds:

$$\begin{aligned} &\dot{\mathbf{V}} = -\lambda \mathbf{V} + \mathbf{D}^\mathsf{T}(\dot{\mathbf{x}} + \lambda \mathbf{x}) + \mathbf{p} \\ &\mathbf{V} \leq \mathbf{T} \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} &\dot{\mathbf{V}} = -\lambda \mathbf{V} + \mathbf{D}^\mathsf{T}(\dot{\mathbf{x}} + \lambda \mathbf{x}) \\ &\mathbf{V} \leq \mathbf{T} - h * \mathbf{p} \ \text{ with } \ h(t) = \Theta(t) e^{-\lambda t}. \end{aligned} \tag{27}$$

Table 1 includes the range of perturbations used throughout this manuscript.

Numerical implementation of *SCN*s

We numerically solve the differential equations (***Equation 8***) describing the temporal evolution of membrane voltage by the forward Euler-Maruyama method. We implemented this method in both MATLAB and Python, and both sets of code can be used interchangeably. We will make all our code for simulation, analysis and figure generation, as well as sample data files, available after publication.

MATLAB code was tested under version R2018b. It only requires the core software without any of the optional MATLAB toolboxes. PYTHON scripts are written with Jupyter Notebook and sped up by Numba.

Simultaneous crossing of multiple bounds

SCN neurons are integrate-and-fire neurons that spike whenever their voltage exceeds their threshold, $V_k \geq T_k$. In our geometric perspective, this happens whenever the readout is located on or outside one or several of the bounds representing these thresholds. Whether by perturbations or because of finite simulation time steps, more than one bound may be crossed during the same step, and more than one neuron may thus be eligible to spike. Therefore, we have devised an algorithm to simulate *SCN*s without time step dependence, while preserving the effect of perturbations (***Supplementary Algorithm 1***).

Note that when considering finite delays θ, delayed lateral recurrence arrives only at the end of each time step (***Supplementary Algorithm 2***).

Iterative adaptation of parameters to avoid ping-pong

In *SCN*s with delays, we can avoid ping-pong either by increasing box size or by removing a number of strongest excitatory connections. In both cases, we compute the minimum required value offline using an iterative procedure (***Supplementary Algorithm 3***). Note that trials must be sufficiently long to avoid false-negative reports of ping-pong.

Movie visualisation

All movies were produced in Python, with the exception of the three-dimensional visualisation of a polytope, for which we used the *bensolve* toolbox for MATLAB [74].

References

[1] P. Dayan, L. F. Abbott, and L. Abbott. Theoretical neuroscience: computational and mathematical modeling of neural systems. 2001.

[2] L. F. Abbott, B. DePasquale, and R.-M. Memmesheimer. Building functional networks of spiking model neurons. *Nature neuroscience*, 19(3):350, 2016.

[3] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[4] C. von der Malsburg. The correlation theory of brain function (internal report 81-2). *Goettingen: Department of Neurobiology, Max Planck Intitute for Biophysical Chemistry*, 1981.

[5] C. M. Gray, P. König, A. K. Engel, and W. Singer. Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, 338(6213):334–337, 1989.

[6] J. J. Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376(6535): 33–36, 1995.

[7] F. Rieke, D. Warland, R. D. R. Van Steveninck, W. S. Bialek, et al. *Spikes: exploring the neural code*. MIT press Cambridge, 1996.

[8] M. N. Shadlen and W. T. Newsome. The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *Journal of neuroscience*, 18(10):3870–3896, 1998.

[9] S. Thorpe, A. Delorme, and R. Van Rullen. Spike-based strategies for rapid processing. *Neural networks*, 14(6-7):715–725, 2001.

[10] R. Gütig and H. Sompolinsky. The tempotron: a neuron that learns spike timing–based decisions. *Nature neuroscience*, 9(3): 420–428, 2006.

[11] R. Gütig. Spiking neurons can discover predictive features by aggregate-label learning. *Science*, 351(6277):aab4113, 2016.

[12] C. Van Vreeswijk and H. Sompolinsky. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274 (5293):1724–1726, 1996.

[13] D. J. Amit and N. Brunel. Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 7(3):237–252, 1997.

[14] N. Brunel. Dynamics of networks of randomly connected excitatory and inhibitory spiking neurons. *Journal of Physiology-Paris*, 94(5-6):445–463, 2000.

[15] A. Renart, J. De La Rocha, P. Bartho, L. Hollender, N. Parga, A. Reyes, and K. D. Harris. The asynchronous state in cortical circuits. *science*, 327(5965):587–590, 2010.

[16] T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science*, 334(6062):1569–1573, 2011.

[17] R. Rosenbaum, M. A. Smith, A. Kohn, J. E. Rubin, and B. Doiron. The spatial structure of correlated neuronal variability. *Nature neuroscience*, 20(1):107, 2017.

[18] Y. Ahmadian and K. D. Miller. What is the dynamical regime of cerebral cortex? *arXiv:1908.10101*, 2019.

[19] M. Boerlin, C. K. Machens, and S. Denève. Predictive coding of dynamical variables in balanced spiking networks. *PLoS computational biology*, 9(11), 2013.

[20] R. Bourdoukan, D. Barrett, S. Deneve, and C. K. Machens. Learning optimal spike-based representations. In *Advances in neural information processing systems*, pages 2285–2293, 2012.

[21] D. G. Barrett, S. Denève, and C. K. Machens. Firing rate predictions in optimal balanced networks. In *Advances in Neural Information Processing Systems*, pages 1538–1546, 2013.

[22] S. Denève and C. Machens. Efficient codes and balanced networks. *Nature Neuroscience*, 19(3):775–82, March 2016.

