Temporal computations in recurrent neural networks

Manuel Beiran

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à École Normale Supérieure

Temporal computations in recurrent neural networks

Soutenue par
Manuel BEIRAN
Le 03 décembre 2020

École doctorale n°158
Cerveau, Cognition, Comportement

Spécialité
Neurosciences

Composition du jury :

Julijana GJORGJIEVA Max Planck Institute	*Rapporteur*
Jaime DE LA ROCHA IDIBAPS	*Rapporteur*
Virginie VAN WASSENHOVE CEA NeuroSpin	*Membre du jury*
Mehrdad JAZAYERI MIT	*Membre du jury*
Vincent HAKIM École Normale Supérieure	*Président du jury*
Srdjan OSTOJIC École Normale Supérieure	*Directeur de thèse*

ENS | PSL

Preface

This thesis gathers the work I carried out in the last three years as a doctoral student at École Normale Supérieure in Paris. Numerous people have contributed to the work in this journey and I would like to acknowledge them here.

I would like to deeply thank my supervisor Srdjan Ostojic for his generous guidance and careful mentoring. His scientific engagement, honesty, and drive for understanding, together with his exceptional kindness and support, have taught me more than I could ever expect, both academically and personally. Secondly, I would like to thank Mehrdad Jazayeri, jury member and co-supervisor of the last part of this thesis. The regular virtual meetings opened me the door to a new stimulating world of research, and encouraged me to always think one step beyond. I would like to extend this appreciation to the whole Jazlab for hosting me, in particular to Alexandra Ferguson, a wonderful collaborator, always ready to share and discuss new ideas.

I would like to thank as well to the other jury members, for devoting their time and knowledge to this project, as well as Matthew Chalk and Romain Brette, for their encouraging feedback in the yearly thesis committee meetings. I express my sincere gratitude to the École des Neurosciences de Paris (ENP), who provided me with financial, logistic and mentoring support to accomplish this work, in particular to Yvette Henin, who set up such an excellent PhD program.

I would like to thank Francesca Mastrogiuseppe, whose remarkable findings with Srdjan have been an endless source of inspiration for this work, but, more importantly, I would like to thank her for her exceptional goodness, scientific rigor, and generosity; which have left an even bigger trace. I would like to thank my other fellow PhD students Giulio Bondanelli and Adrian Valente, admired friends from whom I learned something new every day; as well as all the wonderful fellow team members: Rupesh Kumar, Josef Ladenbauer, Alexis Dubreuil, Ljubica Cimesa, Joao Barbosa and all the undergraduadate students that joined along this journey. Every day I felt extremely fortunate to bike to the lab to learn with such an extraordinary team. It has been a real pleasure. Special thanks to Adrian and Joao for proofreading parts of this dissertation. I would like to thank also all present and former members of the Group for Neural Theory, for creating together a very welcoming and warm atmosphere, ideal for discussions and distractions.

I wish to acknowledge Mies O'Ward for his unconditional support and presence during the writing of an important part of this thesis, and Emmanuel and Pauline who made it possible.

I cannot conclude without mentioning my dear friends, especially those who were far these years but felt very close, to the family I found in France, to my family, and to Adeline. For their enduring love.

October 2020

Contents

Introduction I

La musique est le salaire que l'homme doit au temps.
Plus précisément : à l'intervalle mort qui fait les rythmes.

— Pascal Quignard, *La haine de la musique*

Music is what man owes to time.
More precisely: to the dead interval that produces rhythms.

— Pascal Quignard, *The Hatred of Music*

I.1 Biological substrate for neural computations

Humans perceive objects, feel emotions, generate complex thoughts, coordinate movements, store and recall past memories, make decisions and plan strategies. Such sensory, cognitive and motor processes that produce adaptive behavior with the environment are embodied in the neural tissue of the brain.

Pioneering studies starting at the end of the 19th century discovered that the neural tissue is formed, among other cells, by separate individual cells called neurons, which show electrical excitability and are connected to each other (Ramón y Cajal, 1909). Since then, the physiology of neurons and their connections –synapses– has been extensively studied. All across the outer layer of the brain, the cerebral cortex, neurons are similarly organized. Cortical neurons are organized into horizontal layers and grouped vertically in columns that are recurrently connected with their surrounding columns (Fig. I.1A). However, different areas of cortex receive inputs from different cortical and subcortical structures, and are responsible for different functions. Broadly speaking, cortex can be functionally categorized into sensory, motor and association areas. This flexible functionality on an extended neural tissue with common anatomical features have led to hypothesize that the network structure of cortical neurons is the biological substrate of the computations that the brain performs. An essential question in neuroscience consists in describing the map between a specific behavior or neural computation and the activity of interconnected cortical neurons.

Neurons interact with each other by firing stereotyped fast electrical discharges, referred to as spikes or action potentials, that are then felt by their connecting neurons. Influenced by the view that neurons are the basic structural and functional units in the brain, the first recordings of *in vivo* activity in cortical areas studied the link between the environment (i.e., external stimuli) and the activity of single neurons (measured as the number of spikes

elicited in a certain temporal window). Starting in the 1950s, David Hubel developed a microelectrode able to record the action potentials of single neurons *in vivo* (Hubel, 1957). Together with Torsten Wiesel, this technique allowed them to identify the features in the visual field that elicited a response in cortical neurons (Hubel and Wiesel, 1959, 1962), such as the location or shape of the visual stimulus. This mapping was formalized by the concept of receptive fields or tuning curves: a mathematical function that maps certain parameters of the inputs to the firing rate of single neurons (Fig. I.1B). Similar receptive fields have been characterized in other sensory cortices such as the spectrotemporal receptive fields of neurons in primary auditory cortex (Aertsen and Johannesma, 1981), and the receptive fields of somatosensory cortex, that links the touch in different body locations to the neural responses (Gardner, 1988).

Receptive fields in upstream cortical areas are sensitive to more abstract features of the environment. For instance, in inferior temporal cortex, which receives input from higher visual areas such as V4, and in the temporo-occipital cortex, neurons are rather insensitive to simple features such as location in the visual field and stimulus size but respond instead to complex properties such as particular combinations of shape and color (Gross et al., 1972; Gross, 1992). In higher cortical areas, the tuning of single neurons has been found to depend not only on sensory parameters but also on cognitive variables, such as memory demands. In non-human primates' prefrontal cortex (PFC), the cortical area responsible for many forms of executive control, some neurons show increased firing activity when a stimulus with a given orientation is held in memory (Funahashi, 1989). In sensory discrimination tasks where monkeys compared the frequencies of two vibratory stimuli temporally separated by a time delay, the firing rate of PFC neurons was monotonically tuned during the delay (Fig. I.1C, Romo et al. (1999)). However, a detailed classification of the tuning properties of neurons in PFC in more naturalistic behavior is challenged by the fact that responses are correlated with sensory stimuli, task rules, motor responses and any possible combination of these (Rigotti et al., 2013).

Advances in neural recording techniques have also allowed researchers to record action potentials from multiple cortical neurons simultaneously, starting in the 1990s with the introduction of tetrodes (Gray et al., 1995) that could record from tens of neighboring neurons simultaneously, to the more recent Neuropixel electrodes that can record thousands of neurons in multiple brain areas (Steinmetz et al., 2018), together with the development of imaging techniques that can track other proxies of single cell neural activity (Dombeck et al., 2007; Chen et al., 2013). Multi-unit recordings, together with algorithmic progress in spike sorting, were able to reduce the selection bias of single-neuron physiological studies, where neurons that are more responsive to the task are more likely to be identified. This new picture has broadened the view on the huge variability in neural responses that is observed in cortical areas *in vivo*. These results, enabled by the constant progress in recording techniques, together with theoretical studies focusing on the emergent properties of neuronal networks at the collective level, questions the notion that individual neurons are the functional units for computation in the brain and propounds that computations emerge in cortical areas at the level of local neural networks (Yuste, 2015).

One aspect that is often overlooked when correlating environmental features to the response of individual neurons is their complex dynamics. Often, tuning curves map the time-averaged response of neurons to a given set of stimuli, i.e., the mean firing rate during the stimulus presentation. Such simplifications leave aside the temporal heterogeneity in neural responses. This heterogeneity can be partly explained by the intricate recurrent connections present in cortical networks (Sompolinsky et al., 1988; Brunel, 2000). Moreover, neural responses show a large variability to the same stimulus in repeated trials (Mainen and Seinowski, 1995), which is also averaged out in traditional tuning curve analyses. In recordings of multiple neurons, the noise-correlations, i.e., the trial-to-trial variability independent of the stimulus, are affected by learning experience and attention, giving support to their role in brain computations (Cohen and Maunsell, 2009; Ni et al., 2018). Furthermore,

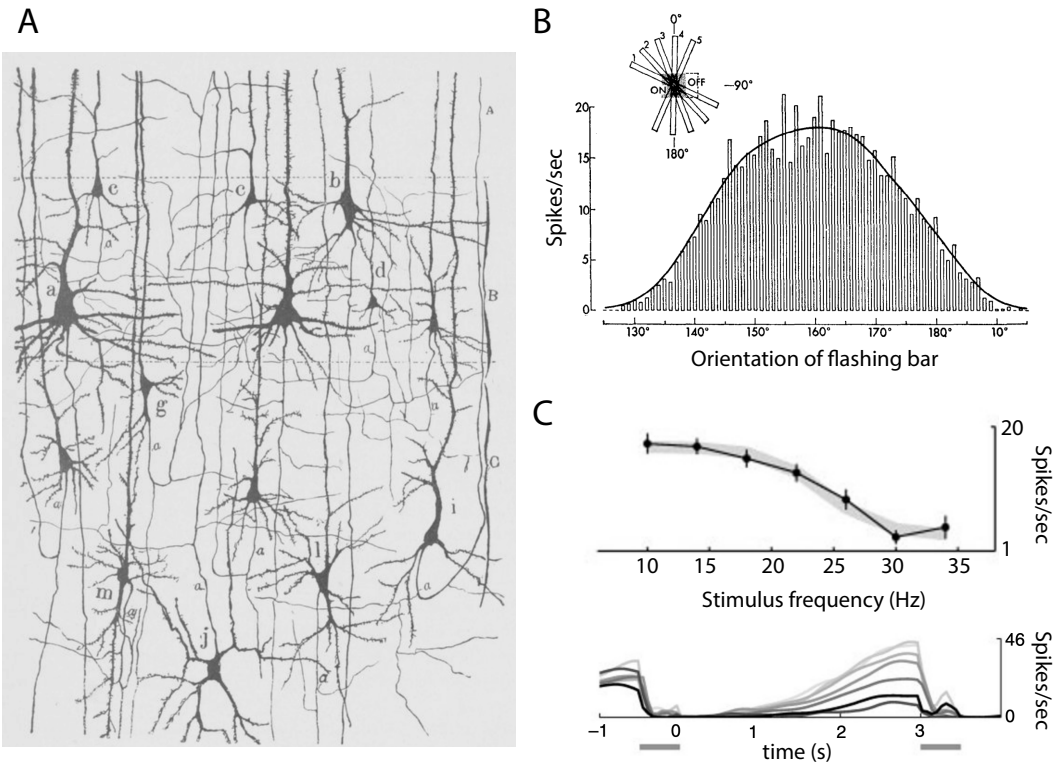

FIGURE I.1: **Cortical neurons and tuning curves in cortical networks**. **A** Cortical neurons in deep layers of human visual cortex. Drawing from Ramón y Cajal (1909), based on Golgi's staining method. **B** Tuning curve of one cell in cat's visual cortex mapping the orientation of a flashing bar to the firing rate of the neuron. Each bar corresponds to the firing rate averaged over 13 different trials, each trial being 500ms long. Adapted from Henry et al. (1974) **C** Top. Tuning curve of the averaged firing rate of one neuron in prefrontal cortex, in a two-interval forced choice task using vibratory stimuli, during the three-second delay. The tuning curve measures the average firing rate during the delay for seven different frequencies of the first stimulus. Bottom. Temporal profile of the instantaneous firing rate of the neuron at different time points of the trial. Adapted from Romo et al. (1999). The tuning curve does not take into consideration the ramping profile of the firing activity of this neuron.

when task parameters are decoded from the joint activity of multiple neurons, stimulus features can be better assessed than considering separate single neurons (Rigotti et al., 2013; Fusi et al., 2016). Such recents findings point towards an integrative view of cortical computations, where local networks must be taken into account to understand computations (Saxena and Cunningham, 2019).

The work presented in this thesis is grounded on a theoretical framework by which neural computations -representations from the outer world, cognitive variables, motor commands- are based on the temporal dynamics of the joint activity of ensembles of neurons. In this Chapter, we first briefly introduce the basic concepts the mathematical framework, coined as computation-through-dynamics (Vyas et al., 2020; Remington et al., 2018b; Horio and Aihara, 2008). Secondly, we define the computations that are object of this study: temporal computations, with special emphasis on recent work focusing on neural mechanism for flexible timing tasks. In particular, we summarize recent experimental findings in timing

tasks that motivate part of this work. In the last section, based on the described theoretical and experimental findings, we outline the structure of this thesis.

I.2 Neural computations through dynamics

Cortical networks can be described mathematically as dynamical systems. This approach takes into account the interactions between neurons, the cellular biophysics of single neurons, and their responses to external inputs from other brain areas.

A dynamical system is defined by a set of state variables, that can be jointly represented as a multidimensional vector at any point in time $\mathbf{x}(t)$. These variables account for the time-dependent features that model the neural network. The temporal derivatives of the state variables are defined by a function f, often non-linear, that receives as arguments the *state variables* themselves, and possibly, some external inputs $\mathbf{I}(t)$:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t), \mathbf{I}(t)). \tag{I.1}$$

To build an intuition of a dynamical, we can think of the dynamics of an ideal pendulum, as illustrated in Vyas et al. (2020). The state of a pendulum is defined by its angular speed v and its angular position p, so that the state variables are defined by vector $\mathbf{x} = (p, v)$. In the absence of any external perturbations, when the input $I(t)$ is zero at all times, the dynamics are called autonomous, and given by the motion equation

$$\frac{d\mathbf{x}}{dt} = \begin{cases} \frac{dp}{dt} = v \\ \frac{dv}{dt} = -\sin p. \end{cases} \tag{I.2}$$

The state-space of the pendulum is therefore a two-dimensional system, given by axes p and v. If the initial conditions at a given time point are given, it is possible to solve the dynamical equation integrating over time, to fully determine the trajectory of the pendulum, i.e., the series of states (positions and speeds) that the pendulum will go through. This *trajectory* represents a curve in the two-dimensional *state-space*, that is parameterized by time. Different initial conditions generally generate different trajectories. One common way to visualize the possible trajectories of a two-dimensional dynamical system consists of plotting the *phase portrait* or *flow field* in state-space: a dense representation of the trajectories that can be generated in neural space. The flow field of the pendulum displays two special points in state space where the dynamics of the state vector are zero: at $(p = 0, v = 0)$ and $(p = \pm\pi, v = 0)$, which correspond to the pendulum located with zero velocity in the vertical direction, either in the bottom or the top. These special points are called *fixed points*, $\mathbf{x_0}$. Fixed points can be either stable or unstable, depending on whether small perturbations at the fixed point decay back to its initial state or are amplified towards a different state. Generally, the stability of the fixed points can be analyzed by approximating the dynamical system (Eq. I.1) close to the fixed point $\mathbf{x_0}$ up to the linear order:

$$\frac{d\mathbf{x}(t)}{dt} = A\mathbf{x} + B\mathbf{I} \tag{I.3}$$

where A and B are matrices that depend on the particular fixed point and the motion equation (Eq. I.2). Matrix A is denoted the Jacobian, and its eigenvalues carry information about the stability of the fixed point. If the real parts of all eigenvalues are negative, the fixed point is stable. If the Jacobian has at least one eigenvalue with positive real part, the fixed point is not stable, since small perturbations in the direction of the associate eigenvectors will be amplified as time evolves. Eigenvalues with value zero or imaginary eigenvalues lead to marginally stable dynamics close to the fixed point, so that the stability of the fixed point must be established based on additional analysis tools. In the pendulum,

the fixed point corresponding to the top position happens to be unstable, whereas the fixed point corresponding to the bottom position is marginally stable, because small perturbations around it will not move away from the fixed point nor decay back to it.

The study of a dynamical system is not restricted to the analysis of the existence and stability of fixed points. The pendulum for example can produce oscillations: closed trajectories in state space where the dynamics $f(\mathbf{x})$ are never zero. Such curves in state space are called *limit cycles*. Other types of trajectories can also occur. For instance, if the pendulum is initiated very close to the top position, with a finely tuned velocity, it is possible to make the pendulum rotate fully just one time around the clock and stop at the top position. This behavior corresponds to an *orbit*, a non-trivial trajectory in state-space that starts and ends at a fixed point.

External inputs can interfere in different ways with the autonomous behavior of a dynamical system. In particular, the trajectories produced by a specific input depend on the state of the dynamical system when the input is received. For example, if a pendulum receives a kick when it lies still in the bottom position, it will produce a trajectory that is different to the trajectory when the pendulum receives the exact same kick when it lies still in the top position. Therefore, we can conclude that the response of dynamical systems to inputs is *state-dependent*. In the case of inputs that are constant in time, it is possible to consider them as part of the function f that determines the dynamics, altering the flow field in state-space.

The pendulum is a two-dimensional dynamical system. Dynamical systems with more than two dimensions can generate, apart from fixed points and cycles and orbits, chaotic behavior. *Chaotic dynamics* is a property of dynamical systems in which two trajectories with nearby initial conditions diverge from each other in state-space exponentially fast as time evolves. Chaos is an ubiquitous feature of high-dimensional non-linear dynamical systems. For instance, a double pendulum (one pendulum hanging below a first pendulum) is a dynamical system with four state variables, that produces chaotic behavior.

So far, we have described dynamics that are deterministic; dynamical systems initiated at a fixed given state will always give rise to the same trajectory in state-space. Nevertheless, when dynamical systems are used to model biological processes, it is common to add noise:

$$\frac{d\mathbf{x}(t)}{dt} = f(\mathbf{x}(t), \mathbf{I}(t)) + \xi(t), \tag{I.4}$$

where ξ is defined as a multi-dimensional random process, that can be defined at the level of its spatial correlations and temporal statistics. Noise is usually considered to be random perturbations that are not explicitly modeled in the system, and accounts for the large variability present at other levels of descriptions of the biological process. Dynamical systems subject to noise are stochastic dynamical systems. Such stochastic dynamical system will generate different trajectories even when initiated at the same initial condition. However, stochastic dynamics are not necessarily chaotic: two trajectories with similar initial conditions can produce different trajectories, but they do not necessarily diverge from each other.

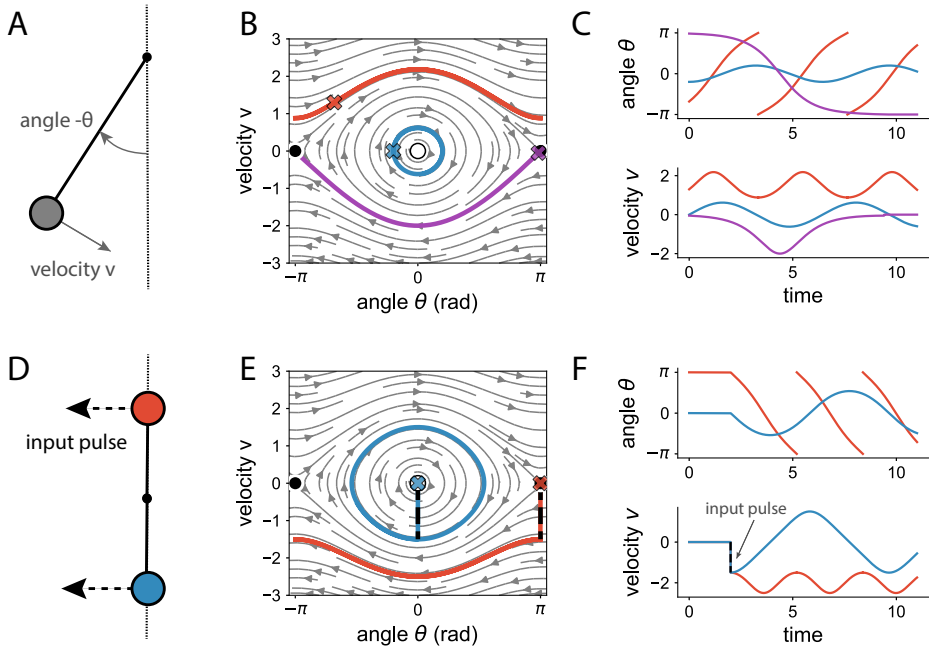

Figure I.2: **Dynamical system: the pendulum**. **A** The state of a pendulum is defined by two variables, the angle θ, that determines its position in space, and its angular velocity v. Therefore, the state-space of the pendulum has dimension two. The corresponding dynamics is given by Eq. (I.2). **B** In 2D dynamical systems, we can plot the flow field, indicating the possible trajectories in state-space. The marginally stable fixed point is filled in black, the unstable fixed point in white (note the periodic boundaries for the angle, constrained between $-\pi$ and π). The colored curves correspond to three different trajectories with initial conditions given by the crosses. The red trajectory corresponds to a limit cycle, where the pendulum always oscillates in the same direction. The blue curve is a limit cycle, where the pendulum oscillates around the marginal stable fixed point, alternating positive and negative velocities. The purple curve corresponds to a cycle: a trajectory that starts and ends at (the vicinity) of fixed points. **C** Neural trajectories are parameterized by time. We show the projections of the trajectories in state-space from **B** onto two different directions: the axis given by the angle θ (top) and the axis given by the velocity v. We can observe that the oscillations of the red trajectory are faster than the oscillations of the blue trajectory. When we plot the flow field by showing only the possible trajectories, there is no information about the speed of the trajectories. **D** We show two different pendula, the red one initially fixed in the top position ($\theta = \pi$) and the blue one, initially fixed in the bottom position ($\theta = 0$). Both systems receive the same input pulse (a kick) at time point $t = 2$, that instantenously increases the angular velocity. **E** Given that the pendula are at different states, they generate different trajectories to the same input. The red pendulum oscillates always in the same direction after the input, while the blue pendulum oscillates back and forth around the bottom position. Dynamical systems are state-dependent, because they process inputs differently depending on their initial state. **F** Dynamics of the angle and velocity, receiving a pulse in the velocity at time $t = 2$.