[23] D. G. Barrett, S. Deneve, and C. K. Machens. Optimal compensation for neuron loss. *Elife*, 5:e12454, 2016.

[24] M. Boerlin and S. Denève. Spike-based population coding and working memory. *PLoS computational biology*, 7(2), 2011.

[25] W. Brendel, R. Bourdoukan, P. Vertechi, C. K. Machens, and S. Denéve. Learning to represent signals spike by spike. *PLoS computational biology*, 16(3):e1007692, 2020.

[26] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images., 1996. ISSN 0028-0836.

[27] A. Wohrer, M. D. Humphries, and C. K. Machens. Population-wide distributions of neural activity during perceptual decision-making. *Progress in neurobiology*, 103:156–193, 2013.

[28] A. Roxin, N. Brunel, D. Hansel, G. Mongillo, and C. van Vreeswijk. On the distribution of firing rates in networks of cortical neurons. *Journal of Neuroscience*, 31(45):16217–16226, 2011.

[29] E. Moreno, Y. Fernandez-Marrero, P. Meyer, and C. Rhiner. Brain regeneration in drosophila involves comparison of neuronal fitness. *Current Biology*, 25(7):955–963, 2015.

[30] J. H. Morrison and P. R. Hof. Life and death of neurons in the aging brain. *Science*, 278(5337):412–419, 1997.

[31] D. E. Bredesen, R. V. Rao, and P. Mehlen. Cell death in the nervous system. *Nature*, 443(7113):796, 2006.

[32] D. S. Coelho, S. Schwartz, M. M. Merino, B. Hauert, B. Topfel, C. Tieche, C. Rhiner, and E. Moreno. Culling less fit neurons protects against amyloid-β-induced brain damage and cognitive and motor decline. *Cell reports*, 25(13):3661–3673, 2018.

[33] A. A. Faisal, L. P. Selen, and D. M. Wolpert. Noise in the nervous system. *Nature reviews neuroscience*, 9(4):292, 2008.

[34] A. Destexhe, M. Rudolph, J.-M. Fellous, and T. J. Sejnowski. Fluctuating synaptic conductances recreate in vivo-like activity in neocortical neurons. *Neuroscience*, 107(1):13–24, 2001.

[35] J.-M. Fellous, M. Rudolph, A. Destexhe, and T. J. Sejnowski. Synaptic background noise controls the input/output characteristics of single cells in an in vitro model of in vivo activity. *Neuroscience*, 122(3):811–829, 2003.

[36] M. Chalk, B. Gutkin, and S. Denève. Neural oscillations as a signature of efficient coding in the presence of synaptic delays. *eLife*, (5):e13824, 2016. doi: $10.7554/\mathrm{eLife}.13824$.

[37] C. E. Rullán Buxó and J. W. Pillow. Poisson balanced spiking networks. *bioRxiv*, 2019. doi: $10.1101/836601$.

[38] P. J. Gonçalves, A. B. Arrenberg, B. Hablitzel, H. Baier, and C. K. Machens. Optogenetic perturbations reveal the dynamics of an oculomotor integrator. *Frontiers in neural circuits*, 8:10, 2014.

[39] C. R. Fetsch, N. N. Odean, D. Jeurissen, Y. El-Shamayleh, G. D. Horwitz, and M. N. Shadlen. Focal optogenetic suppression in macaque area mt biases direction discrimination and decision confidence, but only transiently. *eLife*, 7:e36523, 2018.

[40] M. E. Csete and J. C. Doyle. Reverse engineering of biological complexity. *science*, 295(5560):1664–1669, 2002.

[41] H. Kitano. Biological robustness. *Nature Reviews Genetics*, 5(11):826, 2004.

[42] J. M. Whitacre. Biological robustness: paradigms, mechanisms, and systems principles. *Frontiers in genetics*, 3:67, 2012.

[43] M.-A. Félix and M. Barkoulas. Pervasive robustness in biological systems. *Nature Reviews Genetics*, 16(8):483, 2015.

[44] C. Savin and S. Deneve. Spatio-temporal representations of uncertainty in spiking neural networks. In *Advances in Neural Information Processing Systems*, pages 2024–2032, 2014.

[45] D. Thalmeier, M. Uhlmann, H. J. Kappen, and R.-M. Memmesheimer. Learning universal computations with spikes. *PLoS computational biology*, 12(6), 2016.

[46] H. S. Seung. How the brain keeps the eyes still. *Proceedings of the National Academy of Sciences*, 93(23):13339–13344, 1996.

[47] K. Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *Journal of Neuroscience*, 16(6):2112–2126, 1996.

[48] A. A. Koulakov, S. Raghavachari, A. Kepecs, and J. E. Lisman. Model for a robust neural integrator. *Nature neuroscience*, 5(8):775, 2002.

[49] M. Goldman, J. Levine, G. Major, D. Tank, and H. Seung. Dendritic hysteresis adds robustness to persistent neural activity in a model neural integrator. *Cereb. Cortex*, 13:1185–1195, 2003.

[50] S. Lim and M. S. Goldman. Balanced cortical microcircuitry for maintaining information in working memory. *Nature neuroscience*, 16(9):1306, 2013.

[51] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[52] A. S. Morcos, D. G. Barrett, N. C. Rabinowitz, and M. Botvinick. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018.

[53] D. G. Barrett, A. S. Morcos, and J. H. Macke. Analyzing biological and artificial neural networks: challenges with opportunities for synergy? *Current opinion in neurobiology*, 55:55–64, 2019.

[54] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[55] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[56] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, and A. Madry. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[57] S. B. Wolff and B. P. Ölveczky. The promise and perils of causal circuit manipulations. *Current opinion in neurobiology*, 49:84–94, 2018.