I.3 State-space: cortical networks as dynamical systems

Neural recordings of cortical networks can be studied as a dynamical system, using the mathematical tools that we applied for the ideal pendulum. The recorded activity of N neurons can be thought of as a trajectory in a state space of N dimensions, where each axis corresponds to a single neuron. For instance, if no neurons fire and suddenly two neurons emit simultaneously an action potential, the neural trajectory will move from the origin along the plane spanned by the two firing neurons. Often, for the analysis of neural trajectories, instead of considering the spike trains of recorded neurons, the trajectories are calculated based on a temporally-filtered sequence of the spiking patterns, the firing rate, so that the state variables are continuous (Fig. I.3A). In actual in vivo network recordings, the firing rate activity of single neurons is highly heterogeneous, generating complex patterns in neural state-space. In this framework, computations, thought of as representation of environmental variables or cognitive processes, are a result of the voyage of neural trajectories in state-space (Buonomano and Maass, 2009). This framework, also referred to as computation-through-dynamics, extends the classical view of neuronal coding, based on time-averaged responses of neurons to a stimulus, by taking into account the large and often complex dynamics of neuronal responses.

The dimensionality of neural trajectories is given by the number of recorded neurons, which define the state variables. Using recent recording techniques, this number is usually high, ranging from tens to thousands of neurons. Nevertheless, neural trajectories often span linear subspaces whith much lower dimensionality. For instance, if all recorded neurons fired at the same time, the neural trajectory would be constrained to the straight line in neural space, given by the direction $(1, 1, \ldots, 1)$. The dimensionality of the linear subspace is the *embedding dimensionality*, and is defined as the number of cartesian coordinates (collective variables) that describe a given neural trajectory in state space. The embedding dimensionality can be estimated by applying dimensionality reduction techniques, such as principal component analysis to the neural data. Often, neural trajectories recorded under experimentally controlled conditions display an embedding dimensionality that is much lower than the number of recorded neurons.

Instead of studying single trajectories that correspond to single trials, we can consider a group of trials that share some experimental condition. This corresponds in neural space to an ensemble of trajectories. The region in neural space explored by the ensemble of neural trajectories is called a *neural manifold* (Fig. I.3B). Neural manifolds can sometimes be described by less variables than the dimensionality of the linear subspace in which they are contained. For example, if a set of neural trajectories is constrained to the surface of a 3D sphere in neural space, the embedding dimensionality of the corresponding manifold is three. However, it is possible to describe any state on the manifold with only two variables, the altitude and the azimuth on the sphere (Fig. I.3D). These variables are often called latent variables of the dynamics, and the number of latent variables defines the *intrinsic dimensionality* of a neural manifold. The intrinsic dimensionality is often hard to assess in neural recordings, and usually requires additional theoretical assumptions about the geometry of neural trajectories (Yu et al., 2009).

The quantitative analysis of neural recordings in the framework of dynamical systems can be extended beyond the assessment of the dimensionality of the neural trajectories (Vyas et al., 2020). Recently, it has been assessed whether different tasks correspond share the same embedding dimensions or not. For instance, motor activity and premotor activity concur into largely non-overlapping subspaces in motor cortices (Kaufman et al., 2014; Elsayed et al., 2016; Gallego et al., 2018). In neural recordings of neurons in visual areas V1 and V2, a similar analysis tool showed that some activity is shared across the two areas in a given subspace, whereas another fraction of the recorded activity evolved in dimensions belonging to only one of the two populations (Semedo et al., 2019). Motor and premotor cortical responses have been recently compared based on the notion of ”tangling”, how

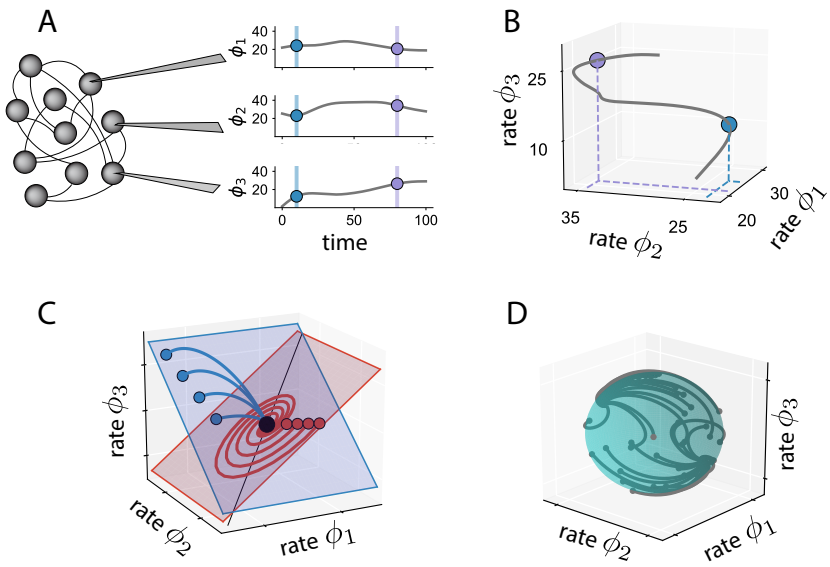

Figure I.3: **Neural trajectories, manifolds and dimensionality**. **A** Illustration of the firing rates of three units in a (biological or artificial) recurrent network. The firing rate for spiking neuron recordings can be estimated by averaging the number of spikes over a short time window. Two time points, blue and purple, are highlighted. **B** The neural state-space of such network has the dimensionality of the number of units considered. We show here the neural trajectory along three recorded neurons. Each point in time (such as the blue and purple dots) corresponds to a point in neural space, whose coordinates are given by the activity of each neuron. The collection of all time points is the neural trajectory (grey curve). **C** Illustration of eight different trajectories, grouped by two different trial conditions (blue and red). The set of trajectories for each trial condition spans a neural manifold. In this illustration, there are two neural manifolds that correspond to a linear plane, therefore, the embedding and latent dimensionality of both manifolds is two. Often, in high-dimensional neural networks, the axis correspond to linear combinations of the activity of different neurons, instead to single neuron activity. **D** Set of neural trajectories that evolve along a sphere in neural state-space. The embedding dimensionality of the manifold is three, because the sphere is a 3D object, whereas the intrinsic dimensionality is two, because any point on the sphere can be mapped by using two angles (altitude and azimuth).

much trajectories that are close to each other in state-space at an initial time point evolve through similar trajectories (Russo et al., 2018). Furthermore, it has been observed that the neural manifolds used for motor actions can be systematically accessed over several days, even if the set of recorded neurons changes (Gallego et al., 2020). Other approaches have assessed how much the embedding subspaces hosts trajectories that are invariant to time, determining the "temporal scaling" index of the subspace Remington et al. (2018a).

I.4 Recurrent neural networks

The computation-through-dynamics approach is not limited to the analysis of neural recordings. On the contrary, it offers a direct link to artificial recurrent networks, where all the state variables that define the network are known and accessible. One common way to describe recurrent neural networks is to use the following dynamics:

$$\frac{dx_i}{dt} = -x_i + \sum_{j=1}^{N} J_{ij}\phi\left(x_j\right) + I_i\left(t\right) \tag{I.5}$$

for $i = 1, \ldots, N$, where $x_i\left(t\right)$ is a continuous variable that correspond to the input received by the i-th neuron at time t. The firing rate of the neuron is then a non-linear transformation of the input $\phi\left(x_i\right)$, often a sigmoidal or threshold-linear function. The matrix element J_{ij} measures the synaptic strength of the connection from neuron j to neuron i. The function I_i represents the possibly time-dependent input that the i-th neuron is received from other brain regions. Under mild mathematical assumptions, a recurrent neural network model as in Eq. (I.5) can theoretically approximate with arbitrary precision any given function (Doya, 1993), so that it can also be used to replicate recorded activity or solve any cognitive task (Sussillo and Barak, 2013; Laje and Buonomano, 2013; Chaisangmongkon et al., 2017; Wang et al., 2018; Pinto et al., 2019; Yang et al., 2019). The unknown parameters of the network can be found in practice by applying different learning/training algorithms such as backpropagation through time (Werbos, 1990), FORCE learning (Sussillo, 2014), or more biologically-inspired algorithms such as Hebbian learning (Hebb, 1949; Gerstner and Kistler, 2002). In its final goal, this approach provides an *in silico* model of a biological neural network, allowing for inexpensive explorations of the dynamics (parameter space, response to perturbations, etc) to guide new experimental paradigms.

Nevertheless, the theoretical study of recurrent neural networks has long preceded the development of multi-unit recordings and training of large recurrent networks. The first neural network models focused on feedforward architectures, emulating the structure of early sensory systems, such as the perceptron and the multi-layer perceptron (Rosenblatt, 1962). These networks are able to perform many cognitive function, such as pattern classification and image recognition and are the basis of the state-of-the-art algorithms such as convolutional neural networks (Bengio et al., 2017). However, feed-forward architectures only deal with external information sequentially, in a way that ignores the temporal fluctuations of both the external inputs and the processing units.

In order to account for the complex dynamics of both the external stimuli and neuron activity, it is necessary to build networks with recurrent connections. Recurrent architectures consists of processing nodes that are not hierarchically organized into different layers of processing, but interact with each other, together with the inputs, at the same processing level. In Eq. (I.5), the connectivity between different neurons is described by the matrix J_{ij} and determines the interactions between neurons. In the past decades, recurrent network models have been proposed in theoretical neuroscience, to explain the dynamics observed in neural recordings (Sompolinsky et al., 1988) or to describe computations that could be implemented at the level of the whole network, such as associative memory (Hopfield, 1982).

By analogy to how a downstream area could receive inputs from a local network, a common practice is to define the output of a recurrent network as a linear combination of the activity of the individual nodes, in other words, a linear projection of (a function of) the state variables. In the recurrent network model presented in Eq. (I.5), downstream areas would likely have access only to the firing rate of neurons, $\phi\left(x_i\left(t\right)\right)$, instead of the inputs that the neurons are receiving, $x_i\left(t\right)$. Furthermore, the dynamics of a neural network can also be described by additional state variables that take into account the subcellular processes of individual neurons, such as the opening and closing of ionic channels and synapses. These state variables are defined as *hidden* state variables, in opposition to the

active state variables that can be accessed by subsequent local networks, or the experimental recordings. The interaction of these hidden neural states can have a major impact on the full network dynamics, and the processing of spatiotemporal inputs (Buonomano and Maass, 2009).

Recurrent networks can model the dynamics of the brain with different levels of biophysical detail. In a way, the interconnected neurons in the network (indicated by the subindex i in Eq. (I.5)) do not necessarily need to correspond to a single neuron in a biological network. This same equation can be applied to describe the dynamics of cortical microcolumns (ensemble of cortical neurons with shared input and output projections). For that reason, we often refer to the individual network components as network units or nodes, instead of neurons. In this work, we focus on *rate units*, where the input and output of single units are defined by a continuous function. This level of abstraction at the single unit level allows for more mathematical tractability in the analysis of the network dynamics. Theoretical research has also studied the dynamics of recurrent networks where neuron's explicitly fire action potentials, using inhomogeneous Poisson processes, integrate-and-fire-neurons or conductance-based neuron models. Recently, spiking networks have also been used as a trainable dynamical system to produce a given target dynamics (Bellec et al., 2018). The equivalence between dynamical regimes in spiking vs rate-based recurrent networks is however still a matter of debate in the field.

In this thesis, we focus on the network mechanisms that allow for solving cognitive tasks. In particular, it is still to be elucidated which are the properties of recurrent neural network, in terms of connectivity structure and biophysical processes, that allow to implement observed neural dynamics and solve tasks. We focus on the study of recurrent neural networks that are able to perform computations that require processing temporal information.

I.5 Temporal computations

Temporal computations refer to any sensory, cognitive or motor subtask that requires an explicit processing of time, either at the level of the temporal information of the inputs that cortical networks receive, or at the level of the output that these networks are required to produce. These computations are necessary for a wide variety of tasks with different levels of complexity, from estimating the duration, order or structure of stimuli, recognizing temporal patterns, to producing a motor response, engaging in a conversation or playing an instrument in an orchestra. Temporal computations require processing time over a broad range of timescales (Paton and Buonomano, 2018), ranging from milliseconds (i.e., in sound localization, speech production, motion detection), to seconds (conscious time estimation, syntax in language) and hours (appetite, sleep). In natural tasks, temporal processing of multiple timescales must be integrated simultaneously. Different biological mechanisms are used by the brain to account for such wide range. Microsecond processing for instance is largely limited by the conduction delays in sensory neurons conveying the stimulus information (Thorpe et al., 1996; Grothe et al., 2010). At the level of hours and days, circadian rhythms are controlled by molecular oscillators coupled to different physiological rhythms (Dunlap, 1999). In this thesis, we focus on timescales ranging from tens to hundreds of milliseconds. Temporal processing at such timescales is fundamental in most animal's natural behavior, and its neural basis is still an open scientific question.

Early models of temporal computations proposed the idea of a timing area in the brain, an internal clock, able to feed other brain areas with the task-specific temporal information. However, several studies, using multidisciplinary approaches (psychophysics, imagining techniques, electrophysiology) have shown that the ability of performing at least some temporal computations is present in multiple cortical and subcortical brain areas and is a prevalent intrinsic feature of neural circuits (Mauk and Buonomano, 2004). For instance, the cerebellum, a subcortical brain area traditionally described as a motor region, is involved in motor

learning, controls the fine details of motor actions and acts as a feed-forward predictor of motor outputs and sensory inputs. It is involved in learning and performing timing tasks, such as eyelid conditioning experiments or interval timing tasks (Mauk and Donegan, 1997; Heiney et al., 2014). The basal ganglia, a group of subcortical structure, are involved in motor action timing, often considered in the range of a few seconds (Mello et al., 2015). However, the relying mechanism of timing in the basal ganglia remain largely unknown, and they likely involve interactions with other brain areas (Yin, 2014). Many cortical areas are involved in temporal computations: auditory, visual, associative and motor cortices, including areas such as dorsolateral and parietal prefrontal cortex. To date, no strong tuning of single neurons to temporal features of stimuli has been found in cortical neurons, such as order or duration, which contrasts with the large sensitivity to spatial features in sensory areas, such as tuning to orientation in V1 neurons or to spectral content in auditory cortex. Nevertheless, it is often possible to decode timing information from the population activity of cortical cells (see Cueva et al. (2020)). Based on the fact that many different cortical areas are necessary to perform a wide range of timing tasks, timing is considered a general computation of cortical circuits (Paton and Buonomano, 2018).

Classical models of timing can be organized broadly into two different categories: internal clock models and spectral models using commonly oscillatory units. Internal clock models are based on two modules: (i) an internal mechanism that, similar to the tick of a clock, generates well timed responses; for example, constant ramps that are reset after reaching a threshold, and (ii) a neural integrator that counts the number of ticks (Douglas Creelman, 1962; Killeen and Fetterman, 1988). Such models however do not explain how the timing responses arise from the network connectivity. Spectral models are based on having a heterogeneous population of units where one intrinsic single unit parameter (response delays, timescales, periods of oscillation, etc) ranges over a spectrum of values (Moore et al., 1989; Grossberg and Schmajuk, 1989). Often, oscillatory units are used, that are either coupled to each other or span a wide range of different frequencies (Miall, 1989; Ahissar et al., 1997; Todd et al., 2002). Such oscillatory activity has however not been found at the level of single unit activity in many cortical areas involved in timing. Furthermore, it remains unclear how this hard-wired selectivity to different timescales can be generalized to complex temporal tasks.

In the last two decades, novel timing models have been developed based on the computation-through-dynamics framework, also called state-dependent timing models, where the collective activity of a network at different time points is mapped to different states of neural trajectories (Karmarkar and Buonomano, 2007; Buonomano and Maass, 2009). These network models posit that the voyage over different neural states contains the implicit temporal information used for solving temporal tasks, instead of having a dedicated built-in timing module. However, it remains unclear how such models can solve flexible timing tasks, where the timescales can quickly vary over different ranges.

In the last part of this thesis, we focus on flexible sensorimotor timing tasks to investigate how the computation-through-dynamics framework can solve such tasks, and what are the network mechanisms employed. This work builds on recent experimental and theoretical studies on flexible timing that we review below.

Flexible motor timing tasks

Task paradigms Recent studies have focused on studying motor timing performing flexible tasks. Classical experimental paradigms in motor timing are based on reflexive actions, such as the eye-blinking conditioning paradigms, or weakly demanding paradigms, for example, executing a motor action after a fixed delay. In commonly used sensory timing tasks, such as time interval discrimination, simple strategies like comparing the sensory information to a fixed threshold value can be used to correctly perform the task, which does not require flexibly integrating temporal information over a range of timescales (Reming-

ton et al., 2018b). New efforts have been devoted to studying the neural basis of flexible time control, where animals need to integrate contextual cues and sensory feedback on a trial-by-trial basis in order to correctly perform the task.

We present here a series of experimental paradigms that have been used to underpin the neural basis of flexible timing in non-human primate electrophysiological recordings (Fig. I.4). The logic behind this set of tasks is to start from more basic cognitive operations, and increase the level of complexity by adding additional constraints to the task. These cognitive operations can be separated into several modules: motor production, interval estimation, and combination of sensory estimates under uncertainty.

The first task here presented, the Cue-Set-Go task, focuses on understanding this control of time in movement initiation, i.e., the production epoch Wang et al. (2018). The Cue-Set-Go task requires to produce a time interval t_p after a signal is presented ('Set', Fig. I.4A). This interval production is flexible: different intervals must be produced depending on the contextual cue. In particular, the produced time interval t_p depends on a cue input, that is shown at the beginning of the trial.

A following task, the Ready-Set-Go task, added an interval estimation epoch before the production of the time interval. The expected produced interval t_p depends on the time interval t_s elapsed between two previously presented stimuli ("Ready" and 'Set') (Jazayeri and Shadlen, 2015; Remington et al., 2018a; Sohn et al., 2019). In one variant of the task (Remington et al., 2018a), the produced interval is defined as a linear function of the sampled interval, $t_p = gt_s$ where the parameter g, defined as the gain, takes different values (1 or 1.5 in the experiments) based on contextual information given at the moment of Ready (Fig. I.4B). An alternative Ready-Set-Go paradigm establishes a fixed gain $g = 1$, so that the produced interval t_p should match the sampled interval t_s of the estimation epoch but they combine two different prior distributions of the sampled intervals (Fig. I.4C, Sohn et al. (2019)). Effectively, this task is equivalent to producing the third beat of a rhythm, where the two first beats are given. Sampled intervals can be drawn from a distribution of short intervals (from 480 to 800 ms) or from a distribution of long intervals (from 800 to 1200 ms). These experiments provide information about how multiple stored priors can be flexibly combined at a trial-by-trial timescale.

Finally, an ulterior task added one more repetition of the sampled interval, the 1-2-3 Go task (Egger et al. (2019), Fig. I.4C). Here the animal has to produce the fourth beat of a rhythm, after being exposed to the first three beats, indicated by stimuli '1', '2' and '3'. In this task, the animal can improve the performance by combining the estimations from the two sample intervals. This extension of the task allows for a new cognitive computation: combining sensory information under uncertainty at the scale of a single-trial.

Behavior Monkeys were able to learn all this series of tasks after training, with slightly above-average-human performance. They could produce flexible time intervals t_p, based on cue associations or linear functions of previously estimated intervals, combining prior information about the sampling distribution, and refining their estimation when exposed to a repeated sampled interval. One remarkable feature, common to all tasks, is that longer intervals lead to a wider variability in the responses; a well studied property of timing tasks called "scalar variability" (Fig. I.5A, Malapani and Fairhurst (2002)). A second property appears consistently in all tasks that require estimating a sampled interval: the produced intervals showed a systematic deviation towards the mean of the sampling distribution (Fig. I.5Aii). This is referred to as "regression-to-the-mean", and can be explained by a normative Bayesian model, that combines the knowledge of the stimulus distribution (which is uniform and bounded between two extrema) with the noisy sensory information (Jazayeri and Shadlen, 2010).

During the production epoch, multiple brain areas evolve towards a fixed movement initiation state at the required time to produce the motor action, that is then reset (Wang

et al., 2018). One normative way to model this dynamic evolution of the neural state towards a final movement-initiation state is to used a ramp-to-threshold mechanism (Jazayeri and Shadlen, 2015). At every trial, a build-up signal evolves as a linear ramp towards a threshold, corresponding to the motor initiation (Jazayeri and Shadlen, 2015). Given that the movement-initiation state, and therefore the threshold, is fixed, it is possible to flexibly produce a time interval by modifying the average speed of this build-up signal (Fig. I.5A). Scalar variability is inherent to this behavioral model, assuming that the build-up ramp is subject to suitable noise. Flexibly controlling the timing computation consists on being able to control the speed of the ramping signal.

Single-neuron responses The firing activity of individual neurons in medial frontal cortex does not however generate ramping signals at different speeds during the production epoch (Fig. I.5B). Instead, the dynamics of single neurons are highly heterogeneous during the whole trial, in general showing a broad variety of non-monotonic responses, a ubiquitous feature of *in vivo* activity in cortical areas. Nevertheless, there is a remarkable property found in all tasks in the production epoch (the time between 'Set' and 'Go' in the CSG and RSG task, or between "3" and 'Go' in the 1-2-3 Go task): neural responses are considerably compressed or expanded in time at a single trial level to adjust to the produced interval. This is consistent with the ramp-to-threshold model, in the sense that the average speed of the dynamics controls the timing of motor initiation, although is at odds with the idea that neuronal activity, at the single neural level, provides the ramping signal. Interestingly, this speed-control of the dynamic evolution was present in downstream areas (basal ganglia), providing evidence that the mechanism is further passed on to produce a motor action, but is not present at the level of the inputs (thalamus), so that the mechanism must be supported internally by the cortical network.

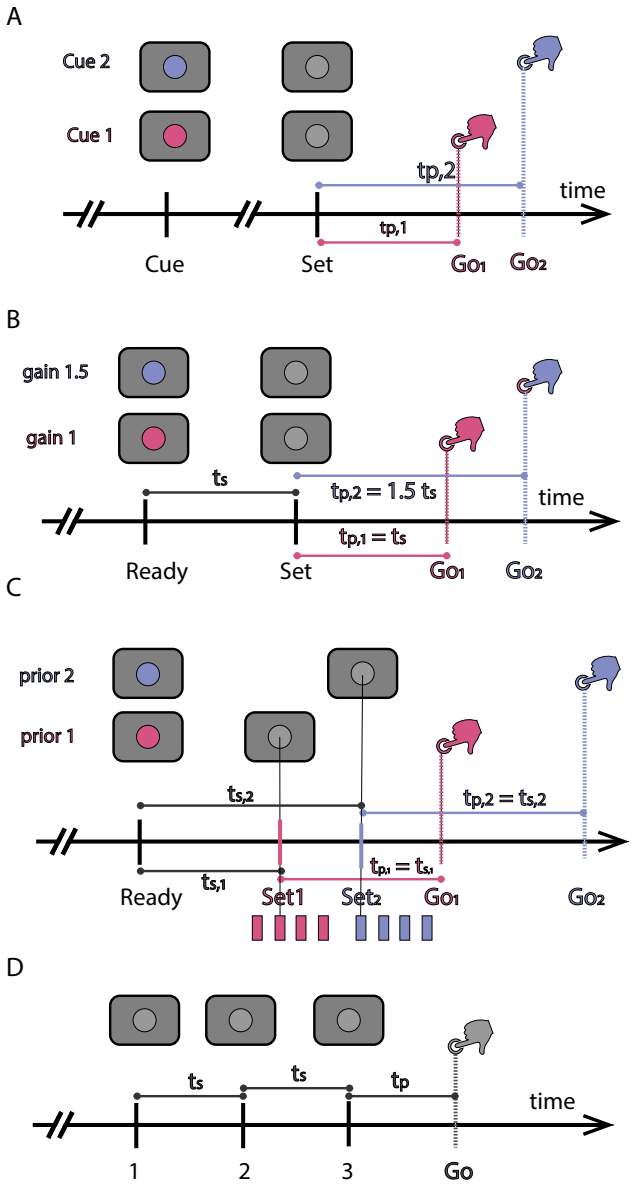

Figure I.4: **Flexible timing tasks**. **A** Cue-Set-Go task. At the beginning of every trial, one of two cues are presented, indicating the animal to produce either a short or a long interval, $t_{p,1}, t_{p,2}$. A second flash, 'Set', appears, corresponding to the beginning of production. The animal must execute a motor action at either time $t_{p,1}$ after 'Set' or $t_{p,2}$ to maximize the reward. This self-initiated action is denominated 'Go'. Adapted from (Wang et al., 2018). **B-C** Ready-Set-Go task. Two flashes, 'Ready' and 'Set' are presented separated by a time t_s. The animal must perform a motor action a time gt_s after the 'Set' stimulus. Parameter g represents the gain. In **B**, the 'Ready-Set-Go' task where two different gains, $g = 1$ and $g = 1.5$ are randomly alternated. The information about the gain is given at the beginning of the trial. Adapted from (Remington et al., 2018a). In **C**, two different prior distributions for the sampled interval t_s are used (both priors are uniformly distributed, with different mean values. Illustrated as bar plots at the bottom). The animal explicitly receives the information about the prior at the beginning of the trial. **D** 1-2-3 Go task. Three different flashes ('1', '2' and '3') are presented at every trial, separated by a time interval t_s. The animal correctly performs the task by producing a motor action at a time $t_p = t_s$ after '3'. Adapted from (Egger et al., 2019).

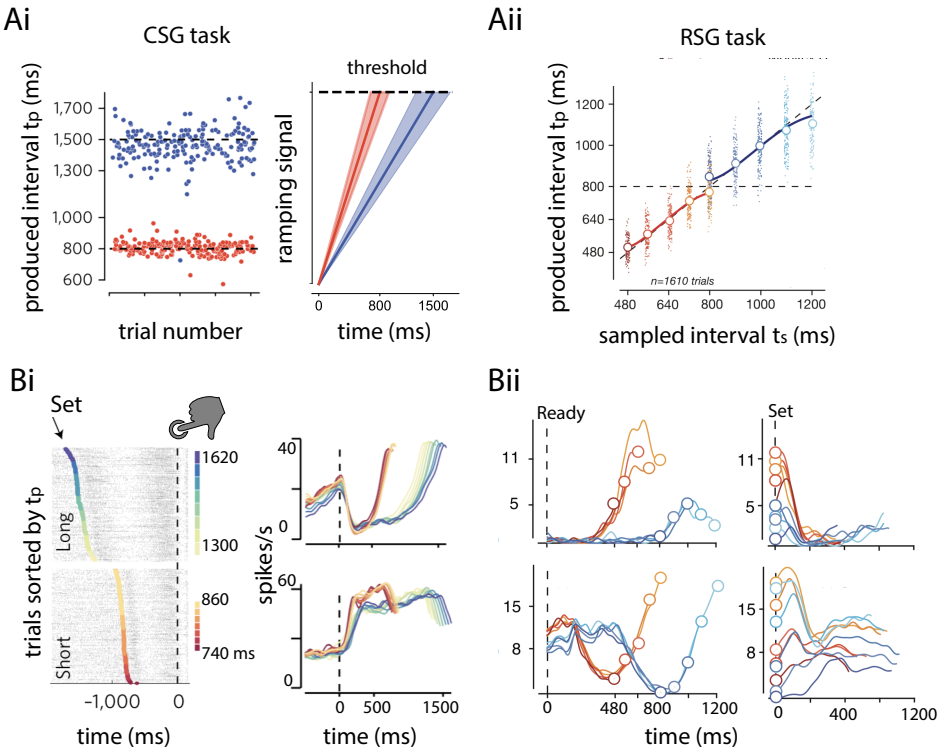

FIGURE I.5: **Behavior and single neuron responses in flexible timing tasks**. **A** Timing performance in the 'Cue-Set-Go' (CSG) task (i, left) using two different cues, corresponding to 800 ms and 1500 ms, adapted from Wang et al. (2018). A ramp-to-threshold model (right), where the speed of the ramp (the slope) is flexibly controlled can describe the observed behavior. (ii) Performance in the 'Ready-Set-Go' (RSG) task alternating two different priors. Adapted from Sohn et al. (2019). Animals show more timing variability for long intervals (target: 1500 ms in CSG) than short intervals (target: 800 ms in CSG). In the RSG, produced intervals are systematically shifted towards the mean of the prior. For instance, if the samples interval is 800 ms, the responses are biased towards longer times for the prior with mean 1000 ms, whereas the produced interval for the same t_s for the red prior is systematically lower (mean of prior: 660 ms). (ii). **B** Single neuron activity in the CSG (i) and RSG task with two different priors (ii). In the CSG task, trials are sorted and clustered by colors based on t_p (left). Two single neuron trial-averaged responses are shown. In the RSG with two different priors, each line corresponds to a different example neuron. Left: Estimation epoch, from Ready to Set. Right: Production epoch, from Set to Go. Responses are non-monotonic in time, showing complex profiles. However, part of the variability from trial to trial can be explained by temporal scaling.

State-space trajectories The temporal scaling of single neuron activity suggests that a build-up signal evolving towards movement initiation might be present at the level of the population response, as a trajectory in neural state-space. The first remarkable feature of neural trajectories is that they are restricted to a relatively low-dimensional subspace in neural space at every epoch of the tasks. Such low-dimensional trajectories can be visualized by projecting them onto two or three directions of neural space. For example, during production, trajectories for different intervals can be projected onto the three orthogonal dimensions that explain most of the variance, i.e., the principal components (Fig. I.6A, left). Trajectories for different produced intervals (red vs blue trajectories) strongly overlap along the first two principal components, whereas they evolve along different states of the third principal component. Further analysis allows to project the activity on the dimensions where the activity shows the highest temporal scaling, the so-called scaling components. On this subspace, that significantly overlaps with the subspace of the principal components, trajectories of different intervals completely overlap (Fig. I.6A, right). This provides evidence that trajectories for different intervals evolve in neural state-space in parallel at different speeds along some subspace where the activity is temporally scaled, while, different intervals evolve along an orthogonal subspace determined by the external cues -the input subspace- that controls the speed.

We illustrate this geometrical arrangement into input and recurrent subspace in Fig. I.6B, where we assume for simplicity that both subspaces are one dimensional. Along the scaling line, trajectories for all time intervals evolve from the same initial state towards a common final state, but they do so at different speeds. Along the non-scaling dimension, the input subspace, the trajectories corresponding to each produced interval remain mostly constant at a different level. This generates parallel trajectories, where the speed in the scaling direction is determined by the activity on the non-scaling subspace. This mechanism, initial conditions on an input subspace of neural space that control the speed of trajectories on an orthogonal subspace, has been found in neural trajectories during production in all tasks (Fig. I.6 C, Ready-Set-Go task).

During the estimation epoch, neural trajectories evolve along a low-dimensional subspace that is mostly orthogonal to the production subspace. At the beginning of the estimation epoch, when the first stimulus arrives, neural trajectories produce transient response along a curved manifold (Fig. I.6 D, from the initial state -square- towards the final states -dots-). The second stimulus is perceived at different stages on this manifold, so that these different states can be mapped onto the different initial conditions required for the flexible production (Remington et al., 2018a; Sohn et al., 2019). In the Ready-Set-Go task with different gains, trajectories with different gains evolve in parallel curved manifolds, implying that the information about the gain is already used during the estimation Remington et al. (2018a). When the Ready-Set-Go task is learned with two different prior distributions of sampled intervals, the different priors analogously elicit different trajectories during measurement, so that the prior information is used from the beginning of the trial.

In the 1-2-3-Go task, where there are two consecutive sampled intervals per trial, neural trajectories during the first estimation display the same properties than in the estimation epoch of the Ready-Set-Go task. During the second estimation, however, neural trajectories show features consistent with both the measurement epoch and the production epoch. While trajectories evolve along transient trajectories from an initial state towards a different final state, there is also a significant temporal scaling subspace along which trajectories evolve. This gives support to the idea that some of the neural computations implemented during the second estimation are predicting the production interval, in order to provide an error signal that updates the first estimate (Egger et al., 2019).

Recurrent network modeling Population responses in neural state-space can be further studied *in silico* by training recurrent neural networks trained to perform flexible timing

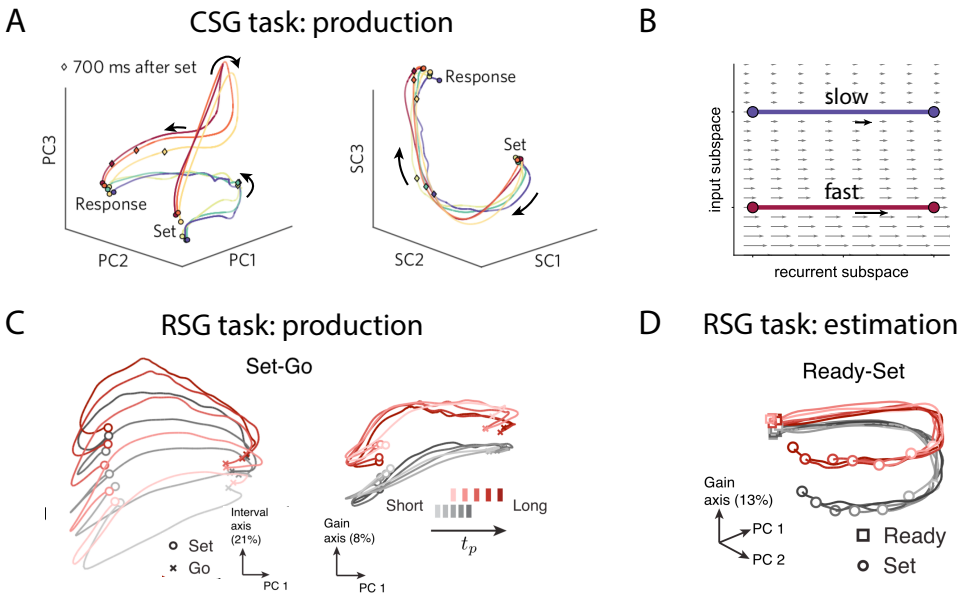

FIGURE I.6: **Population activity in flexible timing tasks**. **A** Cue-Set-Go task, between 'Set' and 'Go'. Colors as in Fig. I.5Ai. Left: Projection of neural trajectories on to the first principal components (orthogonal directions in neural state-space with the highest explained variance). The projection over the two first principal components strongly overlap for both short and long produced interval. The activity on the third principal component do not coincide for short and long trials. Right: Projection of neural trajectories on the three orthogonal directions of neural space with the highest temporal scaling. Trajectories overlap, indicating that there is a temporal scaling subspace. **B** Illustration of two different 2D-trajectories going through a temporal scaling subspace. Trajectories evolve in parallel along the recurrent subspace (the horizontal direction in this picture). However, they go through this subspace at different speeds, as indicated by the flow field. The initial conditions in the orthogonal input subspace (vertical direction) controls the speed of the trajectories. In the neural recordings, the input and temporal scaling subspace are low-dimensional, but span more than one dimension. **C** 2D-projections of neural trajectories in the production epoch of the 'RSG' task, with two different gains (red: $g = 1$, grey: $g = 1.5$.). In some projections (the recurrent subspace, left), the trajectories evolve along the same path, whether in some other dimensions (right: input subspace in the vertical direction), the trajectories of different gains are separated. **D** 2D-projections of neural trajectories in the estimation epoch of the 'RSG' task, with two different gains. The trajectories evolve along a curved manifold. For different gains, the different gain manifolds evolve in parallel. Panel A adapted from Wang et al. (2018), panels C and D from Remington et al. (2018a).

tasks. Given that the full neural state is known at every time point, it is possible to perturb recurrent networks to reject different hypotheses. One key aspect of neural network training concerns the modeling of the inputs and readout output of the network. For these series of tasks, the assumptions are that the inputs network received are simple (either short pulses or tonic inputs) and low dimensional. Therefore, during the time points between inputs, the autonomous dynamics of the network are responsible for generating the suitable temporal patterns that produce a given computation. Regarding the output, based on the ramping-to-threshold model, a linear readout of the firing activity is required to grow towards a fixed threshold. The 'Go' time point, which corresponds to the motor action of the animal in the experiment, is mapped in the recurrent network to the crossing of the threshold.

Networks were successfully trained in all tasks described above. For the Cue-Set-Go task, Wang et al. (2018) found that the trained networks that show similar neural activity to those recorded require a constant external input to account for the 'Cue' information. Different cue values are represented by different amplitudes of this constant external input. When producing a time interval, the neural trajectories start at an initial stable fixed point, and when the 'Set' pulse is received, trajectories evolve towards a different final fixed. These trajectories are low-dimensional and show perfect temporal scaling along a given linear subspace, while the constant 'Cue' input sets the level of the trajectories in the input subspace (Fig. I.7A). The speed of trajectories during the delay, which evolve along the scaling recurrent subspace determines the production interval. Modifying the amplitude of the 'Cue' causally produces correspondingly different temporal intervals. They further studied the linearized dynamics in the vicinity of the final fixed point, and how they are affected by the 'Cue' input. Stronger cue amplitudes corresponded to slower timescales of the linearized dynamics, which at the level of single neurons correlated with neurons closer to their saturating firing rate.

In the Ready-Set-Go tasks, networks trained with constant background input for the context information produced responses similar to the neural recordings (Fig. I.7 B and C, Remington et al. (2018a); Sohn et al. (2019)). The information about gain or the information about the used prior distribution was provided to the network at all times in the amplitude of a constant input. Furthermore, they perturbed the recurrent neural networks right before the end of the sampling epoch, and found that: (i) changing the curvature of the manifold of neural states produced temporal responses with a stronger bias towards the mean of the prior and (ii) shifting all neural states in one direction along the manifold systematically produced either longer or shorter time intervals. This establishes *in silico* a causal link between the neural state before the end of the measurement epoch and the timing production, supporting Bayesian computations in the curvature of neural state-space.

Once the recurrent dynamics in trained networks are identified, the authors built simple neural circuits, with only a few units, using the same mechanisms. For the production of a time interval, they developed a toy model of two mutually inhibiting neurons receiving shared input. This simple model is dynamically bistable, with two basins of attraction delimited by a separatrix in neural space. These dynamics can produce a ramping signal as trajectories evolve from one fixed point towards the other when the 'Set' pulse sends the neural trajectories above the separatrix. Increasing the shared input moves the two stable fixed points closer to each other, and also slows downs the local dynamics around the stable states. Such mechanism can be used to flexibly control the speed of neural trajectories. This two-neuron network can be combined to other modules with similar intrinsic dynamics, forming a control system that can solve more complex tasks, such as the "1-2-3 Go" task, with a performance similar to those obtained in behavioral experiments Egger et al. (2020).

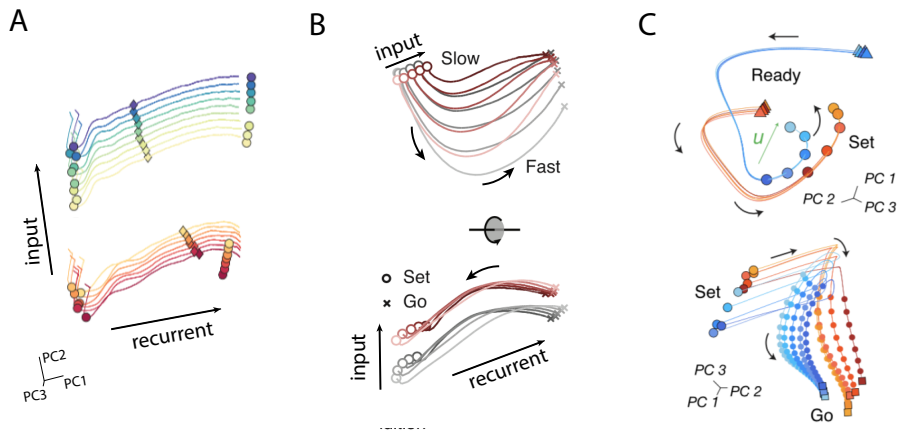

FIGURE I.7: **Recurrent network modeling in flexible timing tasks**. **A** Projection of neural trajectories between 'Set' and 'Go' in the CSG task. The activity of the recurrent networks evolve in parallel along the recurrent subspace, separated by speed along the input subspace. Color lines as in Fig. I.5B. **B** Two different projections of neural trajectories in the RSG task with two different priors. Colors as in Fig. I.6C. The trajectories are arranged into recurrent and input subspaces, similar to the 'CSG' task. **C** Projection of neural trajectories during the estimation epoch (between Ready-Set, top), and the production epoch (between Set-Go, bottom). Colors as in Fig. I.5Aii. During production, the activity for different produced interval evolves in parallel, as in other tasks. In the estimation epoch, the activity evolves along a curved manifold. Estimation manifolds for different priors are parallel to each other. These trained recurrent networks show qualitatively similar trajectories in neural space than those found in the neural recordings. In order to achieve such results, the context information given at the beginning of each trial (cue, gain or prior) must be provided to the network as a tonic input, present during the whole trial duration. Adapted from Wang et al. (2018) (panel A), Remington et al. (2018a) (panel B), and Sohn et al. (2019) (panel C).

I.6 Outline of the work

In this thesis, we investigate how large networks of recurrent units perform flexible temporal computations, by studying the network dynamics, their link to single neuron properties and their connectivity structure. The dissertation is organized into three chapters, that gather the work carried out these last three years, and can be read independently.

A pre-requirement for networks to solve any temporal task in the second and sub-second range is to be able to produce slow timescales, much slower than the membrane time constant of single neurons, which is in the order of a few tens of milliseconds. Theoretical studies have contemplated two different hypotheses for such coordinated slow activity: either (i) cellular and subcellular slow processes, intrinsic to single neurons, that are then transferred to the network dynamics, generating a rich variety of activity timescales (Buonomano and Maass, 2009), or (ii) structure in the network connectivity, that generates the required slow timescales, using the non-linearities of single neurons or by moving the dynamical system in parameter space close to a bifurcation (Huang and Doiron, 2017). For instance, network dynamics that are bistable can virtually generate any slow timescale in the vicinity of the separatrix. Alternatively, the global connectivity structure can push the network dynamics close to a bifurcation, by increasing the synaptic strengths in a network with random all-

to-all connectivity, which slows down the dynamics.

1. In the first chapter, we explore whether slow timescales in recurrent neural networks can be a by-product of slow biophysical processes at the single neuron level, that are often not considered at the network level. In particular, we contrast the effects at the level of network dynamics of two different processes: adaptation and synaptic filtering. Adaptation refers to the slow decrease over time of neural responses to a repeated input, which is a ubiquitous property of excitable biological systems. Synaptic filtering refers to the low-pass filtering of time-varying inputs by the synapses, since neurotransmitters at the synapses regulate at their own speed. We studied the dynamics of a randomly connected recurrent network of rate units displaying either adaptation or synaptic filtering, and applied mean-field theory tools to investigate the dynamical landscapes generated by these networks. We found that the timescale of adaptation has a weak effect at slowing down the network dynamics. In contrast, synaptic filtering does increase the timescale of single neuron activity in the network. This Chapter finally shows that intrinsic slow processes at the cellular level are generally not easily accessible at the network level, challenging the view that they might support the required slow timescales to solve timing tasks.

2. In the second chapter, the starting point is the idea that slow timescales in network activity arise from some structural features of the connectivity that can generate slow dynamics. Based on (Mastrogiuseppe and Ostojic, 2018), we develop a theoretical framework that links how structured network connectivities generate different types of low-dimensional dynamics. We describe large structured networks where the connectivity strengths are drawn randomly from a fixed probability distribution and show that such networks can universally approximate any given dynamical system. Furthermore, we find that there are two key parameters, independent of each other, that constrain the recurrent dynamics: (i) the rank of the connectivity matrix, that introduces the structure in the connectivity, and (ii) the number of neural populations (groups of neurons with different connectivity statistics). The rank of the connectivity matrix sets the dimensionality of the network dynamics. The number of populations determines how flexible the network is to approximate any given low-dimensional dynamics. For low-rank networks with a single statistical population of neurons, in general, only one pair of stable fixed points can be generated. Then, we describe different mechanisms that can be used to generate dynamics with more than two stable fixed points by considering several neural populations. Finally, we propose an algorithm for approximating any dynamical system with a low-rank network, given a large number of populations.