[58] N. Li, K. Daie, K. Svoboda, and S. Druckmann. Robust neuronal dynamics in premotor cortex during motor planning. *Nature*, 532 (7600):459, 2016.

[59] S. Trouche, P. V. Perestenko, G. M. van de Ven, C. T. Bratley, C. G. McNamara, N. Campo-Urriza, S. L. Black, L. G. Reijmers, and D. Dupret. Recoding a cocaine-place memory engram to a neutral engram in the hippocampus. *Nature neuroscience*, 19(4): 564–567, 2016.

[60] T. O'Leary and E. Marder. Temperature-robust neural function from activity-dependent ion channel regulation. *Current Biology*, 26(21):2935–2941, 2016.

[61] S. A. Haddad and E. Marder. Circuit robustness to temperature perturbation is altered by neuromodulators. *Neuron*, 100(3): 609–623, 2018.

[62] W. Maass and C. M. Bishop. *Pulsed neural networks*. MIT press, 1999.

[63] T. P. Vogels, K. Rajan, and L. F. Abbott. Neural network dynamics. *Annu. Rev. Neurosci.*, 28:357–376, 2005.

[64] J. J. Hopfield and A. V. Herz. Rapid local synchronization of action potentials: Toward computation with coupled integrate-and-fire neurons. *Proceedings of the National Academy of Sciences*, 92(15):6655–6662, 1995.

[65] M. Abeles. *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press, 1991.

[66] M. Herrmann, J. Hertz, and A. Prügel-Bennett. Analysis of synfire chains. *Network: computation in neural systems*, 6(3):403–414, 1995.

[67] M. Diesmann, M.-O. Gewaltig, and A. Aertsen. Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402(6761):529–533, 1999.

[68] C. Van Vreeswijk and H. Sompolinsky. Chaotic balanced state in a model of cortical circuits. *Neural computation*, 10(6):1321–1371, 1998.

[69] C. Eliasmith. A unified approach to building and controlling spiking attractor networks. *Neural computation*, 17(6):1276–1314, 2005.

[70] G. Hennequin, T. P. Vogels, and W. Gerstner. Optimal control of transient dynamics in balanced networks supports generation of complex movements. *Neuron*, 82(6):1394–1406, 2014.

[71] S. W. Keemink and C. K. Machens. Decoding and encoding (de) mixed population responses. *Current Opinion in Neurobiology*, 58: 112–121, 2019.

[72] F. Mastrogiuseppe and S. Ostojic. Linking connectivity, dynamics, and computations in low-rank recurrent neural networks. *Neuron*, 99(3):609–623, 2018.

[73] T. Hromádka, M. R. DeWeese, and A. M. Zador. Sparse representation of sounds in the unanesthetized auditory cortex. *PLoS biology*, 6(1):e16, 2008.

[74] A. Löhne and B. Weißing. The vector linear program solver bensolve–notes on theoretical background. *European Journal of Operational Research*, 260(3):807–813, 2017.

Supplementary material

$K \leftarrow \{k \mid k \in \mathbb{N} : 1 \leq k \leq N\}$ // all neurons
initialise $V_k(0)\ \forall\ k \in K$
for $t = 0$ *to* t_{max} *in steps* Δt **do**
  $R \leftarrow \left\{ k \mid k \in K : t - \arg\max_{t' < t} \left(s_k(t') = 1\right) < \tau_{\text{ref}} \right\}$ // in refraction
  $C \leftarrow \{k \mid k \in K \backslash R : V_k(t) > T_k(t)\}$ // spike candidates
  while $C \neq \emptyset$ **do**
    $w \leftarrow \arg\max_{k \in C} \left(V_k(t) - T_k(t)\right)$ // furthest above threshold
    $s_w(t) \leftarrow 1$ // spike
    $\mathbf{V}(t) \leftarrow \mathbf{V}(t) - \mathbf{D}^{\mathsf{T}}\mathbf{D}_w$ // instant recurrence
    $R \leftarrow R \cup \{w\}$ // refraction
    $C \leftarrow \{k \mid k \in K \backslash R : V_k(t) > T_k(t)\}$ // spike candidates
  end
  sample $\boldsymbol{\eta}(t) \sim \mathcal{N}(\mathbf{0}, \sigma_V \mathbf{I})$
  $\mathbf{V}(t + \Delta t) \leftarrow \mathbf{V}(t) + \Delta t \left(-\lambda \mathbf{V}(t) + \lambda \mathbf{D}\mathbf{x}(t)\right) + \sqrt{\Delta t}\ \boldsymbol{\eta}(t)$
end

Supplementary Algorithm 1: Numerical implementation of a general *SCN* with voltage noise σ_V and refractory period τ_{ref}.