The results of this chapter provide powerful analytical tools that relate network connectivity features to specific implemented network dynamics and indicate that low-rank recurrent neural networks might be useful to study how cortical networks solve cognitive tasks, in particular timing tasks.

3. In the last chapter, we exploited the framework of low-rank networks to study the network mechanisms that implement temporal computations in flexible timing tasks. First, we train recurrent neural networks constrained to have minimal rank to solve the different tasks. We reverse-engineer the trained networks to identify the basic dynamical components that carry out the temporal computations. In a second step, we tested such dynamical components in simplified network models, and used them to implement the different timing tasks. We found that neural trajectories rely on low-dimensional slow manifolds to carry out different temporal computations. Such slow manifolds arise from networks with quasi-isotropic connectivity structure. Small deviations from the isotropic structure shape the dynamics within the manifold, as well as the on=manifold speed can be modulated by tonic external inputs. Overall, we uncovered novel dynamical mechanisms in large recurrent networks that support flexible temporal computations.

Summary of Chapter 1

Neural activity in awake behaving animals exhibits a vast range of timescales that can be several fold larger than the membrane time constant of individual neurons. Two types of mechanisms have been proposed to explain this conundrum. One possibility is that large timescales are generated by a network mechanism based on positive feedback, but this hypothesis requires fine-tuning of the strength or structure of the synaptic connections. A second possibility is that large timescales in the neural dynamics are inherited from large timescales of underlying biophysical processes, two prominent candidates being intrinsic adaptive ionic currents and synaptic transmission. How the timescales of adaptation or synaptic transmission influence the timescale of the network dynamics has however not been fully explored.

To address this question, here we analyze large networks of randomly connected excitatory and inhibitory units with additional degrees of freedom that correspond to adaptation or synaptic filtering. We determine the fixed points of the systems, their stability to perturbations and the corresponding dynamical timescales. Furthermore, we apply dynamical mean field theory to study the temporal statistics of the activity in the fluctuating regime, and examine how the adaptation and synaptic timescales transfer from individual units to the whole population. Our overarching finding is that synaptic filtering and adaptation in single neurons have very different effects at the network level. Unexpectedly, the macroscopic network dynamics do not inherit the large timescale present in adaptive currents. In contrast, the timescales of network activity increase proportionally to the time constant of the synaptic filter. Altogether, our study demonstrates that the timescales of different bio- physical processes have different effects on the network level, so that the slow processes within individual neurons do not necessarily induce slow activity in large recurrent neural networks.

This chapter is based on the article *Contrasting the effects of adaptation and synaptic filtering on the timescales of dynamics in recurrent networks*, by M. Beiran and S. Ostojic (2019), PLoS Comput Biol 15(3): e1006893.

Effects of adaptation and synaptic filtering on the timescales of network dynamics 1

1.1 Introduction

Adaptive behavior requires processing information over a vast span of timescales (Fairhall et al., 2001), ranging from micro-seconds for acoustic localisation (Grothe et al., 2010), milliseconds for detecting changes in the visual field (Tchumatchenko et al., 2011), seconds for evidence integration (Smith and Ratcliff, 2004) and working memory (Miyashita and Chang, 1988), to hours, days or years in the case of long-term memory. Neural activity in the brain is matched to the computational requirements imposed by behavior, and consequently displays dynamics over a similarly vast range of timescales (Bair and Movshon, 2004; Bernacchia et al., 2011; Murray et al., 2014). Since the membrane time constant of an isolated neuron is of the order of tens of milliseconds, the origin of the long timescales observed in the neural activity has been an outstanding puzzle.

Two broad classes of mechanisms have been proposed to account for the existence of long timescales in the neural activity. The first class relies on non-linear collective dynamics that emerge from synaptic interactions between neurons in the local network. Such mechanisms have been proposed to model a variety of phenomena that include working memory (Wang, 2001), decision-making (Wang, 2008) and slow variability in the cortex (Litwin-Kumar and Doiron, 2012). In those models, long timescales emerge close to bifurcations between different types of dynamical states, and therefore typically rely on the fine tuning of some parameter (Huang and Doiron, 2017). An alternative class of mechanisms posits that long timescales are directly inherited from long time constants that exist within individual neurons, at the level of hidden internal states (Buonomano and Maass, 2009). Indeed biophysical processes at the cellular and synaptic level display a rich repertoire of timescales. These include short-term plasticity that functions at the range of hundreds of milliseconds (Zucker and Regehr, 2002; Markram et al., 1998), a variety of synaptic channels with timescales from tens to hundreds of milliseconds (Newberry and Nicoll, 1984; Batchelor et al., 1994; Garthwaite, 1991; Lester et al., 1990), ion channel kinetics implementing adaptive phenomena (Johnston and Wu, 1995), calcium dynamics (Berridge et al., 2003) or shifts in ionic reversal potentials (Gal et al., 2010). How the timescales of these internal processes affect the timescales of activity at the network level has however not been fully explored.

In this study, we focus on adaptative ion-channel currents, which are known to exhibit timescales over several orders of magnitude (La Camera et al., 2006; Benda and Herz, 2003; Ermentrout et al., 2001). We contrast their effects on recurrent network dynamics with

the effect of the temporal filtering of inputs through synaptic currents, which also expands over a large range of timescales (Hennig, 2013). To this end, we extend classical rate models (Wilson and Cowan, 1972, 1973; Sompolinsky et al., 1988; Abbott, 1994) of randomly connected recurrent networks by including for each individual unit a hidden variable that corresponds to either the adapting of the synaptic current. We systematically determine the types of collective activity that emerge in such networks. We then compare the timescales on the level of individual units with the activity within the network.

1.2 Results

We consider N coupled inhibitory and excitatory units whose dynamics are given by two variables: the input current x_i and a slow variable s_i or w_i that accounts for the synaptic filtering or adaptation current respectively. The instantaneous firing rate of each neuron is obtained by applying a static non-linearity $\phi(x)$ to the input current at every point in time. For simplicity, we use a positive and bounded threshold-linear transfer function

$$\phi(x) = \begin{cases} [x-\gamma]^+ & \text{if } x-\gamma < \phi_{\max} \\ \phi_{\max} & \text{otherwise,} \end{cases} \tag{1.1}$$

where $[\cdot]^+$ indicates the positive part, γ is the activation threshold and $\phi_{\max}$ the maximum firing rate.

Single neuron adaptation is described by the variable $w(t)$ that low-pass filters the linearized firing rate with a timescale τ_w, slower than the membrane time constant τ_m, and feeds it back with opposite sign into the input current dynamics (see *Methods*). The dynamics of the i-th adaptive neuron are given by

$$\begin{cases} \tau_m \dot{x}_i(t) = -x_i(t) + \sum_{j=1}^{N} J_{ij}\phi(x_j(t)) - g_w w_i(t) + I_i(t) \\ \tau_w \dot{w}_i(t) = -w_i(t) + x_i(t) - \gamma, \end{cases} \tag{1.2}$$

where $I_i(t)$ is the external input current to neuron i.

Synaptic filtering consists in low-pass filtering the synaptic input received by a cell with time constant τ_s, before it contributes to the input current. The dynamics of the i-th neuron in a network with synaptic filtering are

$$\begin{cases} \tau_m \dot{x}_i(t) = -x_i(t) + s_i(t) \\ \tau_s \dot{s}_i(t) = -s_i(t) + \sum_{j=1}^{N} J_{ij}\phi(x_j(t)) + I_i(t). \end{cases} \tag{1.3}$$

The matrix element J_{ij} corresponds to the synaptic coupling strength from neuron j onto neuron i. In this study we focus on neuronal populations of inhibitory and excitatory units, whose connectivity is sparse, random, with constant in-degree: all neurons receive exactly the same number of excitatory and inhibitory connections, C_E and C_I, as in (Amit and Brunel, 1997; Brunel, 2000; Mastrogiuseppe and Ostojic, 2017). All excitatory synapses have equal strength J and all inhibitory neurons $-gJ$. Furthermore, we consider the large network limit where the number of synaptic neurons N is large while keeping the excitatory and inhibitory inputs C_E and C_I fixed.

1.2.1 Single unit: timescales of dynamics

In the models studied here the input current of individual neurons is described by a linear system. Thus, their activity is fully characterized by the response $h(t)$ to a brief impulse signal, i.e. the linear filter. When such neurons are stimulated with a time-varying input $I(t)$, the response is the convolution of the filter with the input, $x(t) = (h * I)(t)$. These filters can be determined analytically for both neurons with adaptation or synaptic filtering

and directly depend on the parameters of these processes. Analyzing the differences that these two slow processes produce in the linear filters is useful for studying the differences in the response of adaptive and synaptic filtering neurons to temporal stimuli (Fig. 1.1 A), and will serve as a reference for comparison to the effects that emerge at the network level.

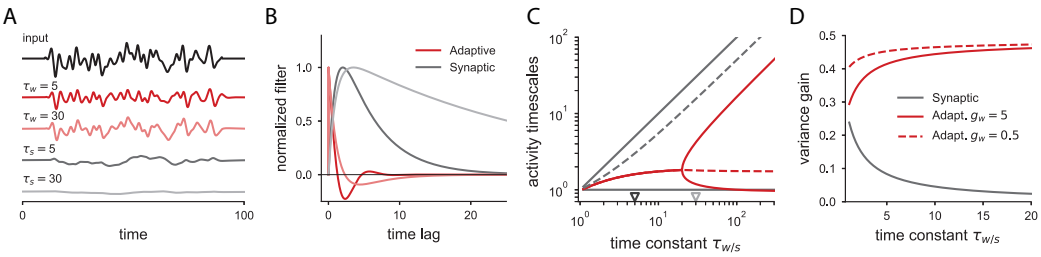

FIGURE 1.1: **Activity of individual neurons with adaptation or synaptic filtering.** A: Firing rate response of two different neurons with adaptation (red curves) and two different neurons with synaptic filtering (grey curves) to the same time-varying input (black curve). B: Normalized linear filters for the neurons shown in A. C: Timescales of the linear filter for neurons with adaptation (red lines) and for neurons with synaptic filtering (grey lines) as a function of the timescale τ_w or τ_s, respectively. The dashed lines indicate the effective timescale of the evoked activity obtained by weighing each individual timescale with its amplitude in the linear filter. The effective timescale for neurons with adaptation saturates for large adaptation time constants, while it grows proportionally to the synaptic time constant for neurons with synaptic filtering. Note that for the adaptive neuron, if the two eigenvalues are complex conjugate, there is only one decay timescale. The triangles on the temporal axis indicate the time constants used in A and B. Adaptation coupling $g_w = 5$. D: Variance of the input current as a function of the slow time constant when the adaptive and synaptic neurons are stimulated with Gaussian white noise of unit variance. In the case of neurons with adaptation, two different values of the adaptation coupling g_w are shown. Time in units of the membrane time constant τ_m.

In particular, the filter of a neuron with synaptic filtering, $h_s(t)$, is the sum of two exponentially decaying filters of opposite signs and equal amplitude, with time constants τ_s and τ_m:

$$h_s(t) = \frac{1}{\tau_s - \tau_m}\left(e^{-\frac{t}{\tau_s}} - e^{-\frac{t}{\tau_m}}\right)\Theta(t), \tag{1.4}$$

where $\Theta(t)$ is the Heaviside function (see *Methods*). Thus, the current response of a neuron to an input pulse received from an excitatory presynaptic neuron is positive and determined by two different timescales. The response first grows with timescale τ_m, so that the neuron cannot respond to any abrupt changes in the synaptic input faster than this timescale, and then decreases back to zero with timescale τ_s (grey curves, Fig. 1.1 B).

The adaptation filter is given as well by the linear combination of two exponential functions. In contrast to the synaptic filter, since the input in the adaptive neuron model affects directly the current variable $x_i(t)$, there is an instantaneous change in the firing rate to an input delta-function (red curves, Fig. 1.1 B). The timescales of the two exponentials can be calculated as

$$\tau^{\pm} = \frac{2\tau_m\tau_w}{\tau_w + \tau_m}\left(1 \pm \sqrt{1 - \frac{4\tau_m\tau_w(1+g_w)}{(\tau_m+\tau_w)^2}}\right)^{-1}. \tag{1.5}$$

When the argument of the square root in Eq. (1.5) is negative, the two timescales correspond to a pair of complex conjugate numbers, so that the filter is an oscillatory function whose amplitude decreases monotonically to zero at a single timescale. If the argument of the square root is positive, for slow enough adaptation, the two timescales are real numbers and correspond to exponential functions of opposing signs of decaying amplitude. However, the amplitudes of these two exponentials are different (see *Methods*). To illustrate this, we focus on the limit of large adaptation time constants with respect to the membrane time constant, where the two exponential functions evolve with timescales that decouple the contribution of the membrane time constant and the adaptation current. In that limit, the adaptive filter reads

$$h_w(t) = \left(-\frac{g_w}{\tau_w} e^{-(1+g_w)\frac{t}{\tau_w}} + \frac{1}{\tau_m} e^{-\frac{t}{\tau_m}} \right) \Theta(t). \tag{1.6}$$

The amplitude of the slow exponential is inversely related to its timescale so that the integral of this mode is fixed, and independent of the adaptation time constant. This implies that a severalfold increase of the adaptation time constant does not lead to strong changes in the single neuron activity for time-varying signals (Fig. 1.1A).

Furthermore, we can characterize the timescale of the single neuron response as the sum of the exponential decay timescales weighed by their relative amplitude, and study how this characteristic timescale evolves as a function of the time constants of either the synaptic or the adaptive current (Fig. 1.1C). For adaptive neurons, the activity timescale is bounded as a consequence of the decreasing amplitude of the slow mode, i.e. increasing the adaptation time constant beyond a certain value will not lead to a slower response. In contrast, the activity of an individual neuron with synaptic filtering scales proportionally to the synaptic filter time, since the relative amplitudes of the two decaying exponentials are independent of the time constants.

When any of the two neuron types are stimulated with white Gaussian noise, the variance in the response is always smaller than the input variance, due to the low pass filtering properties of the neurons. However, this gain in the variance of the input currents is modulated by the different neuron parameters (Fig. 1.1D). For a neuron with synaptic filtering, the gain is inversely proportional to the time constant τ_s. In contrast, for a neuron with adaptation, increasing the adaptation time constant has the opposite effect of increasing the variance of the current response. This is because when the adaptation time constant increases, the amplitude of the slow exponential decreases accordingly, and the low-pass filtering produced by this slow component is weaker. Following the same reasoning, increasing the adaptation coupling corresponds to strengthening the low-pass filtering performed by adaptation, so that the variance decreases (Fig. 1.1D, dashed vs full red curves).

1.2.2 Population-averaged dynamics

In the absence of any external input, a non-trivial equilibrium for the population averaged activity emerges due to the recurrent connectivity of the network. The equilibrium firing rate is identical across network units, since all units are statistically equivalent. We can write the input current x_0 at the fixed point as the solution to the transcendental equation

$$(1 + g_w)\, x_0 = J\,(C_E - gC_I)\,\phi(x_0) + g_w \gamma, \tag{1.7}$$

for the network with adaptation, and to

$$x_0 = J\,(C_E - gC_I)\,\phi(x_0), \tag{1.8}$$

for synaptic filtering (see *Methods*). Based on Eq. (1.7), we find that the adaptation coupling g_w reduces the mean firing rate of the network, independently of whether the network is

dominated by inhibition or excitation (Fig. 1.2A). Synaptic filtering instead does not play any role in determining the equilibrium activity of the neurons, since Eq. (1.8) is independent of the synaptic filtering parameter τ_s.

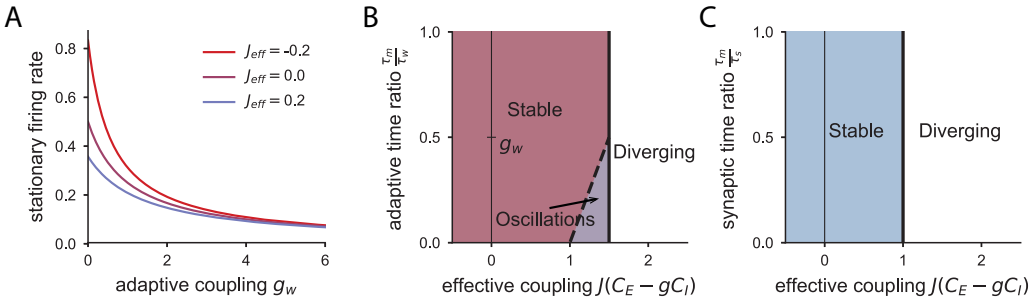

FIGURE 1.2: **Equilibrium firing rate and phase diagrams of the population-averaged dynamics.** A: Firing rate of the network with adaptation at the equilibrium $\phi(x_0)$ for increasing adaptive couplings and three different values of the effective recurrent coupling $J_{\text{eff}} = J(C_E - gC_I)$. Stronger adaptation leads to lower firing rates at equilibrium. B: Phase diagram of the population-averaged activity for the network with adaptation. C: Phase diagram for the network with synaptic filtering.

We next study the stability and dynamics of the equilibrium firing rate in response to a small perturbation uniform across the network, $x_i(t) = x_0 + \delta x(t)$. Because of the fixed in-degree of the connectivity matrix, the linearized dynamics of each neuron are identical, so that the analysis of the homogeneous perturbation on the network reduces to the study of a two-dimensional deterministic system of differential equations which corresponds to the dynamics of the population-averaged response (see *Methods*). The stability and timescales around equilibrium depend on the two eigenvalues of this linear 2D-system. More specifically, the fixed point is stable to a homogeneous perturbation if the two eigenvalues of the dynamic system have negative real part, in which case the inverse of the unsigned real part of the eigenvalues determines the timescales of the response. For both the network with synaptic filtering and the network with adaptive neurons, the order parameter of the connectivity that determines the stability of the fixed point is the effective recurrent coupling $J(C_E - gC_I)$ each neuron receives, resulting from the sum of all input synaptic connections. A positive (negative) effective coupling corresponds to a network where recurrent excitation (inhibition) dominates and the recurrent input provides positive (negative) feedback (Brunel, 2000; Mastrogiuseppe and Ostojic, 2017).

For networks with synaptic filtering, we find that the synaptic time constant does not alter the stability of the equilibrium state, so that the effective coupling alone determines the stability of the population-averaged activity. As the effective input coupling strength is increased, the system undergoes a saddle-node bifurcation when the effective input is $J(C_E - gC_I) = 1$ (Fig. 1.2C). In other words, the strong positive feedback loop generated by the excitatory recurrent connections destabilizes the system.

To analyze the timescales elicited by homogeneous perturbations, we calculate the eigenvalues and eigenvectors of the linearized dynamic system (see *Methods*). We find that for inhibition-dominated networks ($J(C_E - gC_I) < 0$), the network shows population-averaged activity at timescales that interpolate between the membrane time constant and the synaptic time constant. As the effective coupling is increased, the slow timescale at the network level can be made arbitrarily slow by tuning the effective synaptic coupling close to the bifurcation value, a well-known network mechanism to achieve slow neural activity (Huang and Doiron, 2017).

In the limit of very slow synaptic timescale, the two timescales of the population-averaged activity are

$$\tau^{+} = \frac{\tau_s}{1 - J\left(C_E - gC_I\right)}, \tag{1.9}$$

$$\tau^{-} = \tau_m \left(1 - J\left(C_E - gC_I\right)\frac{\tau_s}{\tau_m}\right), \tag{1.10}$$

so that the timescale τ^- is proportional to the membrane time constant and τ^+ is proportional to the slow synaptic time constant, effectively decoupling the two timescales. The relative contribution of these two timescales is the same, independently of the time constant τ_s, as we found in the single neuron analysis.

The network with adaptation shows different effects on the population-averaged activity. First, the presence of adaptation modifies the region of stability: the system is stable when the effective recurrent input $J\left(C_E - gC_I\right)$ is less than the minimum of $1 + g_w$ and $1 + \frac{\tau_m}{\tau_w}$ (see *Methods*). Therefore, the stability region is larger than for the network with synaptic filtering (Fig. 1.2B vs Fig. 1.2C). In other words, the effective excitatory feedback required to destabilize the network is larger due to the counterbalance provided by adaptation. Moreover, adaptation allows the network to undergo two different types of bifurcations as the effective input strength increases, depending on the adaptation parameters. One possibility is a saddle-node bifurcation, as in the synaptic case, which takes place when $J\left(C_E - gC_I\right) = 1 + g_w$. Beyond the instability all neurons in the network saturate. The other possible bifurcation, which happens if $\frac{\tau_m}{\tau_w} < g_w$, at an effective coupling strength $J\left(C_E - gC_I\right) = 1 + \frac{\tau_m}{\tau_w}$, is a Hopf bifurcation: the fixed point of network becomes unstable, leading in general to oscillating dynamics of the population-averaged response. Note that in the limit of very slow adaptation, the system can only undergo a Hopf bifurcation (Fig. 1.2B).

The two timescales of the population-averaged activity in the stable regime for the adaptive network decouple the two single neuron time constants when adaptation is much slower than the membrane time constant. In this limit, up to first order of the adaptive time ratio $\frac{\tau_m}{\tau_w}$, the two activity timescales are

$$\tau^{+} = \frac{\tau_m}{1 - J\left(C_E - gC_I\right)}, \tag{1.11}$$

$$\tau^{-} = \frac{\tau_w\left(1 - J\left(C_E + gC_I\right)\right)}{1 + g_w - J\left(C_E - gC_I\right)}. \tag{1.12}$$

Similar to the single neuron dynamics, the amplitude of the slow mode, corresponding to τ^-, decreases as τ_w is increased, so that the contribution of the slow timescale is effectively reduced when τ_w is very large. On the contrary, the mode corresponding to τ^+, proportional to the membrane time constant can be tuned to reach arbitrarily large values. This network mechanism to obtain slow dynamics does not depend on the adaptation properties.

1.2.3 Heterogeneous activity

1.2.3.1 Linear stability analysis

Previous studies have shown that random connectivity can lead to heterogeneous dynamics where the activity of each unit fluctuates strongly in time (Sompolinsky et al., 1988; Rajan et al., 2010; Kadmon and Sompolinsky, 2015; Mastrogiuseppe and Ostojic, 2017). To assess the effects of additional hidden degrees of freedom on the emergence and timescales of such fluctuating activity, we examine the dynamics when each unit is perturbed independently away from the equilibrium, $x_i\left(t\right) = x_0 + \delta x_i\left(t\right)$. By linearizing the full $2N$-dimensional

dynamics around the fixed point, we can study the stability and timescales of the activity characterized by the set of eigenvalues of the linearized system, λ_s and λ_w for the network with synaptic filtering neurons and adaptation, respectively. These sets of eigenvalues are determined by a direct mapping to the eigenvalues of the connectivity matrix, λ_J (see *Methods*). The eigenvalues λ_J of the connectivity matrices considered are known in the limit of large networks (Rajan and Abbott, 2006; Mastrogiuseppe and Ostojic, 2017): they are enclosed in a circle of radius $J\sqrt{C_E + g^2 C_I}$, except for an outlier that corresponds to the population-averaged dynamics, studied in the previous section. Therefore, we can map the circle that encloses the eigenspectrum λ_J into a different shape in the space of eigenvalues $\lambda_{s/w}$ (insets Fig. 1.3). In order to determine the stability of the response to the perturbation, we assess whether the real part of the eigenspectrum $\lambda_{s/w}$ is negative at all possible points. Furthermore, the type of bifurcation is determined by whether the curve enclosing the eigenvalues $\lambda_{s,w}$ crosses the imaginary axis at zero frequency or at a finite frequency when the synaptic coupling strength is increased, leading respectively to a zero-frequency or to a Hopf bifurcation (Bimbard et al., 2016).

The order parameter of the connectivity that affects the stability and dynamics of the network is now the radius of the circle of eigenvalues λ_J, i.e. $J\sqrt{(C_E + g^2 C_I)}$. This parameter is the standard deviation of the synaptic input weights of a neuron (see *Methods*), which contrasts with the order parameter of the population-averaged response, that depends on the mean of the synaptic input weights. The mean and standard deviation of the synaptic connectivity can be chosen independently, so that while the population-averaged activity remains stable, the individual neurons might not display stable dynamics. To analyze solely the heterogeneous response of the network to the perturbation, we focus in the following on network connectivities whose population-averaged activity is stable, i.e. the effective synaptic coupling is inhibitory or weakly excitatory.

We find that in the network with synaptic filtering, the eigenspectrum λ_s always crosses the stability bound through the real axis, which takes place when the spectral radius of the connectivity is one, $J\sqrt{C_E + g^2 C_I} = 1$. Thus the system undergoes a zero-frequency bifurcation similar to randomly connected networks without hidden variables (Sompolinsky et al., 1988; Kadmon and Sompolinsky, 2015; Schuecker et al., 2018; Mastrogiuseppe and Ostojic, 2017), leading to strong fluctuations at the single neuron level that are self-sustained by the network connectivity (Fig. 1.3 Bii-Biv). The critical coupling at which the equilibrium firing rate loses stability is independent of the synaptic time constant, i.e. synaptic filtering does not affect the stability of heterogeneous responses (Fig. 1.4 A). However, the synaptic time constant τ_s affects the timescales at which the system returns to equilibrium after a perturbation, because the eigenvalues λ_s (see Eq. (1.69) in *Methods*) depend explicitly on τ_s.

For a network with adaptive neurons, we calculate the eigenspectrum λ_w and find that the transition to instability $\mathrm{Re}\,(\lambda_w) = 0$ can happen either at zero frequency or at a finite frequency (see *Methods*), leading to a Hopf bifurcation (as in inset Fig. 1.3 Aiii). In particular, the network dynamics undergo a Hopf bifurcation when

$$\tau_w > \frac{\tau_m}{g_w + \sqrt{2 g_w \left(g_w + 1\right)}}, \tag{1.13}$$

so that strong adaptation coupling and slow adaptation time constants lead to a finite frequency bifurcation. In particular, if the coupling g_w is larger than $\sqrt{5} - 2 \approx 0.236$, only the Hopf bifurcation is possible, since by construction $\frac{\tau_m}{\tau_w} < 1$. We can also calculate the frequency of oscillations at the Hopf bifurcation. We find that, for slow adaptive currents, the Hopf frequency is inversely related to the adaptation time constant (Fig. 1.4B), so that slower adaptation currents produce slower oscillations at the bifurcation.

Adaptation also increases the stability of the equilibrium firing rate to a heterogeneous perturbation, in comparison to a network with synaptic filtering (Fig. 1.4 C). This can be

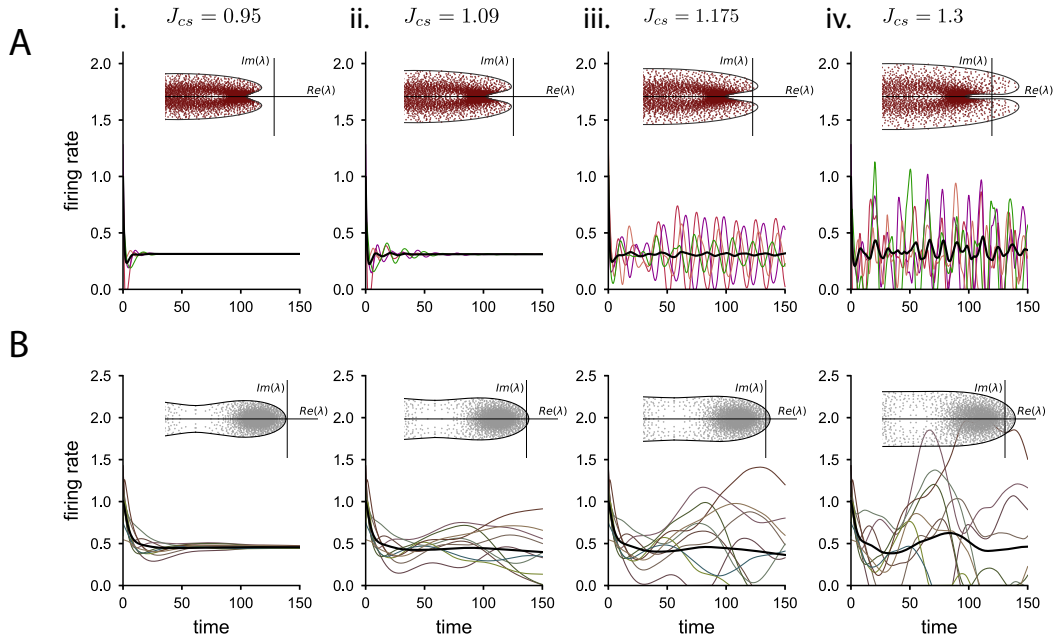

Figure 1.3: **Dynamical regimes as the coupling strength is increased.** Numerical integration of the dynamics for the network with adaptive neurons (row A) and the network with synaptic filtering (row B), as the coupling standard deviation $J_{cs} = J\sqrt{C_E + g^2 C_I}$ is increased. Colored lines correspond to the firing rates of individual neurons, the black line indicates the population average activity. Insets: complex eigenspectrum $\lambda_{w/s}$ of the linearized dynamical matrix around the fixed point. Dots: eigenvalues of the connectivity matrix used in the network simulation. Solid line: theoretical prediction for the envelope of the eigenspectrum. The imaginary axis, $\mathrm{Re}\,(\lambda) = 0$, is the stability boundary. i. Both the network with adaptation and synaptic transmission are stable. ii. The network with synaptic filtering crosses the stability boundary and shows fluctuations in time and across neurons, while the network with adaptation remains stable. iii. The network with synaptic filtering displays stronger fluctuations. The network with adaptive neuron undergoes a Hopf bifurcation leading to strong oscillations at a single frequency with uncorrelated phases across units. Note in the inset that for this connectivity matrix there is only one pair of complex conjugate unstable eigenvalues in the finite network. iv. The network with synaptic filtering shows strong fluctuations. The network with adaptation displays fluctuating activity with an oscillatory component. Parameters: in A, $g_w = 0.5$, and $\tau_w = 5$, in B, $\tau_s = 5$.

intuitively explained in geometrical terms by analyzing how adaptation modifies the shape of the eigenspectrum λ_w with respect to the circular eigenspectrum of the connectivity matrix λ_J.

The Hopf bifurcation leads to the emergence of a new dynamical regime in the network (Fig. 1.3 Aiv), which is studied in the following section. Right at the Hopf bifurcation, the system shows marginal oscillations at a single frequency that can be reproduced in finite-size simulations whenever only one pair of complex conjugate eigenvalues is unstable (Fig. 1.3 Aiii).

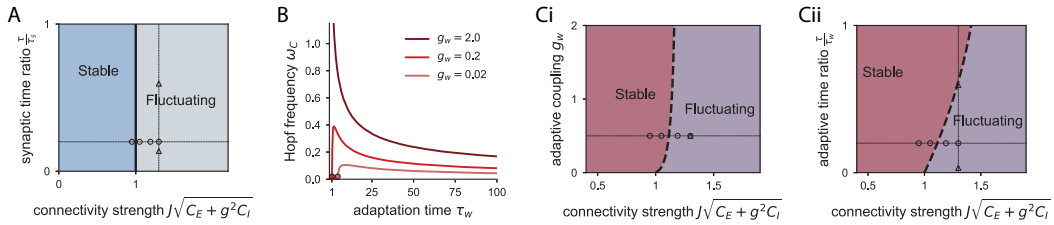

FIGURE 1.4: **Phase diagram and frequency of the bifurcation for the heterogeneous activity.** A: Phase diagram for the network with synaptic transmission. The only relevant parameter to assess the dynamical regime is the connectivity strength. The circles indicate the parameters used in Figs 1.3 and 1.6. Triangles correspond to the parameter combinations used in Fig. 1.5. B: Frequency at which the eigenspectrum loses stability for the network with adaptive neurons as a function of the ratio between membrane and adaptation time constant, τ_m/τ_w, for three different adaptive couplings. The dots indicate the fastest adaptive time constant for which the system undergoes a Hopf bifurcation (Eq. 1.84). C: Phase diagrams for the two adaptation parameters, (i) the coupling g_w and (ii) the adaptive time constant τ_w vs the coupling standard deviation.

1.2.3.2 Fluctuating activity: dynamical mean field theory

The classical tools of linear stability theory applied so far can only describe the dynamics of the system up to the bifurcation. To study the fluctuating regime, we take a different approach and focus on the temporal statistics of the activity, averaged over different connectivity matrices: we determine the mean and autocorrelation function of the single neuron firing rate, and characterize the timescale of the fluctuating dynamics (Sompolinsky et al., 1988; Rajan et al., 2010; Aljadeff et al., 2015a; Harish and Hansel, 2015; Kadmon and Sompolinsky, 2015; Schuecker et al., 2018; Mastrogiuseppe and Ostojic, 2017). For large networks, the dynamics can be statistically described by applying dynamical mean field theory (DMFT), which approximates the deterministic input to each unit by an independent Gaussian noise process. The full network is then reduced to a two-dimensional stochastic differential equation, where the first and second moments of the noise must be calculated self-consistently. We solve the self-consistent equations using a numerical iterative procedure, similar to the schemes followed in (Stern et al., 2014; Lerchner et al., 2006; Dummer et al., 2014; Wieland et al., 2015; Rajan et al., 2010) (see *Methods* for an explanation of the iterative algorithm and its practical limitations).

For the network with synaptic filtering, we find that the autocorrelation function of the firing rates in the fluctuating regime corresponds to a monotonically decreasing function (Fig. 1.5 A), qualitatively similar to the correlation obtained in absence of synaptic filtering (Mastrogiuseppe and Ostojic, 2017). This fluctuating state has often been referred to as rate chaos and shows non-periodical heterogeneous activity which is intrinsically generated by the network connectivity. The main effect of synaptic filtering is on the timescale of these fluctuations. When the synaptic time constant is much larger than the membrane time constant, the timescale of the network activity is proportional to the synaptic time constant τ_s, as indicated by the linear dependence between the half-width of the autocorrelation function and the synaptic timescale τ_s, when all other network parameters are fixed (Fig. 1.5 B).

For the network with adaptation, we focus on large adaptation time constant τ_w, where the network dynamics always undergo a Hopf bifurcation. The autocorrelation function in such a case displays damped oscillations (Fig. 1.5 C). The decay in the envelope of the autocorrelation function is due to the chaotic-like fluctuations of the firing rate activity.

We define the time lag at which the envelope of the autocorrelation function decreases as

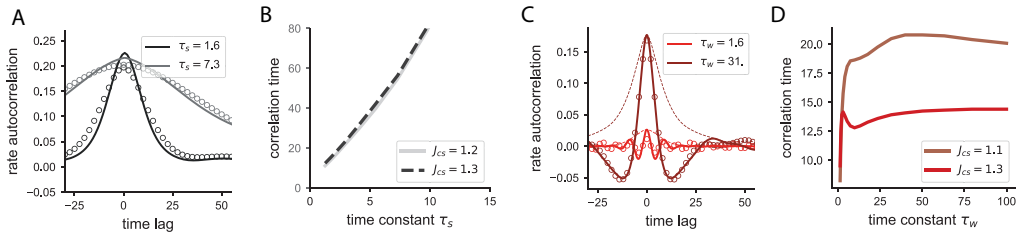

Figure 1.5: **Autocorrelation function and timescale of the network activity in the fluctuating regime.** A: Autocorrelation function of the firing rates in the network with synaptic filtering; dynamical mean field results (solid lines) with their corresponding envelopes (dashed lines), and results from simulations (empty dots). Connectivity strength $J_{cs} = J\sqrt{C_E + g^2 C_I} = 1.2$. B: Effective timescale of the network activity as a function of the synaptic time constant for the network with synaptic filtering. The network coupling does not have a strong effect on the effective timescale. C: Autocorrelation function of the firing rates, as in A, for the system with adaptive neurons. $J_{cs} = 1.3$. D: Effective timescale of the firing rates, as in B, for the system with adaptive currents.

the timescale of the network dynamics (see *Methods*). The timescale of the activity increases as the adaptation timescale is increased, when all the other parameters are fixed (Fig. 1.5 D). However, this activity timescale saturates for large values of the adaptation timescale: the presence of very slow adaptive currents, beyond a certain value, will not slow down strongly the network activity. This saturation value depends on the connectivity strength.

Effects of noise The networks studied so far, for a fixed connectivity matrix, are completely deterministic. We next study the effects of additional white noise inputs to each neuron, as a proxy towards understanding recurrent networks of spiking neurons with adaptation and synaptic filtering. On the mean-field level, such noise is equivalent to studying a recurrent network whose neurons fire action potentials as a Poisson process with instantaneous firing rate $\phi(x_i(t))$ (Ostojic and Brunel, 2011; Kadmon and Sompolinsky, 2015).

Numerical simulations show that in the stable regime the additive external noise generates weak, fast stationary dynamics around the fixed point (Fig. 1.6 Ai, Bi). The timescale of these fluctuations and their amplitude depend on the distance of the eigenspectrum to the stability line, so that the stable fluctuations for weak synaptic coupling standard deviation (Fig. 1.6Ai) are smaller in amplitude than those for larger coupling standard deviation (Fig. 1.6Aii), whose eigenspectrum is closer to the stability boundary. For adaptation, in the fluctuating regime beyond the Hopf bifurcation, the network activity shows again a combination of fluctuating activity and oscillations.

We further extend the DMFT analysis to account for the additional variance of the external white noise sources (see *Methods*). The autocorrelation function of the firing rates, as predicted by DMFT, does not vary drastically when weak noise is added to the network, except for very short time lags, at which white noise introduces fast fluctuations (see Fig. 1.7). For the network with adaptation, the autocorrelation function of the firing rates still shows damped oscillations (Fig. 1.7 A), while for the network with synaptic filtering, similarly, weak noise does not affect much the decay of the autocorrelation function (Fig. 1.7 D). Very strong external noise on the other hand will reduce the effect of the underlying recurrent dynamics of the rate network, since the signal to noise ratio in the synaptic input of all neurons is low.

For a fixed external noise intensity, reducing the adaptation coupling or increasing the adaptation time constant increases the variance of the firing rate (Fig. 1.7B), which resembles the dependence of the variance gain for individual neurons (Fig. 1.1D). Conversely,

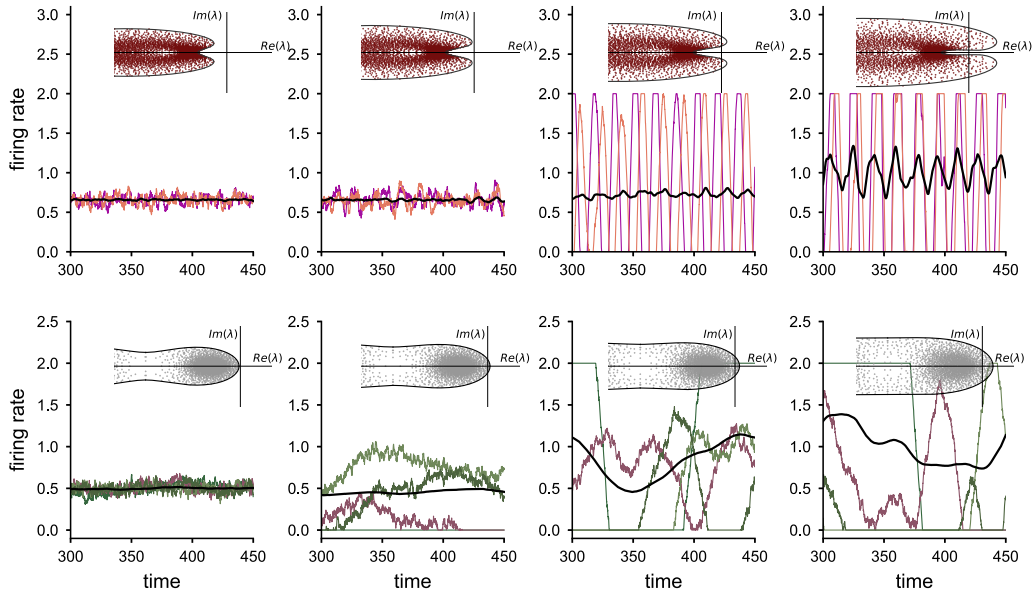

FIGURE 1.6: **Dynamical regimes for the network with adaptation or synaptic filtering with additive external noise.** Numerical integration of the dynamics for the network with adaptive neurons (row A) and the network with synaptic filtering (row B) with units receiving additive external white noise, as a proxy for spiking noise. Colored lines correspond to the firing rate of individual neurons, the black line indicates the population average activity. Insets: complex eigenspectrum $\lambda_{w/s}$ of the dynamic matrix at the fixed point. Dots: eigenvalues of the connectivity matrix used in the network simulation. Solid line: theoretical prediction for the envelope of the eigenspectrum. i. Both the network with adaptation and synaptic transmission are stable, the external noise generates stationary fluctuations around the fixed point. ii. The network with synaptic filtering undergoes a zero-frequency bifurcation. Noise adds fast temporal variability in the firing rates. The network with adaptation remains stable, and the fluctuations are larger in amplitude. iii. The network with adaptation undergoes a Hopf bifurcation. The firing rate activity combines both the fast fluctuations produced by white noise and the chaotic activity with an oscillatory component. iv. The network with adaptation shows highly irregular activity, and strong effects due to the activation and saturation bounds of the transfer function. Parameters as in Fig. 1.4, external noise $\sigma_\eta = 0.06$.

slower synaptic filtering reduces the variance of the neuron's firing rates. This is because in the network with synaptic filtering the noise is also filtered at the synapses –in the limit of very large τ_s, the whole white noise is filtered out– whereas in the network with adaptation the noise affects directly the input current, without being first processed by the adaptation variable.

However, the timescale of the activity is nonetheless drastically affected by strong noise. External noise adds fast fluctuations on top of the intrinsically generated dynamics of the heterogeneous network with adaptation or synaptic filtering. If the noise is too strong, the effective timescale of the activity takes into account mostly this fast component. In that limit, the timescale of the activity is almost independent of the synaptic or adaptive time constants (Fig. 1.7 C and F, largest noise intensity).

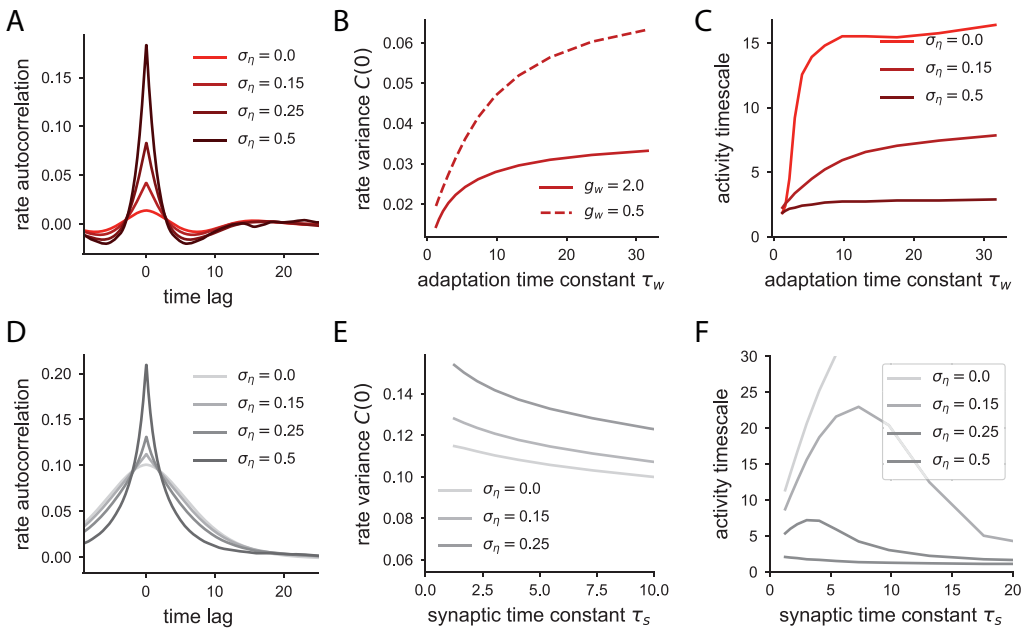

Figure 1.7: **Autocorrelation function, variance of the firing rates and timescale of the network activity with external noise predicted by dynamical mean field theory.** A: Autocorrelation function of the firing rates for the network with adaptive neurons for three different noise intensities. Adaptation time constant $\tau_w = 1.25$. B: Variance of the firing rate as a function of the adaptation time constant for two different adaption couplings g_w. Increasing the adaptation time constant or decreasing the adaptation coupling increases the variance. $\sigma_\eta = 0.15$. C: Timescale of the firing rate as a function of the adaptation time constant, and three different noise levels. Parameters: $g_w = 0.5$, and $J\sqrt{C_E + g^2 C_I} = 1.2$. D: Autocorrelation function of the firing rate for the network with synaptic transmission for three different noise levels. Synaptic time constant $\tau_s = 1.25$. E: Variance of the firing rate as a function of the synaptic time constant, for three different external noise levels. Synaptic filtering reduces the variance. F: Timescale of the activity for the network with synaptic filtering and external noise.