$K \leftarrow \{k \mid k \in \mathbb{N} : 1 \leq k \leq N\}$ // all neurons
initialise $V_k(0)\ \forall\ k \in K$
$\boldsymbol{\Omega} = \mathbf{D}^\mathsf{T}\mathbf{D}$ // standard recurrent matrix
if $\theta > 0$ **then**
  $\boldsymbol{\Omega}^f = \mathrm{diag}(\boldsymbol{\Omega})$ // instant self-reset vector
  $\boldsymbol{\Omega}^\theta = \boldsymbol{\Omega} - \mathrm{diag}(\boldsymbol{\Omega}^f)$ // delayed recurrence matrix
end
for $t = 0$ *to* t_{max} *in steps* Δt **do**
  sample $\boldsymbol{\Omega}^*(t) \leftarrow \boldsymbol{\Omega} + \Delta\boldsymbol{\Omega}(t)$ // synaptic noise
  if $\theta > 0$ **then**
    $\boldsymbol{\Omega}^f = \mathrm{diag}(\boldsymbol{\Omega}^*(t))$ // instant self-reset vector
    $\boldsymbol{\Omega}^\theta = \boldsymbol{\Omega}^*(t) - \mathrm{diag}(\boldsymbol{\Omega}^f)$ // delayed recurrence matrix
  end
  $R \leftarrow \left\{k \mid k \in K : t - \arg\max_{t' < t}\left(s_k(t') = 1\right) < \tau_{\mathrm{ref}}\right\}$ // in refraction
  $C \leftarrow \{k \mid k \in K \backslash R : V_k(t) > T_k(t)\}$ // spike candidates
  while $C \neq \emptyset$ **do**
    $w \leftarrow \arg\max_{k \in C}\left(V_k(t) - T_k(t)\right)$ // furthest above threshold
    $s_w(t) \leftarrow 1$ // spike
    if $\theta > 0$ **then**
      $V_w(t) \leftarrow V_w(t) - \boldsymbol{\Omega}^f_w$ // instant self-reset
    else
      $\mathbf{V}(t) \leftarrow \mathbf{V}(t) - \boldsymbol{\Omega}^*_w$ // instant recurrence
    end
    $R \leftarrow R \cup \{w\}$ // refraction
    $C \leftarrow \{k \mid k \in K \backslash R : V_k(t) > T_k(t)\}$ // spike candidates
  end
  $\Delta\mathbf{V} = \Delta t\left(-\lambda\mathbf{V}(t) + \lambda\mathbf{D}\mathbf{x}(t)\right)$ // normal dynamics
  sample $\boldsymbol{\eta}(t) \sim \mathcal{N}(\mathbf{0}, \sigma_V\mathbf{I})$
  $\Delta\mathbf{V} \leftarrow \Delta\mathbf{V} + \sqrt{\Delta t}\,\boldsymbol{\eta}(t)$ // current noise
  $\Delta\mathbf{V} \leftarrow \Delta\mathbf{V} + \Delta t\,\mathbf{p}(t)$ // optogenetic currents
  if $\theta > 0$ **then**
    $\Delta\mathbf{V} \leftarrow \Delta\mathbf{V} - \boldsymbol{\Omega}^\theta\mathbf{s}(t + \Delta t - \theta)$ // delayed recurrence
  end
  $\mathbf{V}(t + \Delta t) \leftarrow \mathbf{V}(t) + \Delta\mathbf{V}$
end

Supplementary Algorithm 2: Numerical implementation of a general *SCN* with finite delays θ, refractory period τ_{ref}, current noise σ_V, time-varying synaptic noise $\Delta\boldsymbol{\Omega}(t)$ and time-varying optogenetic currents $\mathbf{p}(t)$.

initialise $T \leftarrow T_{\min} > 0$ // current box width
initialise $T^* \leftarrow 0$ // best box width so far
initialise $k \leftarrow 0$ // trial counter
while $k < K$ **do**
  $k \leftarrow k + 1$
  simulate SCN with N neurons and box width T
  for $1 < j \leq N$ **do**
    $\Theta_j \leftarrow \{t \mid s_j(t) = 1\}$ // spike times
    $S_j \leftarrow \left\{t - t' \mid t, t' \in \Theta_j \ \wedge \ t = \arg\min_x (x > t')\right\}$ // intervals
  end
  $S \leftarrow \bigcup_{j=1}^{N} S_j$ // pool interspike intervals
  $A \leftarrow \{a \in S \mid 2\theta - \epsilon < a < 2\theta + \epsilon\}$ // SISIs near double-delay
  $P \leftarrow \frac{|A|}{|S|} > \gamma$ // Boolean: ping-pong present?
  if P **then**
    if $w^* > 0$ **then**
      $w \leftarrow T^*$ // use previous estimate...
      $k \leftarrow K$ // ...and quit
    else
      $T \leftarrow \alpha T$ // increase box size
      $k \leftarrow 0$ // restart trial counter
    end
  else if $k = N$ **then**
    $T^* \leftarrow w$ // update best estimate
    $T \leftarrow \beta T$ // slightly decrease box size
    $k \leftarrow 0$ // restart trial counter
  end
end

Supplementary Algorithm 3: Numerical search for the "safe width" of a bounding box, avoiding ping-pong. Typical parameters are $T_{\min} = 0.55$, $\alpha = 1.5$, $\beta = 0.95$, $\gamma = 0.1$, $\epsilon = 0.05 \cdot 2\theta$, $N = 100$. In each trial, all neurons j have the same threshold T_j, and the box is thus widened or narrowed symmetrically.

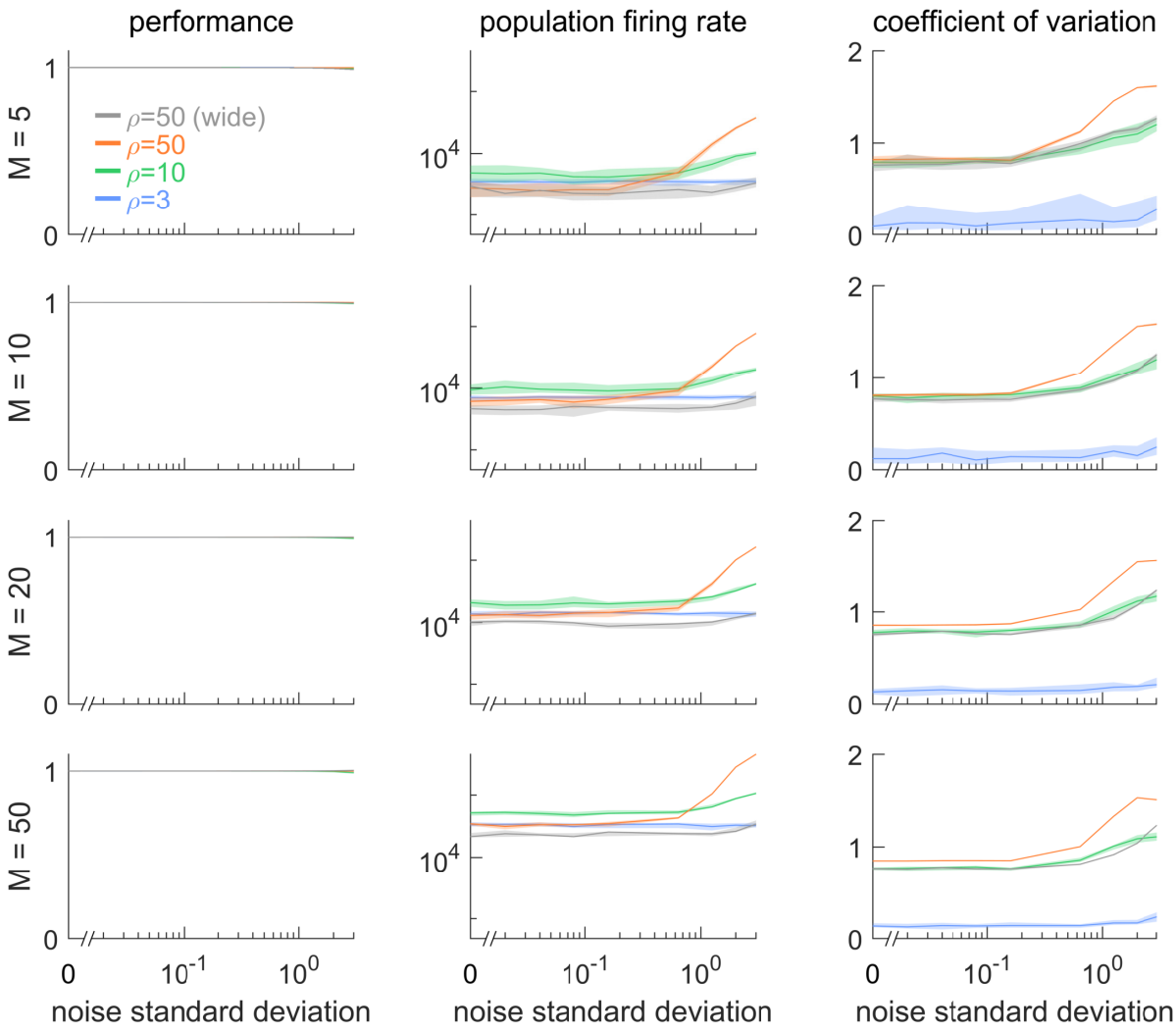

Figure 6–Figure supplement 1. Robustness to noise for different signal dimensionalities. Comparison of SCN's robustness to noise for different signal dimensionalities ($M = 5$, 10, 20, and 50). Network performance relative to an identical reference network without noise (left), population firing rate (middle), and the average (across neurons) coefficient of variation of the interspike intervals (right). Overall, dimensionality does not qualitatively affect robustness to noise. ρ is the redundancy, with $\rho \in \{3, 10, 50\}$. Threshold is $T = 0.55$ by default, unless labelled 'wide', which corresponds to an expanded threshold of $T = 1.0$. Lines show medians, and shaded regions indicate interquartile ranges.