1.3 Discussion

We examined dynamics of excitatory-inhibitory networks in which each unit had a hidden degree of freedom that represented either firing-rate adaptation or synaptic filtering. The core difference between adaptation and synaptic filtering was how external inputs reached the single-unit activation variable that represents the membrane potential. In the case of adaptation, the inputs directly entered the activation variable, which was then filtered by the hidden, adaptive variable through a negative feedback loop. In the case of synaptic filtering, the external inputs instead reached first the hidden, synaptic variable and were therefore low-pass filtered before being propagated in a feed-forward fashion to the activation variable. While both mechanisms introduce a second timescale in addition to the membrane time constant, our main finding is that the interplay between those two timescales is very different in the two situations. Surprisingly, in presence of adaptation, the membrane timescale remains the dominant one in the dynamics, while the contribution of the adaptation timescale appears to be weak. In contrast, in a network with synaptic filtering, the dominant timescale of the dynamics is directly set by the synaptic variable,

and the overall dynamics are essentially equivalent to a network in which the membrane time-constant is replaced with the synaptic one.

We used a highly abstracted model, in which each neuron is represented by membrane current that is directly transformed into a firing-rate through a non-linear transfer function. This class of models has been popular for dissecting dynamics in excitatory-inhibitory (Wilson and Cowan, 1972, 1973; Troyer and Miller, 1997; Murphy and Miller, 2009; Ahmadian et al., 2013) or randomly-connected networks (Sompolinsky et al., 1988; Abbott, 1994; Mastrogiuseppe and Ostojic, 2017), and for implementing computations (Jaeger, 2001; Sussillo and Abbott, 2009). Effects of adaptation in this framework have to our knowledge not been examined so far, but see Muscinelli et al. (2019) for a recent study of adaptation in networks of rate units with random Gaussian connectivity. We therefore extended the standard rate networks by introducing adaptation in an equally abstract fashion (Benda and Herz, 2003), as a hidden variable specified solely by a time constant and a coupling strength. Different values of those parameters can be interpreted as corresponding to different specific membrane conductances that implement adaptation, e.g. the calcium dependent potassium I_{ahp} current or the slow voltage-dependent potassium current I_m, which are known to exhibit timescales over several orders of magnitude (Brown, 2000; Stanley et al., 2011). To cover the large range of adaptation timescales observed in experiments (La Camera et al., 2006), it would be straightforward to superpose several hidden variables with different time constants. Our approach could also be easily extended to include simultaneously adaptation and synaptic filtering.

A number of previous works have studied the effects of adaptation within more biologically constrained, integrate-and-fire models. These works have in particular examined the effects of adaptation on the spiking statistics (Naud et al., 2008; Schwalger et al., 2010; Ladenbauer et al., 2013), firing-rate response (Richardson et al., 2003; Brunel et al., 2003), synchronisation (Ermentrout et al., 2001; Ladenbauer et al., 2012; Augustin et al., 2013; Ladenbauer et al., 2013; Schwalger and Lindner, 2013), perceptual bistability (Laing and Chow, 2002) or single-neuron coding (Naud and Gerstner, 2012; Pozzorini et al., 2013). In contrast, we have focused here on the relation between the timescales of adaptation and those of network dynamics. While our results rely on a simplified firing-rate model, we expect that they can be directly related to networks of spiking neurons by exploiting quantitative techniques for mapping adaptive integrate-and-fire models to effective firing rate descriptions (Augustin et al., 2017).

A side result of our analysis is the finding that strong coupling in random recurrent networks with adaptation generically leads to a novel dynamical state, in which individual units exhibit a mixture of oscillatory and strong temporal fluctuations. The characteristic signature of this dynamical state is a damped oscillation found in the auto-correlation function of single-unit activity. In contrast, classical randomly connected networks lead to a fluctuating, chaotic state in which the auto-correlation function decays monotonically (Sompolinsky et al., 1988; Rajan et al., 2010; Kadmon and Sompolinsky, 2015; Mastrogiuseppe and Ostojic, 2017). Note that the oscillatory activity of different units is totally out of phase, so that no oscillation is seen at the level of population activity. This dynamical phenomenon is analogous to heterogeneous oscillations in anti-symmetrically connected networks with delays (Bimbard et al., 2016). In both cases, the oscillatory dynamics emerge through a bifurcation in which a continuum of eigenvalues crosses the instability line at a finite-frequency. Similar dynamics can be also found in networks in which the connectivity is a superposition of a random and a rank two structured part (Mastrogiuseppe and Ostojic, 2017). In that situation, the heterogeneous oscillations however originate from a Hopf bifurcation due to an isolated pair of eigenvalues that correspond to the structured part of the connectivity.

Our main aim here was to determine how hidden variables could induce long timescales in randomly-connected networks. Long timescales could alternatively emerge from non-random connectivity structure. As extensively investigated in earlier works, one general class of mechanism relies on setting the connectivity parameters close to a bifurcation

that induces arbitrarily long timescales (Sompolinsky et al., 1988; Huang and Doiron, 2017). Another possibility is that non-random features of the connectivity, such as the over-representation of reciprocal connections (Sjöström et al., 2001; Ko et al., 2011) slow down the dynamics away from any bifurcation. A recent study (Martí et al., 2018) has indeed found such a slowing-down. Weak connectivity structure of low-rank type provides yet another mechanism for the emergence of long timescales. Indeed, rank-two networks can generate slow manifolds corresponding to ring attractors provided a weak amount of symmetry is present (Mastrogiuseppe and Ostojic, 2018).

Ultimately, the main reason for looking for long timescales in the dynamics is their potential role in computations performed by recurrent networks (Sussillo, 2014; Barak, 2017). Recent works have proposed that adaptive currents may help implement computations in spiking networks by either introducing slow timescales or reducing the amount of noise due to spiking (Nicola and Clopath, 2017; Bellec et al., 2018). Our results suggest that synaptic filtering is a much more efficient mechanism to this end than adaptation. Identifying a clear computational role for adaptation in recurrent networks therefore remains an open and puzzling question.

1.4 Methods

1.4.1 Network model

We compare the dynamics of two different models: a recurrent network with adaptive neurons, and a recurrent network with synaptic filtering. Each model is defined as a set of $2N$ coupled differential equations. The state of the i-th neuron is determined by two different variables, the input current $x_i(t)$ and the adaptation (synaptic) variable $w_i(t)$ ($s_i(t)$).

Adaptation The dynamics of the recurrent network with adaptive neurons are given by

$$\begin{cases} \tau_m \dot{x}_i(t) = -x_i(t) - g_w w_i(t) + I_i(t) \\ \tau_w \dot{w}_i(t) = -w_i(t) + \phi(x_i(t)), \end{cases} \tag{1.14}$$

where $\phi(x)$ is a monotonically increasing non-linear function that transforms the input current into firing rate. In this study, we use a threshold-linear transfer function with saturation:

$$\phi(x) = \begin{cases} [x-\gamma]^+ & \text{if } x-\gamma < \phi_{\max} \\ \phi_{\max} & \text{otherwise.} \end{cases} \tag{1.15}$$

In Eq. (1.14) adaptation in single neuron rate models is defined as a low-pass filtered version with timescale τ_w of the neuron's firing rate $\phi(x_i(t))$, and is fed back negatively into the input current, with a strength that we call the adaptation coupling g_w. For the sake of mathematical tractability, we linearize the dynamics of the adaptation variable by linearizing the transfer function (Eq. 1.15), $\phi(x_i(t)) \approx x_i(t) - \gamma$. Therefore, the dynamics of the network model with adaptation studied here read

$$\begin{cases} \tau_m \dot{x}_i(t) = -x_i(t) - g_w w_i(t) + I_i(t) \\ \tau_w \dot{w}_i(t) = -w_i(t) + x_i(t) - \gamma, \end{cases} \tag{1.16}$$

Note that this approximation allows for adaptation to increase the input current of a neuron, when the neuron's current is below the activation threshold γ.

Synaptic filtering For the recurrent network with synaptic filtering, the dynamics are

$$\begin{cases} \tau_m \dot{x}_i(t) = -x_i(t) + s_i(t) + I_i(t) \\ \tau_s \dot{s}_i(t) = -s_i(t) + I_i(t). \end{cases} \tag{1.17}$$

In Eqs (1.14), (1.16), and (1.17), $I(t)$ represents the total external input received by the neuron. In general, we are interested in the internally generated dynamical regimes of the network, so that the input is given by the synaptic inputs

$$I_i(t) = I_{\text{syn},i} = \sum_j J_{ij} \phi(x_j(t)). \tag{1.18}$$

The matrix element J_{ij} indicates the coupling strength of the j-th neuron onto the i-th neuron. The connectivity matrix is sparse and random, with constant in-degree (Brunel, 2000; Ostojic, 2014; Mastrogiuseppe and Ostojic, 2017): all neurons receive the same number of input connections C, from which C_E are excitatory and C_I inhibitory. All excitatory synapses have coupling strength J while the strength of all inhibitory synapses is $-gJ$. Moreover, each neuron can only either excite or inhibit the rest of the units in the network,

following Dale's principle. Therefore, the total effective input coupling strength, which is the same for all neurons, is

$$J_{\text{eff}} := \sum_j J_{ij} = J\left(C_E - gC_I\right). \tag{1.19}$$

Table 1.1: **Parameter values used in the simulations.**

Parameter	**Value**
Number of units N	3000
In-degree C	100
Excitatory inputs C_E	80
Inhibitory inputs C_I	20
Ratio I-E coupling strength g	4.1
Threshold γ	-0.5
Maximum firing rate $\phi_{\max}$	2

1.4.2 Single neuron dynamics

The dynamics of each individual neuron are described by a two-dimensional linear system, which implies that the input current response $x(t)$ to a time-dependent input $I(t)$ is the convolution of the input with a linear filter $h(\tau)$ that depends on the parameters of the linear system:

$$x(t) = (h * I)(t) = \int_{-\infty}^{+\infty} dt' h(t') I(t - t'). \tag{1.20}$$

In general, for any linear dynamic system $\dot{z}(t) = Az + b(t)$, where A is a square matrix in $\mathbb{R}^{N\times N}$ and $b(t)$ is a N-dimensional vector, the dynamics are given by

$$z(t) = \int_{-\infty}^{\infty} dt' e^{At'} \Theta(t') b(t - t'), \tag{1.21}$$

where $\Theta(t)$ is the Heaviside function. Thus, comparing Eqs (1.21) and (1.20), the linear filter is determined by the elements of the so-called propagator matrix $P(t) = e^{At}\Theta(t)$.

Synaptic filtering For a single neuron wit synaptic filtering, the dynamics are given by Equation (1.17), where the input $I_i(t)$ represents the external current. We write the response in its vector form $(x(t), s(t))^T$ and the input as $(0, I(t))^T$. The dynamic matrix is

$$A_s = \begin{pmatrix} -\tau_m^{-1} & \tau_m^{-1} \\ 0 & -\tau_s^{-1} \end{pmatrix}. \tag{1.22}$$

The linear filter, $h_s(t')$, is given by the entries of the propagator matrix that links the input $I(t)$ to the output element $x(t)$, which are in this case only the entry in row one and column two: $h_s(t') = [P(t')]_{12}$. To compute the required entry of the propagator, we diagonalize the dynamic matrix $A = VDV^{-1}$. The matrix D is a diagonal matrix with the eigenvalues of matrix A in the diagonal entries, and V is a matrix whose columns are the corresponding eigenvectors. Applying the identity $e^{tVDV^{-1}} = Ve^{tD}V^{-1}$ and the definition of propagator we obtain that

$$h_s(t) = \Theta(t) \frac{1}{\tau_m - \tau_s} \left(e^{-\frac{t}{\tau_m}} - e^{-\frac{t}{\tau_s}}\right). \tag{1.23}$$

The two timescales of the activity are defined by the inverse of the eigenvalues of the system, which coincide with τ_m and τ_s. Every time a pulse is given to the neuron, both modes get activated with equal amplitude and opposing signs, as indicated by Eq. 1.23. This means that there is a fast ascending phase after a pulse, at a temporal scale τ_m, and a decay towards zero with timescale τ_s.

Adaptation The dynamics of a single adaptive neuron are determined by Equation (1.16), where $I_i(t)$ is the external input to the neuron. We apply the same procedure to determine the timescales of the response of an adaptive neuron to time-dependent perturbations. The dynamic matrix for an adaptive neuron reads

$$A_w = \begin{pmatrix} -\tau_m^{-1} & -g_w\tau_m^{-1} \\ \tau_w^{-1} & -\tau_w^{-1} \end{pmatrix}. \tag{1.24}$$

Its eigenvalues are

$$\lambda_w^{\pm} = \frac{1}{2}\left(-\tau_m^{-1} - \tau_w^{-1} \pm \sqrt{\left(\tau_m^{-1} + \tau_w^{-1}\right)^2 - 4\left(1+g_w\right)\tau_m^{-1}\tau_w^{-1}}\right). \tag{1.25}$$

and the eigenvectors

$$\xi^{\pm} = \left(\frac{g_w}{\tau_m}, \frac{1}{2}\left(-\frac{1}{\tau_m} + \frac{1}{\tau_w} \mp \sqrt{\left(\frac{1}{\tau_m} - \frac{1}{\tau_w}\right)^2 - 4\frac{g_w}{\tau_m\tau_w}}\right)\right)^T. \tag{1.26}$$

The eigenvalues are complex if and only if $g_w > \left(4\tau_m\tau_w\right)^{-1}\left(\tau_w - \tau_m\right)^2$, and in that case their real part is $\frac{1}{2\tau_m\tau_w}\left(\tau_m + \tau_w\right)$. As the adaptive time constant becomes slower, at a certain critical adaptation time constant both eigenvalues become real. We are interested in the behavior when the adaptation time constant is large. The absolute value of the inverse of the eigenvalues determines the time constants of the dynamics. Therefore, for large τ_w we can calculate the two real eigenvalues to first order of τ_w^{-1}

$$\lambda_w^{+} = -\tfrac{1+g_w}{\tau_w} + O\left(\tau_w^{-2}\right) \tag{1.27}$$

$$\lambda_w^{-} = -\tau_m^{-1} + g_w\tau_w^{-1} + O\left(\tau_w^{-2}\right). \tag{1.28}$$

In this limit of slow adaptation, the time constant of one eigenmode is proportional to τ_w, whereas the second mode scales with τ_m. We are interested in the amplitude of each mode with respect to the other.
By explicitly calculating the first entry of the propagator matrix we obtain the adaptive filter in terms of the eigenvectors and eigenvalues,

$$h_w(t) = \frac{1}{\xi_1^+\xi_2^- - \xi_1^-\xi_2^+}\left(\xi_1^+\xi_2^- e^{\lambda^+ t} - \xi_1^-\xi_2^+ e^{\lambda^- t}\right), \tag{1.29}$$

where we use the notation ξ_1^+ to indicate the first component of the eigenvector associated to the eigenvalue λ^+. Approximating to leading order of τ_w^{-1} the eigenvectors in Eq. (1.26), we obtain the eigenvectors

$$\xi_- = \frac{1}{\tau_m}\left(g_w, 0\right)^T - \frac{1}{\tau_w}\left(0, g_w\right)^T = g_w\left(\frac{1}{\tau_m}, -\frac{1}{\tau_w}\right)^T \tag{1.30}$$

$$\xi_+ = \frac{1}{\tau_m}\left(g_w, -1\right)^T + \frac{1}{\tau_w}\left(0, 1+g_w\right)^T = \left(\frac{g_w}{\tau_m}, -\frac{1}{\tau_m} + \frac{1+g_w}{\tau_w}\right)^T. \tag{1.31}$$

Then, using Eqs (1.29), (1.30) and (1.31), we determine the linear filter:

$$h_w(t) = \frac{g_w}{\tau_m(2g_w+1)-\tau_w} e^{-\frac{1+g_w}{\tau_w}t} + \frac{1}{\tau_m}\frac{1-(1+g_w)\frac{\tau_m}{\tau_w}}{1-(1+2g_w)\frac{\tau_m}{\tau_w}} e^{-\left(\frac{1}{\tau_m}-\frac{g_w}{\tau_w}\right)t}. \tag{1.32}$$

Interestingly, in contrast with synaptic filtering, the amplitude of the two modes are not equal. The amplitude of the slow mode (first term in Eq. 1.32), whose timescale is proportional to τ_w, decays proportionally to τ_w^{-1} with respect to the fast mode, when $\tau_w \ll \tau_m(2g_w+1)$. Therefore, the area under the linear filter corresponding to this mode is independent of τ_w for very large adaptation time constants:

$$\lim_{\tau_w\to\infty}\int_0^\infty h_w^+(t)\,dt = \lim_{\tau_w\to\infty}\frac{g_w\tau_w}{\tau_m(g_w+1)(2g_w+1)-(g_w+1)\tau_w} = -\frac{g_w}{g_w+1}. \tag{1.33}$$

It follows that, if the adaptation timescale is increased, its relative contribution to the activity will decrease by the same factor, so that very slow adaptive currents will effectively be masked by the fast mode.

1.4.3 Equilibrium activity

The two systems possess a non-trivial equilibrium state at which the input current of all units stays constant. Since all units are statistically equivalent, the equilibrium activity is the same for all units. For synaptic filtering, the input current at equilibrium is given by a transcendental equation, that is obtained by setting to zero the left hand side of Eq. (1.17):

$$x_0 = J(C_E - gC_I)\,\phi(x_0). \tag{1.34}$$

This equilibrium coincides with the fixed point of the system without synaptic filtering.

For adaption, instead, from Eq. (1.16) we obtain that the equilibrium is determined by

$$x_0 = \frac{1}{1+g_w}\left(J(C_E - gC_I)\,\phi(x_0) + g_w\gamma\right). \tag{1.35}$$

We further assume unless otherwise specified that the fixed point of the system is in the linear regime of the transfer function, so that $\phi(x) = x-\gamma$. In that case $x_0 = (J(C_E - gC_I) - g_w)(x_0-\gamma)$, so that larger adaptation coupling corresponds to weaker input currents, i.e. decreasing stationary firing rate. The adaptation time constant does not affect the fixed point.

1.4.4 Dynamics of homogeneous perturbations

We study the neuronal dynamics in response to a small perturbation uniform across the network

$$x_i(t) = x_0 + \delta x(t). \tag{1.36}$$

Synaptic filtering Linearizing Eq. 1.17 we obtain

$$\begin{cases} \tau_m\,\delta\dot{x}_i(t) = -\delta x(t) + \delta s_i(t) \\ \tau_s\delta\dot{s}_i(t) = -\delta s_i(t) + \phi_0'\sum_j J_{ij}\delta x(t), \end{cases} \tag{1.37}$$

where we use the notation $\phi_0' := \left.\frac{d\phi(x)}{dx}\right|_{x_0}$. Because the perturbation δx in Eq. (1.37) is independent of j, using Eq. (1.19) the dynamics for all units are equivalent to the population-averaged dynamics and are given by

$$\begin{cases} \tau_m \, \delta\dot{x}\,(t) = -\delta x\,(t) + \delta s\,(t) \\ \tau_s \delta\dot{s}\,(t) = -\delta s\,(t) + \phi_0' J\,(C_E - gC_I)\,\delta x. \end{cases} \tag{1.38}$$

From Eq. (1.38) we can define the dynamic matrix

$$A_s = \frac{1}{\tau_m}\begin{pmatrix} -1 & 1 \\ \phi_0' J\,(C_E - gC_I)\,\frac{\tau_m}{\tau_s} & -\frac{\tau_m}{\tau_s} \end{pmatrix}. \tag{1.39}$$

The only difference in the linearized dynamics of the population-averaged current with respect to the single neuron dynamics (Eq. 1.22) is the non-diagonal entry $\phi_0' J\,(C_E - gC_I)$. When either the derivative at the fixed point cancels, or when the total effective input is zero, the population dynamics equals the dynamics of a single neuron. The eigenvalues of the population-averaged dynamics are

$$\lambda_s^{\pm} = -\frac{\tau_m + \tau_s}{2\tau_s\tau_m} \pm \sqrt{\left(\frac{\tau_m - \tau_s}{2\tau_s\tau_m}\right)^2 + \frac{J\,(C_E - gC_I)}{\tau_m\tau_s}}. \tag{1.40}$$

and the eigenvectors

$$\xi_s^{\pm} = \left(-1, \frac{\tau_m - \tau_s}{2\tau_s\tau_m} \mp \sqrt{\left(\frac{\tau_m - \tau_s}{2\tau_s\tau_m}\right)^2 + \frac{J\,(C_E - gC_I)}{\tau_m\tau_s}}\right)^T. \tag{1.41}$$

For very large synaptic time constants, the eigenvalues are approximated to leading order as

$$\lambda_s^{+} = \frac{J\,(C_E - gC_I) - 1}{\tau_s} + O\left(\tau_s^{-2}\right) \tag{1.42}$$

$$\lambda_s^{-} = -\frac{1}{\tau_m} - \frac{J\,(C_E - gC_I)}{\tau_s} \tag{1.43}$$

Approximating as well the eigenvectors to leading order, we obtain

$$\xi^{+} = \left(\frac{1}{\tau_m}, \frac{1}{\tau_m} - \frac{1 - J\,(C_E - gC_I)}{\tau_s}\right)^T \tag{1.44}$$

$$\xi^{-} = \left(\frac{1}{\tau_m}, -\frac{J\,(C_E - gC_I)}{\tau_s}\right)^T \tag{1.45}$$

the filter of the linear response to weak homogeneous perturbations reads:

$$h_s\,(t) = \frac{1}{\tau_s}\frac{\xi_1^{-}\xi_1^{+}}{\xi_1^{+}\xi_2^{-} - \xi_1^{-}\xi_2^{+}}\left(e^{\lambda^{-}t} - e^{\lambda^{+}t}\right) \tag{1.46}$$

$$= \frac{1}{\tau_s}\frac{\tau_s - \tau_m\,(1 - J\,(C_E - gC_I))}{\tau_s - \tau_m\,(1 - 2J\,(C_E - gC_I))}\left(e^{\lambda^{-}t} - e^{\lambda^{+}t}\right) \tag{1.47}$$

Note that the amplitude of the two exponential terms is the same, independently of the effective coupling and time constants.