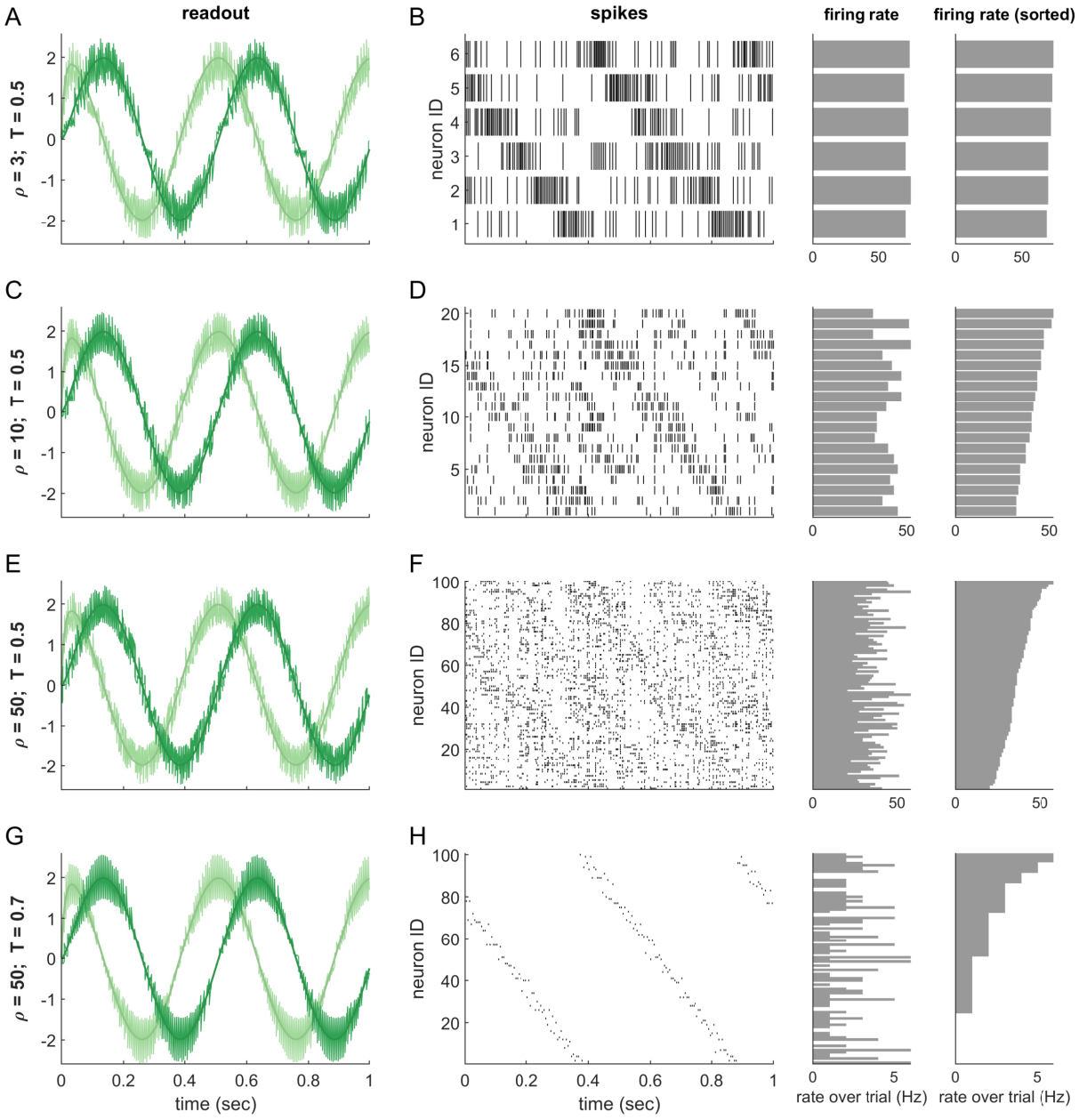

Figure 6–Figure supplement 2. Impact of voltage noise on spike trains and decoded (two-dimensional) signals for networks of different size and box width, as shown in **Figure 6**C. For the sake of clarity, almost uniformly distributed decoders were chosen, as in **Figure 6*A-C***. (**A-B**) redundancy $\rho = 3$ and minimal box width, (**C-D**) redundancy $\rho = 10$ and minimal box width $T = 0.5$, (**E-F**) redundancy $\rho = 20$ and minimal box width, (**G-H**) redundancy $\rho = 50$ and a 40% wider box $T = 0.7$. (**A,C,E,G**) Readout (green lines) and readout target (thin grey lines). Each sinusoid represents one of the two signal dimensions. (**B,D,F,H**) Spike raster plots for all neurons in the network, sorted by decoding weights, from first to last recruited (left). On the right are the firing rates of individual neurons in the same order (centre), as well as sorted from largest to smallest (right).

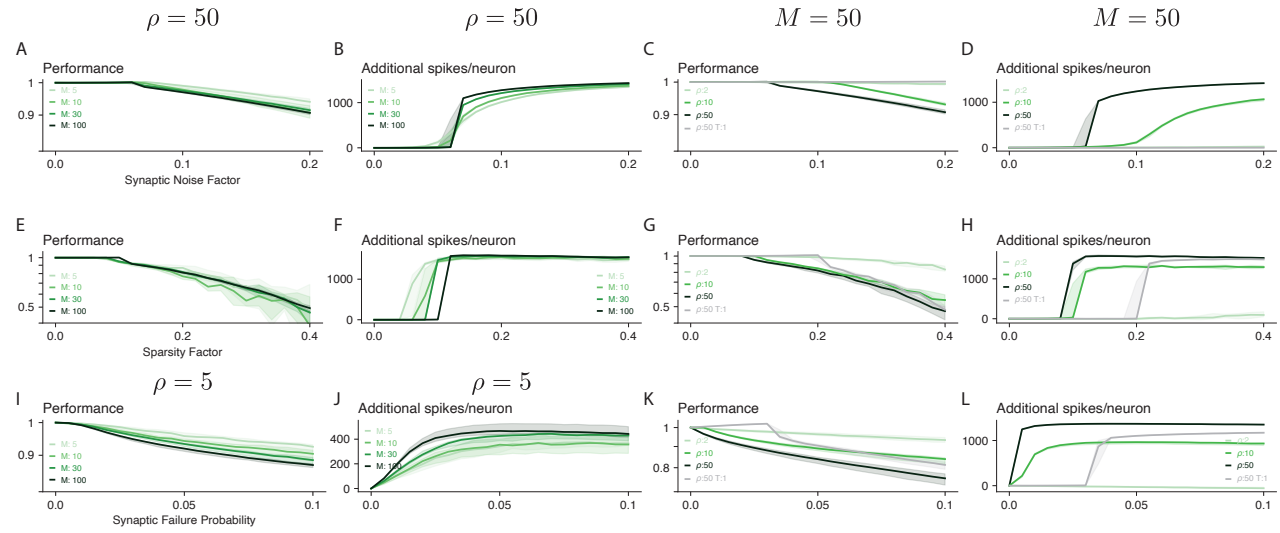

Figure 7–Figure supplement 1. Robustness of *SCN*s for different types of synaptic noise. (**A-D**) Network robustness for time-varying synaptic noise. Here, the synaptic noise factor defines the standard deviation of the multiplicative noise term (see Material and Methods). (**E-H**) Network robustness when varying the sparsity factor. A sparsity factor of 0.2 means that the 20% weakest synapses are truncated at 0. (**I-L**) Network robustness when varying the probability of synaptic failure. Synaptic failure probability of 0.05 means that 5% of spikes passing through a synapse are ignored. (**A,E,I**) Network performance for different dimensionalities. (**B,F,J**) Additional spikes per neuron for different dimensionalities. In (**A,B,E,F**), redundancy ρ is 50 and in (**I-J**) redundancy is 5. (**C,G,K**) Network performance due to the synaptic manipulation for different redundancies and error tolerances (dimensionality $M = 50$). (**D,H,L**) Additional spikes per neuron due to the synaptic manipulation for different redundancies (dimensionality 50).