Adaptation For the system with adaptive neurons, the linearized system reads

$$\begin{cases} \tau_m \, \delta \dot{x}_i \left(t\right) = -\delta x_i \left(t\right) - g_w \delta w_i \left(t\right) \left(t\right) + \phi_0' \sum_j J_{ij} \delta x \left(t\right) \\ \tau_w \delta \dot{w}_i \left(t\right) = -\delta w_i \left(t\right) + \delta x \left(t\right). \end{cases} \tag{1.48}$$

As for the network with synaptic filtering, the dynamics of the perturbation are equivalent for each unit, so that we can write down the dynamic matrix for the population-averaged response to homogeneous perturbations

$$A_w = \frac{1}{\tau_m} \begin{pmatrix} -1 + \phi_0' J \left(C_E - g C_I\right) & -g_w \\ \frac{\tau_m}{\tau_w} & -\frac{\tau_m}{\tau_w} \end{pmatrix}. \tag{1.49}$$

The difference with respect to the linear single neuron dynamics (Eq. 1.48) is that the effective recurrent coupling appears now in the first diagonal entry of the dynamic matrix.

When the fixed point is located within the linear range of the transfer function, the derivative is one, so that we do not further specify the factor ϕ_0' in the following equations. Consequently, the dynamics of the system to small perturbations do not depend on the exact value of the fixed point, which does not hold for more general transfer functions.

The eigenvalues of the system read

$$\lambda_w^{\pm} = \left(-\frac{1 - J_{\text{eff}}}{2\tau_m} - \frac{1}{2\tau_w}\right) \left(1 \pm \sqrt{1 + \frac{4\tau_m \left(J_{\text{eff}} - 1 - g_w\right)}{\tau_w \left(J_{\text{eff}} - 1 - \frac{\tau_m}{\tau_w}\right)^2}}\right), \tag{1.50}$$

with eigenvectors

$$\xi_w^{\pm} = \left(2g_w, \frac{\tau_m}{\tau_w} + J_{\text{eff}} - 1 \mp \sqrt{\left(\frac{\tau_m}{\tau_w} - J_{\text{eff}} + 1\right)^2 - 4\frac{\tau_m}{\tau_w}\left(g_w - J_{\text{eff}} + 1\right)}\right)^T \tag{1.51}$$

In the limit of very slow adaptation, given that the two eigenvalues are real, they can be approximated to leading order as

$$\lambda_w^+ = 1 + \frac{\tau_m}{\tau_w \left(J \left(C_E - g C_I\right) - 1\right)} + O\left(\tau_w^{-2}\right) \tag{1.52}$$

$$\lambda_w^- = -\frac{1}{\tau_w} \left(1 - \frac{g_w}{J \left(C_E - g C_I\right) - 1}\right) + O\left(\tau_w^{-2}\right) \tag{1.53}$$

and the corresponding eigenvectors read

$$\xi_w^+ = \left(1, \frac{1}{J_{\text{eff}} - 1} \frac{\tau_m}{\tau_w}\right)^T \tag{1.54}$$

$$\xi_w^- = \left(g_w, J_{\text{eff}} - 1 + \frac{\tau_m}{\tau_w}\left(1 - \frac{g_w}{J_{\text{eff}} - 1}\right)\right)^T. \tag{1.55}$$

Therefore, if the perturbation is stable (see next section) we can write down the corresponding linear filter as

$$h_w \left(t\right) = \frac{1}{\tau_m} \frac{J_{\text{eff}} - 1 + \frac{\tau_m}{\tau_w}\left(1 - \frac{g_w}{J_{\text{eff}}}\right)}{J_{\text{eff}} - 1 + \frac{\tau_m}{\tau_w}\left(1 - \frac{2g_w}{J_{\text{eff}}}\right)} e^{\lambda_w^+ t} - \frac{g_w}{\tau_w \left(J_{\text{eff}} - 1\right)^2 + \tau_m \left(J_{\text{eff}} - 1 - 2g_w\right)} e^{\lambda_w^- t}. \tag{1.56}$$

The area under the slow mode is again independent of the adaptation time constant in this limit,

$$\lim_{\tau_w \to \infty} \int_0^\infty h_w^-(t)\, dt = -\frac{g_w}{(J_{\text{eff}} - 1)(J_{\text{eff}} - 1 - g_w)}. \tag{1.57}$$

1.4.5 Stability of homogeneous perturbations

The equilibrium point is stable when the real part of all eigenvalues is negative. Equivalently, in a two dimensional system –as it is the case for the population-averaged dynamics–, the dynamics are stable when the trace of the dynamic matrix is negative and the determinant positive.

Synaptic filtering In the system with synaptic filtering, the trace and determinant are

$$\text{Tr}_s = -\frac{1}{\tau_m} - \frac{1}{\tau_s} \tag{1.58}$$

$$\text{Det}_s = \frac{1 - J(C_E - gC_I)}{\tau_m \tau_s}. \tag{1.59}$$

The trace is therefore always negative. The determinant is positive, and therefore the population-averaged dynamics are stable, when the effective coupling $J(C_E - gC_I)$ is smaller than unity. In contrast, if the effective coupling is larger than unity, i.e. if positive feedback is too strong, the equilibrium firing rate is unstable, so that any small perturbation to the equilibrium firing rate will lead the system to a different state. Right at the critical effective coupling, one eigenvalues is zero and the other one equals Tr_s, implying that the population-averaged dynamics undergo a saddle-node bifurcation. Beyond the bifurcation, the network reaches a state where the firing rates of all neurons saturate.

Adaptation In the adaptive population dynamics, the recurrent connectivity has a different effect on the stability of the adaptive population dynamics. The trace and determinant of the dynamic matrix are

$$\text{Tr}_w = -\frac{1}{\tau_m} - \frac{1}{\tau_w} + \tau_m^{-1} J(C_E - gC_I), \tag{1.60}$$

$$\text{Det}_w = (\tau_m \tau_w)^{-1} (1 - J(C_E - gC_I) + g_w). \tag{1.61}$$

Both the timescale τ_w and the strength g_w of adaptation affect the trace and determinant of the dynamic matrix, and therefore the stability. The system is unstable if the determinant is negative (one positive and one negative real eigenvalue) or if the determinant is positive and the trace is positive. The determinant is negative, and therefore the system becomes unstable through a saddle-node bifurcation, when $J(C_E - gC_I) > 1 + g_w$. Note that the adaptation strength increases the stability of the system: a stronger positive feedback loop is required to destabilize the fixed point, in comparison to the network with synaptic filtering. The determinant and trace are positive if $J(C_E - gC_I) < 1 + g_w$ but $J(C_E - gC_I) > 1 + \frac{\tau_m}{\tau_w}$, respectively, leading to a Hopf bifurcation: the system produces sustained marginal oscillations at the bifurcation in response to small perturbations around the fixed point. Beyond the Hopf bifurcation, the oscillations are maintained in time, unless the system shows a fixed point when all neurons saturate ($x_0 = \frac{1}{1-g_w}(J(C_E - gC_I)\phi_{max} + g_w\gamma)$). This fixed point exists if $x_0 > \phi_{max} + \gamma$.

1.4.6 Heterogeneous activity

We next study the network dynamics beyond the population-averaged activity, along modes where different units have different amplitudes. We study perturbations of the type

$$x_i(t) = x_0 + \delta x_i(t). \tag{1.62}$$

We define the $2N$-dimensional vector $\mathbf{x} = \left(\delta x_1, ..., \delta x_N^1, \delta w_1^1, ..., \delta w_N^1\right)^T$. Since the dynamics of each unit is now different, the dynamic matrix of the linearized system, A, is described by a squared matrix of dimensionality $2N$. Therefore, the perturbations generate dynamics along $2N$ different modes whose timescales are determined by the eigenvalues of the matrix A. The eigenvalues are determined by the characteristic equation $|A - \lambda I| = 0$. In order to calculate these eigenvalues, we make use of the following identity which holds for any block matrix $Z = A - \lambda I$, that is composed by the four square matrices $\mathbf{P}$,$\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{S}$ and the block $\mathbf{S}$ is invertible:

$$|Z| := \left|\begin{pmatrix} \mathbf{P} & \mathbf{Q} \\ \mathbf{R} & \mathbf{S} \end{pmatrix}\right| = |\mathbf{S}| \left|\mathbf{P} - \mathbf{Q}\mathbf{S}^{-1}\mathbf{R}\right|. \tag{1.63}$$

Consequently, if we set Eq. (1.63) to zero, since we assumed that $|S| \neq 0$, we obtain

$$|Z| = 0 \implies \left|\mathbf{P} - \mathbf{Q}\mathbf{S}^{-1}\mathbf{R}\right| = 0. \tag{1.64}$$

The identity in Eq. (1.63) can be shown by using the decomposition

$$Z = \begin{pmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{S} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{Q} \\ 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{P} - \mathbf{Q}\mathbf{S}^{-1}\mathbf{R} & 0 \\ \mathbf{S}^{-1}\mathbf{R} & \mathbf{I} \end{pmatrix}, \tag{1.65}$$

together with the fact that when a non-diagonal block is zero. The determinant of such a matrix is the product of determinants of the diagonal blocks.

Synaptic filtering The dynamical matrix for the network with synaptic filtering, obtained by linearizing Eqs (1.17), is

$$A_s = \frac{1}{\tau_m} \left(\begin{array}{c|c} -\mathbf{I} & \mathbf{I} \\ \hline \phi_0' \mathbf{J} \frac{\tau_m}{\tau_s} & -\frac{\tau_m}{\tau_s}\mathbf{I} \end{array}\right), \tag{1.66}$$

The matrix $\mathbf{J}$ is the connectivity matrix. Again, we assume in the following that the fixed point is located in the linear range of the transfer function, so that $\phi_0' = 1$.

The characteristic equation, obtained by combining Eqs (1.64) and (1.66), reads

$$\left| -(1 + \tau_m \lambda_s)\mathbf{I} + \left(\frac{\tau_m}{\tau_s} + \tau_m \lambda_s\right)^{-1} \frac{\tau_m}{\tau_s}\mathbf{J} \right| = -(1 + \tau_m \lambda_s) + \frac{\lambda_J}{1 + \tau_s \lambda_s} = 0, \tag{1.67}$$

where λ_J are the eigenvalues of the connectivity matrix. Solving for λ_J we obtain the equation which maps the eigenvalues of the synaptic filtering network dynamics λ_s onto the eigenvalues of the connectivity matrix λ_J,

$$\lambda_J = (1 + \tau_m \lambda_s)(1 + \tau_s \lambda_s). \tag{1.68}$$

In contrast, solving for the eigenvalues of the dynamic matrix λ_s we obtain the inverse mapping

$$\lambda_s^2 + \frac{\tau_s + \tau_m}{\tau_s \tau_m}\lambda_s + \frac{1 - \lambda_J}{\tau_s \tau_m} = 0. \tag{1.69}$$

In other words, Eqs 1.69 and 1.68 constitute two different approaches to assessing the stability of the system (Bimbard et al., 2016). One approach is to examine whether the domain of eigenvalues λ_s resulting from Eq. (1.69) intersect the line $\mathrm{Re}\left(\lambda_s\right) = 0$ (Fig. 1.3, insets in B). The eigenvalues λ_J of the connectivity matrix are distributed within a circle in the complex plane, whose radius is proportional to the synaptic strength, $\lambda_J < J\sqrt{C_E + g^2 C_I}$ plus an outlier real eigenvalue at $J\left(C_E - gC_I\right)$ that corresponds to the homogeneous perturbations studied above (see (Rajan and Abbott, 2006)). We focus in this section on the bulk of eigenvalues that corresponds to modes of activity with different amplitudes for different units. We can therefore parametrize the eigenvalues λ_J as

$$\lambda_J\left(\theta\right) = J\sqrt{C_E + g^2 C_I}e^{i\theta} \tag{1.70}$$

and introduce the parametrization into Eq. (1.69) to obtain an explicit expression for the curve that encloses the eigenspectrum λ_s. Note that in an abuse of notation, we denote the limits of the eigenspectrum as λ and not the eigenvalues themselves that constitute the eigenspectrum.

The alternative approach is to use the inverse mapping from the eigenvalues λ_s to the eigenvalues of the connectivity λ_J, by mapping the line $\mathrm{Re}\left(\lambda_s\right) = 0$ into the space of eigenvalues λ_J (see Fig. 1.4.6). More specifically, the line $\mathrm{Re}\left(\lambda_s\right) = 0$ can be parametrized as

$$\lambda_s = \pm i\omega, \tag{1.71}$$

and introduced into Eq. (1.68). In this case, the stability is assessed by whether the eigenspectrum of the connectivity matrix $\mathbf{J}$ crosses the stability boundary or not (insets in Fig. 1.4.6). This alternative approach is useful for some calculations due to the simple geometry of the connectivity eigenspectrum λ_J.

Taking the alternative approach, introducing Eq. (1.71) into Eq. (1.68), we obtain the stability bound in the complex plane of eigenvalues λ_J:

$$\lambda_J^{sb} = \left(1 + i\tau_m\omega\right)\left(1 + i\tau_s\omega\right). \tag{1.72}$$

The first point of the stability curve $\lambda_J^{sb}\left(\omega\right)$ intersecting with a circle of increasing radius centered at the origin is the closest point of the curve to the origin, i.e. the minimum of $\left|\lambda_J^{sb}\right|^2$ with respect to ω. The squared distance to the origin is

$$\left|\lambda_J^{sb}\right|^2 = \left(1 + \tau_m^2\omega^2\right)\left(1 + \tau_w^2\omega^2\right), \tag{1.73}$$

whose minimum happens trivially at $\omega = 0$, $\lambda_J = 1$ (see Fig. 1.4.6). In conclusion, the system is unstable if

$$J\left(C_E + g^2 C_I\right) > 1. \tag{1.74}$$

Note that this is the same condition as in the case without synaptic filtering, The synaptic filtering system approaches the no-filtering system when $\tau_s \to 0$. Although we are considering in this work synaptic timescales that are larger than the membrane time constant, the analysis is valid for arbitrarily fast synaptic time constants. In that limit, the stability curve in Eq. (1.72) approaches the curve $\lambda_J^{sb} = 1$, retrieving the stability boundary found in (Sompolinsky et al., 1988).

To study the limit of slow synaptic time constant, $\tau_s \gg \tau_m$, we analyze the direct approach, i.e. study how the parameters of adaptation modify the eigenspectrum of the dynamic matrix A_s in the complex plane of eigenvalues λ_s. To this end, we introduce the parametrized connectivity eigenspectrum (Eq. 1.70) into Eq. (1.69), and approximate it to leading order of $\frac{\tau_m}{\tau_s}$. We obtain that the eigenspectrum of eigenvalues λ_s are enclosed by the curves

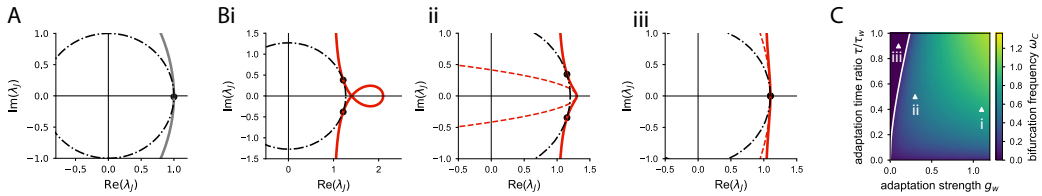

Figure 1.8: **Geometrical description of the bifurcation of the heterogeneous activity.** A: Instability bound for the system with synaptic filtering (grey line, Eq.1.72) and eigenspectrum for the weakest unstable synaptic coupling J. For any parameter combination, the instability bound, a parabola, is first touched by the growing circle of eigenvalues at $\omega = 1$ and value $J\sqrt{C_E + g^2 C_I} = 1$. B: Three different configurations of the instability bound for the system with adaptation in the complex plane of eigenvalues of the connectivity matrix, λ_J. The black dots indicate the intersection between the instability boundary (full red line) and the eigenspectrum of λ_J (dashed black line) with weakest coupling that is unstable. (i) The instability boundary intersects the real axis twice, leading to a Hopf bifurcation. (ii) It intersects the real axis just once and still leads to a Hopf bifurcation, because the intersection with the real axis is not the closest point of the curve to the origin. (iii) It intersects the real axis once and leads to a zero-frequency bifurcation, because the crossing of the real axis is the closest point to the origin. In (ii) and (iii) we draw the parabolic approximation of the instability bound (red dashed line, Eq. 1.82). If the curvature of this parabola is exterior to the λ_J eigenspectrum, as in (iii), the system undergoes a zero-frequency bifurcation. C: Oscillatory frequency at which the network with adaptation undergoes a bifurcation. To the right of the white line (Eq. 1.84), the network displays a Hopf bifurcation, whereas to the left, the bifurcation happens at zero-frequency. The triangles indicate the parameter combinations used in B.

$$\lambda_s^+ \approx \frac{1}{\tau_s}\left(J\sqrt{C_E + g^2 C_I}e^{i\theta} - 1\right) \tag{1.75}$$

$$\lambda_s^- \approx -\frac{1}{\tau_m} - \frac{1}{\tau_s}\left(J\sqrt{C_E + g^2 C_I}e^{i\theta} - 1\right). \tag{1.76}$$

The equations above approximate the full eigenspectrum by two disjoint circles of radius $\tau_s^{-1} J\sqrt{C_E + g^2 C_I}$, the one corresponding to the λ_s^+ eigenvalues centered at $-\frac{1}{\tau_s}$, and the other circle λ^- centered at $-\frac{1}{\tau_m} + \frac{1}{\tau_s}$. The circle centered closer to the instability bound, λ_s^+ sets the slow timescales of the network, and its associated timescale is proportional to τ_s. This gives an intuitive explanation to why the network timescale scales linearly with the synaptic time constant (Fig 1.5).

Adaptation For adaptation, we repeat the same procedure as for the synaptic filtering to determine the stability to heterogeneous perturbations. The dynamical matrix reads

$$A_w = \frac{1}{\tau_m}\left(\begin{array}{c|c} \phi_0'\mathbf{J} - \mathbf{I} & -g_w\mathbf{I} \\ \hline \phi_0'\frac{\tau_m}{\tau_w}\mathbf{I} & -\frac{\tau_m}{\tau_w}\mathbf{I} \end{array}\right), \tag{1.77}$$

Using Eqs (1.64) and (1.77) we can obtain the characteristic equation. Solving for λ_J we obtain the mapping between the λ_w eigenvalues and the connectivity eigenvalues

$$\lambda_J = 1 + \tau_m\lambda_w + g_w\frac{\tau_m}{\tau_w}\left(\tau_m\lambda_w + \frac{\tau_m}{\tau_w}\right)^{-1}, \tag{1.78}$$

while solving for λ_w we obtain the expression for the inverse mapping:

$$(\tau_m \lambda_w)^2 + \left(1 + \frac{\tau_m}{\tau_w} - \lambda_J\right) \tau_m \lambda_w + \frac{\tau_m}{\tau_w} (1 + g_w - \lambda_J) = 0. \tag{1.79}$$

We first explore the inverse mapping. Inserting the parametrization in Eq. (1.71) into Eq. (1.78), the stability curve in the complex plane of connectivity eigenvalues reads

$$\lambda_J^{sb}(\omega) = 1 + \frac{g_w}{1 - \tau_w^2 \omega^2} + i\omega \left(\tau_m - \tau_w \frac{g_w}{1 - \tau_w^2 \omega^2}\right). \tag{1.80}$$

The bifurcation parameters can then be found by determining the closest point of the stability boundary to the origin. The corresponding value of ω determines the oscillatory frequency of the first unstable mode. This value can be zero, corresponding to a zero-frequency bifurcation, which generally leads to slowly fluctuating activity referred to as rate chaos ((Sompolinsky et al., 1988), (Kadmon and Sompolinsky, 2015), (Harish and Hansel, 2015), (Mastrogiuseppe and Ostojic, 2017)). Alternatively, when the parameter ω that minimizes the norm of λ_J^{sb} is non-zero, the system undergoes a Hopf bifurcation.

It is useful to consider the different geometries of the stability curve in Eq. (1.80) in order to identify the closest point of the curve to the origin. Note that the curve shows symmetry with respect to the real axis, $\lambda_J^{sb}(-\omega) = \lambda_J^{sb*}(\omega)$.

The curve might cross the real axis $\mathrm{Im}(\lambda_J) = 0$ either in one or two different values of $|\omega|$. Solving $\mathrm{Re}\left(\lambda_J^{sb}\right) = 0$, we find that the curve crosses twice the real axis, when $\tau_m < \tau_w g_w$ (Fig. 1.8 Bi). In that case, one crossing is the point $\tau_m \lambda_J = 1 + \frac{\tau_m}{\tau_w}$ and the other $\tau_m \lambda_J = 1 + g_w$. This second intersection corresponds to $\omega = 0$. Therefore, it is clear that, since the first crossing of the real axis is closer to the origin than the point at $\omega = 0$, the bifurcation necessarily occurs at non-zero frequency for $\tau_m < \tau_w g_w$.