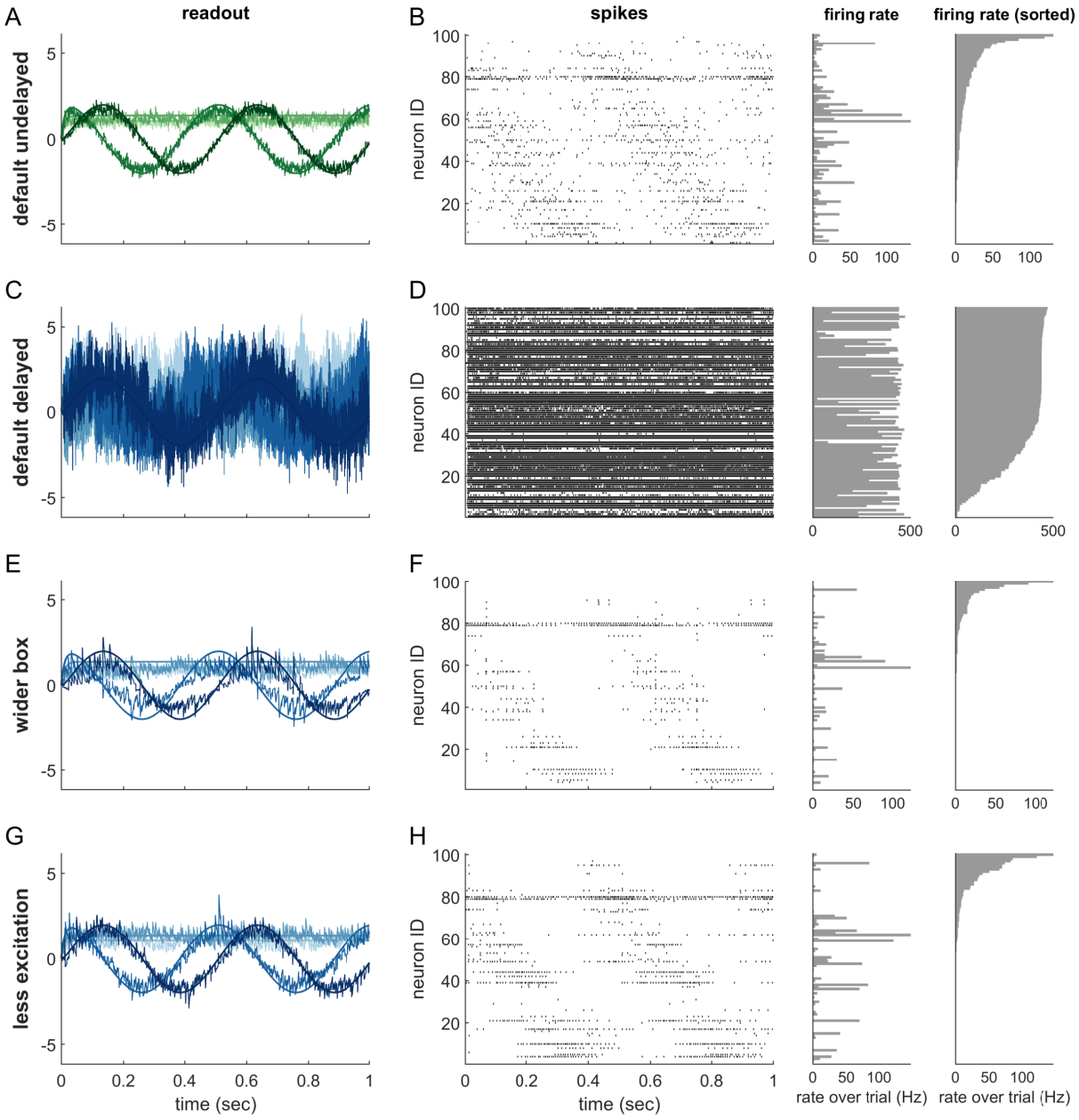

Figure 8–Figure supplement 1. Single trials of delayed *SCN*s at medium dimensions (2-dimensional circular signal in 20 dimensions, redundancy 5). (**A,B**) Undelayed fully connected network with a default box of $T = 0.55$, (**C,D**) delayed fully connected network with a default box, (**E,F**) delayed fully connected network with optimally widened box (see ***Figure Supplement 2A***), (**G,H**) delayed network with default box and optimally reduced excitation (see ***Figure Supplement 2B***). (**C-H**) Delay is $\theta = 1$ms. Panels (**A,C,E,G**) show the readout in each of the first four signal dimensions as a separate line. Dimensions 5 to 20 are hidden to avoid clutter. Panels (**B,D,F,H**) show corresponding spike-time raster plots (left) and trial-averaged single-neuron firing rates (centre), as well as the same rates ordered from largest to smallest (right).

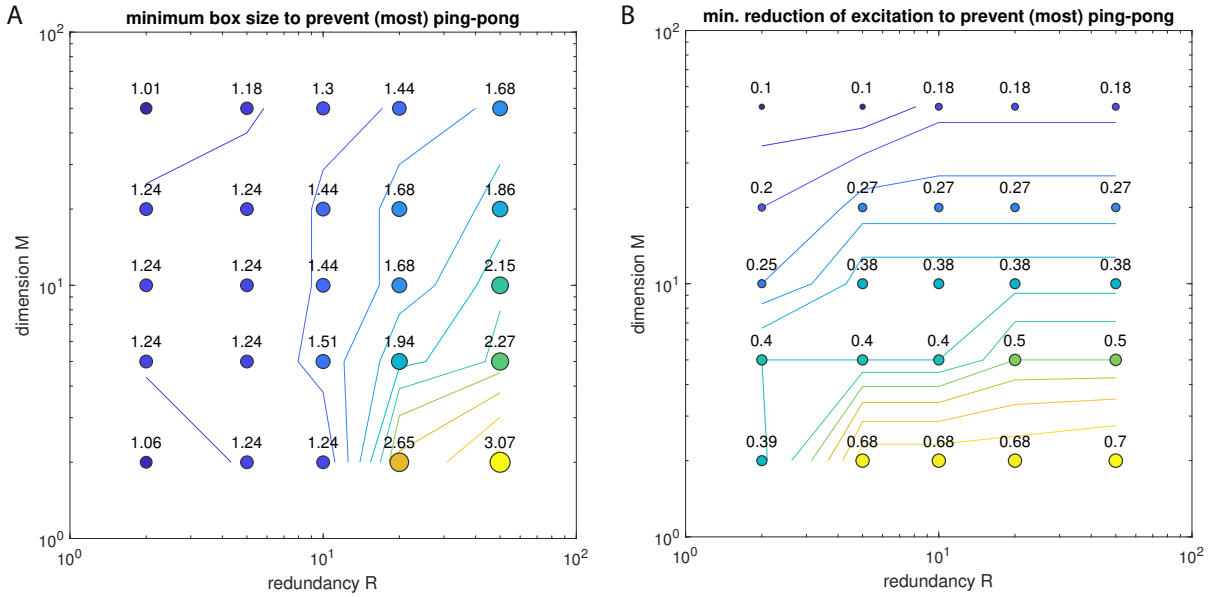

Figure 8–Figure supplement 2. Empirically learned changes needed to avoid ping-pong in delayed *SCN*s with synaptic delays of $\theta = 1$ms. (**A**) Minimum box size. (**B**) Minimum fraction of pairwise excitatory connections to remove, in the order of increasing scalar product between the connected decoders (i.e., beginning with the strongest antipode and gradually including other neurons neighbouring the largest antipode).

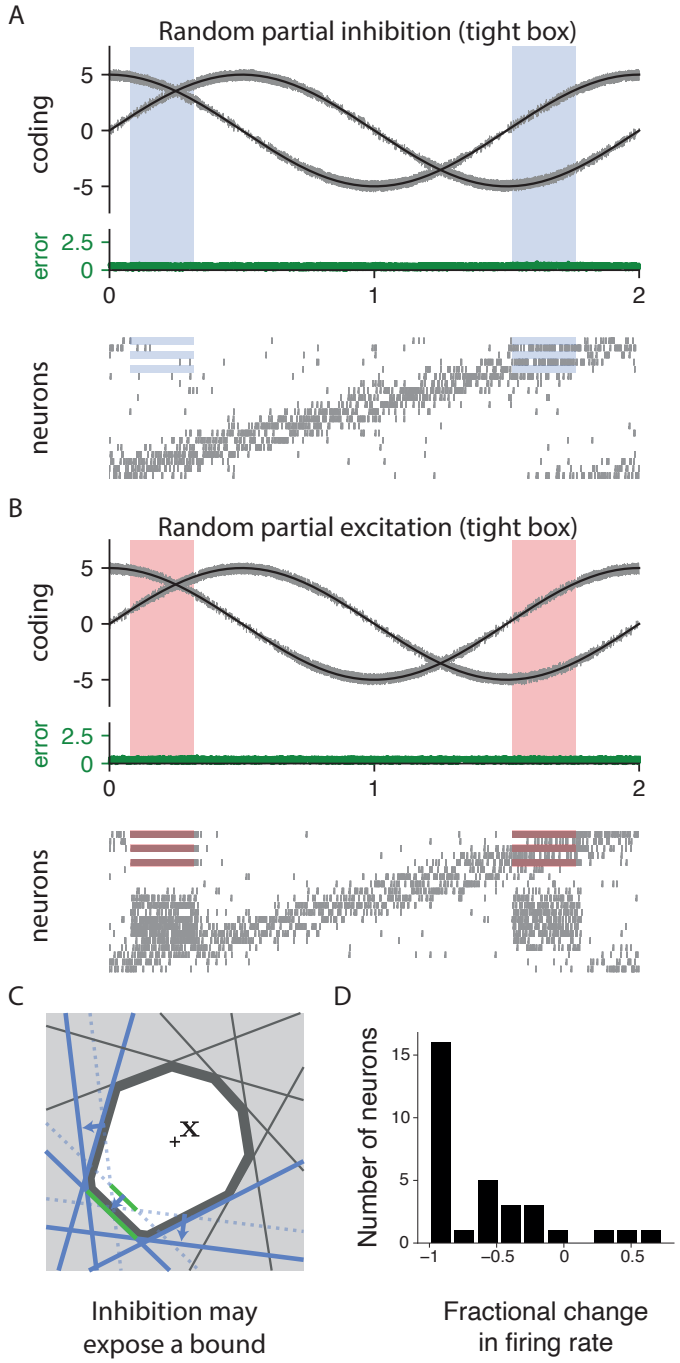

Figure 9–Figure supplement 1. Simulations of random partial inhibition and excitation with tighter box (thresholds of 0.55), and paradoxical effect of optogenetic inhibition. (**A,B**) Simulation of *SCN*s response to random partial optogenetic perturbations with a tight box ($T = 0.55$). In this case we observe no coding bias but a strong network response (ping-pong) for the excitatory perturbation. (**C**) Inhibited neurons may have their bounds contribute with a larger surface of the box (in green) and thus potentially have higher firing rates. (**D**) Fractional change in firing rate for an example simulation. Note that most neurons decrease their firing rate but a small subset increase their activity despite being inhibited.