When the curve crosses only once the zero axis, the point $\lambda_J = 1 + g_w$, corresponding to a zero-frequency, is not necessarily the closest one to the origin (Fig. 1.8 Bii). One approach to determine analytically whether the system undergoes a Hopf or a zero-frequency bifurcation is to look at the curvature at the point $\omega = 0$ and compare it to the curvature of a circle with radius $1 + g_w$. To do so, we approximate both the stability line and the circle by a parabola, and compare their curvatures (dashed curve, Fig1.8 Bii and Biii). First, we write the stability boundary in its implicit form, $\lambda_J^{sb} := x_J^{sb} + i y_J^{sb}$, as

$$\left(y^{sb}\right)^2 - \left(x^{sb} - 1 - g_w\right) \left(x^{sb} - 1 - \frac{\tau_m}{\tau_w}\right)^2 \frac{1}{x^{sb} - 1} = 0. \tag{1.81}$$

Then, we consider small deviations of the coordinates $x^{sb} = 1 + g_w + \epsilon_x$ and $y^{sb} = \epsilon_y$. If we approximate up to first order of ϵ_x and second order of ϵ_y we obtain the parabola

$$\epsilon_y^2 = \frac{\left(g_w - \frac{\tau_m}{\tau_w}\right)^2}{g_w} \epsilon_x + O\left(\epsilon_x^2\right). \tag{1.82}$$

Repeating the same procedure for the circle of eigenvalues, with radius $r = 1 + g_w$ we obtain $\epsilon_y^2 = 2(1 + g_w)\epsilon_x + O\left(\epsilon_x^2\right)$. By requiring the circle of eigenvalues to be interior to the boundary curve (for the same ϵ_x, $\epsilon_{y,\mathrm{circle}}^2 < \epsilon_{y,\mathrm{sb}}^2$), we obtain that the instability parabola is exterior to the circle, therefore the system undergoes a zero-frequency bifurcation (Fig. 1.8C), when

$$\frac{\left(g_w - \frac{\tau_m}{\tau_w}\right)^2}{g_w} \epsilon_x < 2(1 + g_w)\epsilon_x \tag{1.83}$$

which simplifies to

$$\frac{\tau_m}{\tau_w} > g_w + \sqrt{2g_w\left(g_w+1\right)}. \tag{1.84}$$

In the limit of the adaptation timescale approaching the membrane time constant, the left side of the inequality above approaches one. Introducing this value in Eq. 1.84, we find that for adaptive couplings stronger than $g_w > \sqrt{5}-2$ only a Hopf bifurcation is possible.

1.4.7 Dynamical Mean Field Theory

The linearization of the dynamical system from the previous section is only valid up to the instability boundary. A commonly used method to study the dynamics that arise beyond the bifurcation is dynamical mean field theory (DMFT) (Sompolinsky et al., 1988; Rajan et al., 2010; Stern et al., 2014; Harish and Hansel, 2015; Aljadeff et al., 2015a; Kadmon and Sompolinsky, 2015; Mastrogiuseppe and Ostojic, 2017). DMFT approximates the deterministic input to each element of the system by a Gaussian stochastic process, whose first and second moment are determined self-consistently.

The dynamics of the i-th neuron in the synaptic and adaptive network are approximated as

$$\begin{cases} \tau_m\,\dot{x}_i\left(t\right) = -x_i\left(t\right) + s_i\left(t\right) \\ \tau_s\dot{s}_i\left(t\right) = -s_i\left(t\right) + \xi_i\left(t\right), \end{cases} \tag{1.85}$$

$$\begin{cases} \tau_m\,\dot{x}_i\left(t\right) = -x_i\left(t\right) - g_w w_i\left(t\right) + \xi_i\left(t\right) \\ \tau_w\dot{w}_i\left(t\right) = -w_i\left(t\right) + x_i\left(t\right) - \gamma, \end{cases} \tag{1.86}$$

where $\xi_i\left(t\right)$ is a Gaussian variable. In the thermodynamic limit, the noise sources are independent between neurons, so that for $i \neq j$ $\left[\xi_i\left(t\right)\xi_j\left(t'\right)\right] = 0$.

The next step is to determine the self-consistent equations, that links the distribution of ξ_i to the statistics of the original system in Eqs (1.16) and (1.17). First, we relate the statistics of the noise, currents x_i and rates $\phi\left(x_i\right)$ based on the dynamics. Then, we close the equations by explicitly assuring that the transfer function relates the currents and the rates.

To determine the first moment of the noise, we apply that $\xi_i\left(t\right) = \sum_j J_{ij}\phi\left(x_j\left(t\right)\right)$ and average over the population, as in (Mastrogiuseppe and Ostojic, 2017). The first moment of the noise then obeys

$$\left[\xi_i\right] = \left\langle \sum_{j=1}^{N} J_{ij}\phi_j\left(t\right)\right\rangle = J\left(C_E - gC_I\right)\left\langle\phi\right\rangle. \tag{1.87}$$

We calculate next the relation for the second moment of the noise, which again is the same as in (Mastrogiuseppe and Ostojic, 2017):

$$\left[\xi_i\left(t\right)\xi_j\left(t+\tau\right)\right] = \left\langle \sum_{k=1}^{N} J_{ik}\phi_k\left(t\right)\sum_{l=1}^{N} J_{jl}\phi_l\left(t\right)\right\rangle = \delta_{ij}J^2\left(C_E + g^2C_I\right)\left(C\left(\tau\right) - \left\langle\phi\right\rangle^2\right), \tag{1.88}$$

where $C\left(\tau\right) = \left\langle\phi_i\left(t\right)\phi_i\left(t+\tau\right)\right\rangle$.

These equations show that the first and second moment of the Gaussian sources do not depend on the identity of neuron i, so that all neurons are statistically equivalent. Thus, we can reduce the full $2N$-deterministic system to a two-variable stochastic system, describing a prototypical neuron in the network.

The equations (1.87) and (1.88) describe how the noise is related to the properties of the connectivity and the statistics of the rates $\phi(x)$. The next step is to calculate how the first and second moment of the noise are related to the statistics of the input current, which we write as $\mu := [x_i]$ for the first moment and $\Delta(\tau) := [x_i(t)\, x_i(t+\tau)] - \mu^2$ for the second moment.

For the mean of the input current, averaging over units Eqs (1.85) and (1.86) and introducing the result in (1.87) for the synaptic and adaptive system respectively, we obtain

$$\mu_s = [\xi] = J\left(C_E - gC_I\right)\langle\phi\rangle\,, \tag{1.89}$$

$$\mu_w = \frac{1}{1+g_w}\left(g_w\gamma + [\xi]\right) = \frac{1}{1+g_w}\left(g_w\gamma + J\left(C_E - gC_I\right)\langle\phi\rangle\right). \tag{1.90}$$

By differentiating twice $\Delta(\tau)$ with respect to the lag τ and using Eqs (1.85) and (1.88), as in (Sompolinsky et al., 1988; Mastrogiuseppe and Ostojic, 2017) we obtain:

$$\ddot{\Delta}_s(\tau) = \Delta_s(\tau) + \left(Q_s * \Delta_s\right)(\tau) - J^2\left(C_E + g^2 C_I\right)\left(C(\tau) - \langle\phi\rangle^2\right), \tag{1.91}$$

where $Q_s(\tau) := \int_{-\infty}^{+\infty} dt h_s(t)\, h_s(t+\tau)$ is the autocorrelation function of the single neuron filter h_s (Eq.1.23). Equivalently, for the adaptive system, using Eq. (1.86) and (1.88) we obtain

$$\ddot{\Delta}_w(\tau) = \Delta_w(\tau) + \left(g_w\left(g_w Q_w + h_w^{sym} + \dot{h}_w^{sym}\right) * \Delta_w\right)(\tau) - J^2\left(C_E + g^2 C_I\right)\left(C(\tau) - \langle\phi\rangle^2\right). \tag{1.92}$$

where we define in relation to Eq. (1.6) $h_w^{sym}(\tau) = h_w(|\tau|)$, and the autocorrelation function of the adaptive filter $Q_w := \int_{-\infty}^{+\infty} dt h_w(t)\, h_w(t+\tau)$.

Secondly, in order to close the self-consistent description, we can link the statistics of the rates $\phi_i(t)$ with the statistics of the currents $x_i(t)$ by writing the input currents explicitly as Gaussian variables. We can write down the input current at time t and $t+\tau$ explicitly as (see (Rajan et al., 2010)):

$$x(t) = \mu + \sqrt{\Delta(0) - |\Delta(\tau)|}\, z_1 + \operatorname{sgn}(\Delta(\tau))\sqrt{|\Delta(\tau)|}\, z_3 \tag{1.93}$$

$$x(t+\tau) = \mu + \sqrt{\Delta(0) - |\Delta(\tau)|}\, z_2 + \sqrt{|\Delta(\tau)|}\, z_3. \tag{1.94}$$

This explicit construction in terms of Gaussian variables z_1, z_2 and z_3 realizes the constraints $\left[x^2(t)\right] - \mu^2 = \Delta(0)$, $\left[x^2(t+\tau)\right] - \mu^2 = \Delta(0)$ and $\left[x(t)\, x(t+\tau)\right] - \mu^2 = \Delta(\tau)$. Now, explicitly calculating the first moment of the rates by replacing the average for a Gaussian integral and using Eq. (1.93) we obtain

$$\langle\phi\rangle = \int Dz\phi\left(\mu + \sqrt{\Delta(0)}z\right) \tag{1.95}$$

where we use the short-hand notation $\int Dz = \int_{-\infty}^{+\infty} \frac{e^{-\frac{z^2}{2}}}{2} dz$.

For the second moment, introducing Eqs (1.93) and (1.94) into the definition of autocorrelation function of the rate, we get

$$C(\tau) = \int Dz_3 \int Dz_1 \phi\left(\sqrt{\Delta(0) - |\Delta(\tau)|}\, z_1 + \operatorname{sgn}(\Delta(\tau))\sqrt{|\Delta(\tau)|}\, z_3\right) \int Dz_2 \phi\left(\sqrt{\Delta(0) - |\Delta(\tau)|}\, z_2 + \sqrt{|\Delta(\tau)|}\, z_3\right). \tag{1.96}$$

Therefore, in order to determine the self-consistent solution, we need to find a mean and autocorrelation function for the currents that satisfy both Eqs (1.95) and (1.96) and Eqs (1.89) and (1.91) (for the synaptic system) and Eqs (1.90) and (1.92) (for the adaptive system). Once the statistics of the currents and rates are known, it is straight-forward to obtain the statistics of the noise, using Eqs (1.87) and (1.88).

In previous works (Sompolinsky et al., 1988; Kadmon and Sompolinsky, 2015; Harish and Hansel, 2015; Schuecker et al., 2018; Mastrogiuseppe and Ostojic, 2017) it was possible to further simplify the self-consistent equations because the resulting analogous equation to Eqs (1.91) and Eq. (1.92) was a conservative system. However, in the networks studied here, synaptic filtering and adaptation add the convolutional terms in Eqs (1.91) and Eq. (1.92) that make the system non-conservative. Therefore, we followed an alternative approach and found the solutions to the self-consistent equations using an iterative scheme, that circumvents solving directly the integral equations.

Iterative scheme We solve the self-consistent equations numerically following a single-unit iterative scheme, as in (Lerchner et al., 2006; Dummer et al., 2014; Wieland et al., 2015; Stern et al., 2014):

- First, we simulate the dynamics in Eqs (1.85) and (1.86) assuming white Gaussian noise with a certain mean $[\xi]^{(0)}$ and autocorrelation function $[\xi(t)\,\xi(t+\tau)] = \left(\sigma_\xi^{(0)}\right)^2 \delta(\tau)$.
- We calculate the autocorrelation functions of the firing rate and input currents empirically, $\mu^{(0)}$, $\Delta^{(0)}$, $\langle\phi\rangle^{(0)}$ and $C^{(0)}(\tau)$.
- We simulate in the new iteration $k+1$ the noise following the self-consistent statistics obtained in the previous iteration, as indicated by Eqs (1.87) and (1.88)

$$[\xi]^{(k+1)} = J\left(C_E - gC_I\right)\langle\phi\rangle^{(k)} \tag{1.97}$$

$$[\xi(t)\,\xi(t+\tau)]^{k+1} = J^2\left(C_E + g^2 C_I\right)\left(C^{(k)}(\tau) - \langle\phi\rangle^{(k)}\right). \tag{1.98}$$

  In order to numerically generate a Gaussian variable with autocorrelation function $G(\tau)$, we first generate the noise in the Fourier domain, where each frequency component of the noise is given by

$$\tilde{\xi}(\omega) = \sqrt{\tilde{G}(\omega)}e^{i\psi}, \tag{1.99}$$

  where $\tilde{G}(\omega)$ denotes the Fourier transform of the target autocorrelation function, and ψ is a random variable with uniform probability density in the range $[-\pi, \pi]$.
- We repeat the previous step until the values $\mu^{(k)}$, $\Delta^{(k)}$, $\langle\phi\rangle^{(k)}$ and $C^{(k)}(\tau)$ do not vary beyond a certain tolerance for new iterations.

We find that such an iterative method applied to the systems studied here converges to a solution for the parameters of the noise after a few iterations, independently of the noise properties used in the initial step.

The drawbacks of this iterative scheme are that the two-dimensional system needs to be simulated several times at each iteration in order to determine the first and second order statistics of the input current and the firing rate, which is in general a computationally costly operation. We also find that the method converges more robustly to the solution (given the fact that both the trial length in the simulation and the number of trials are

finite), at the expense of initial speed convergence, when the first and second moments of the noise are only partially updated at each iteration, so that

$$[\xi]^{(k+1)} = (1-\alpha)[\xi]^{(k)} + \alpha J (C_E - gC_I) \langle\phi\rangle^{(k)}, \tag{1.100}$$

and similarly for the second-moment equation, where α is a parameter between zero and one. In this work, we used $\alpha = 0.6$.

This method is inefficient for very large adaptation and synaptic time constants, since it requires simulating with both a fine temporal resolution (faster than the membrane time constant) over very large intervals (much larger than the slow adaptive/synaptic timescale). Another drawback of the iterative method is that its convergence is based on the assumption that smooth changes in the noise statistics lead to smooth changes in the statistics of the firing rates. In general, close to a bifurcation, this requirement may not hold.

Dynamics with intrinsic noise We next study how white Gaussian noise, independent between neurons and intrinsic to each unit in the network, affects the dynamics of the system. On the mean-field level, this is equivalent to studying a network where each neuron spikes at a Poisson process whose rate varies in time as $\phi(x_i(t))$ (Kadmon and Sompolinsky, 2015). The additional input to each neuron, whose dynamics are given in Eqs (1.2) and (1.3), is now

$$I_i^{ext}(t) = \eta_i(t), \tag{1.101}$$

where $[\eta_i] = 0$, and $[\eta_i(t)\eta_j(t+\tau)] = \delta_{ij}\frac{\sigma_\eta^2}{2}\delta(\tau)$, and Gaussian distributed. The DMF equations are derived following the same steps as in the absence of intrinsic noise. The stochastic variable $\xi(t)$ is the sum of the recurrent input and the intrinsic noise. Its first moment remains unchanged:

$$[\xi(t)] = \left\langle \sum_{j=1}^{N} J_{ij}\phi(x_j(t)) + \eta_i(t) \right\rangle \tag{1.102}$$

$$= J(C_E - gC_I)\langle\phi\rangle, \tag{1.103}$$

which is the same result as Eq. (1.87). The second moment of the stochastic process is the sum of the variance generated by the recurrent connections and the variance of the intrinsic noise

$$[\xi(t)\xi(t+\tau)] = \left\langle \sum_{k=1}^{N} J_{ik}\phi_k(t) \sum_{l=1}^{N} J_{il}\phi_l(t) + \eta_i(t)\eta_i(t+\tau) \right\rangle \tag{1.104}$$

$$= J^2\left(C_E + g^2 C_I\right)\left(C(\tau) - \langle\phi\rangle^2\right) + \frac{1}{2}\sigma_\eta^2\delta(\tau). \tag{1.105}$$

Accordingly, the iterative scheme now takes into account the equation above, so that the equation for the second moment of the self-consistent relation (Eq. 1.98) reads when there is intrinsic noise

$$[\xi(t)\xi(t+\tau)]^{(k+1)} = J^2\left(C_E + g^2 C_I\right)\left(C^{(k)}(\tau) - \langle\phi\rangle^{(k)}\right) + \frac{1}{2}\sigma_\eta^2\delta(\tau). \tag{1.106}$$

Adding white noise produces a discontinuity in the derivative of the autocorrelation function of the firing rates at zero lag (Fig. 1.7 A and D). This can be shown by integrating

explicitly both sides of the Eqs (1.91) and (1.92) around zero when the external noise is added. It results in the condition

$$\dot{\Delta}\left(0^{+}\right)-\dot{\Delta}\left(0^{-}\right)=\frac{1}{2}\sigma_{\eta}^{2}. \tag{1.107}$$

Since the autocorrelation function is a symmetric function, $\dot{\Delta}\left(0^{+}\right)=-\dot{\Delta}\left(0^{+}\right)$, leading to

$$\dot{\Delta}_{0}=\sigma_{\eta}^{2}. \tag{1.108}$$

Thus, the autocorrelation function of the input current decays linearly at zero time lag with a slope proportional to the external noise intensity, which also extends to the autocorrelation function of the firing rate.

1.4.8 Definition of the timescale of the activity

The activity of multivariable dynamical systems ranges over several timescales. In particular, for stable linear systems, the timescales of the activity are given by the inverse of the absolute values of the real part of the eigenvalues. As we showed before, for single adaptive or synaptic neurons, the activity consists of two modes that evolve at two different timescales. However, the relative contribution of each of the excited modes can make one timescale more predominant than the other, as it happens for slow adaptation time constant, which becomes effectively undetectable in the single neuron dynamics.

In this work, we calculate the timescale of the activity for linear systems as the average of the timescales of the activated input current modes, weighed by their contribution (Fig. 1.1). For a linear system with filter $h\left(t\right)=\sum_{k}a_{k}e^{-\frac{t}{\tau_{k}}}$, the correlation time is

$$\tau_{corr}=\frac{\sum_{k}\left|a_{k}\right|\tau_{k}}{\sum_{k}\left|a_{k}\right|}. \tag{1.109}$$

For large networks, which are high-dimensional non-linear systems, we define the main timescale of the activity as the time lag at which the autocorrelation function has decayed to a fraction $e^{-\frac{1}{2}}$ of its maximum (Figs 1.5 and 1.7):

$$\tau_{corr}=2\cdot\underset{\tau}{\operatorname{argmin}}\left|E\left[C\left(\tau\right)\right]-\frac{E\left[C\left(\tau\right)\right]}{\sqrt{e}}\right|, \tag{1.110}$$

where $E[C\left(\tau\right)]$ is the envelope of the autocorrelation function, calculated as the norm of its analytic signal, computed using the Hilbert transform. This corresponds to the width of the envelope at which the autocorrelation decays to $e^{-0.5}$ of its value. For an exponentially decaying correlation function, this measure corresponds to the decay time constant. For a Gaussian envelope, this measure would correspond to two times its standard deviation, 2σ.

Summary of Chapter 2

An emerging paradigm proposes that neural computations can be understood at the level of dynamical systems that govern low-dimensional trajectories of collective neural activity. How the connectivity structure of a network determines the emergent dynamical system however remains to be clarified.

Here we consider a novel class of models, Gaussian-mixture low-rank recurrent networks, in which the rank of the connectivity matrix and the number of statistically-defined populations are independent hyper-parameters. We show that the resulting collective dynamics form a dynamical system, where the rank sets the dimensionality and the population structure shapes the dynamics. In particular, the collective dynamics can be described in terms of a simplified effective circuit of interacting latent variables. While having a single, global population strongly restricts the possible dynamics, we demonstrate that if the number of populations is large enough, a rank R network can approximate any R-dimensional dynamical system.

This chapter is based on the manuscript *Shaping dynamics with multiple populations in low-rank recurrent networks*, by M. Beiran, A. Dubreuil, A. Valente, F. Mastrogiuseppe and S. Ostojic, submitted.

Shaping dynamics with multiple populations in low-rank recurrent networks 2

2.1 Introduction

A newly emerging paradigm posits that neural computations rely on collective dynamics in the state-space corresponding to the joint activity of all neurons in a network (Churchland and Shenoy, 2007; Rabinovich et al., 2008; Buonomano and Maass, 2009; Saxena and Cunningham, 2019; Vyas et al., 2020). Experiments in behaving animals have found that trajectories of neural activity are typically restricted to low-dimensional manifolds in that space (Machens et al., 2010; Mante et al., 2013; Rigotti et al., 2013; Gao et al., 2015; Gallego et al., 2018; Chaisangmongkon et al., 2017; Wang et al., 2018; Sohn et al., 2019), and can therefore be described by a small number of collective, latent variables. It has been proposed that these collective variables form dynamical systems that implement computations through their responses to inputs (Paulin, 2004; Hennequin et al., 2014; Rajan et al., 2016; Remington et al., 2018a,b). How synaptic connectivity shapes the effective dynamics of collective variables, and therefore computations, however remains to be clarified.

Recurrent neural networks (RNNs) trained to perform neuroscience tasks are an ideal model system to address this question and further develop the theory of computations through dynamics (Sussillo et al., 2015; Rajan et al., 2016; Barak, 2017; Wang et al., 2018; Yang et al., 2019). A recently introduced class of models, low-rank RNNs, directly embodies the idea of low-dimensional collective dynamics, opens the door to relating connectivity and dynamics, and provides a framework that unifies a number of specific RNN classes (Mastrogiuseppe and Ostojic, 2018). Low-rank RNNs rely on connectivity matrices that are restricted to be low rank, which directly generate low-dimensional dynamics. The rank of the network determines the number of collective variables needed to provide a full description of the collective dynamics. While previous works have shown that other specific classes of RNNs can approximate arbitrary dynamical systems (Doya, 1993; Maass et al., 2007), the range of collective dynamics that can be implemented by low-rank RNNs however remains to be clarified.

In this work, we focus on low-rank RNNs in which neurons are organized in distinct populations that correspond to clusters in the space of low-rank connectivity patterns. Each population is defined by its statistics of connectivity, described by a multi-variate Gaussian distribution, so that the full network is specified by a mixture of Gaussians. The total number of populations in the network is a hyper-parameter distinct from the rank of connectivity. Previous works have considered low-rank networks consisting of a single, global Gaussian population (Mastrogiuseppe and Ostojic, 2018, 2019; Schuessler et al., 2020a). In

the opposite limit, by increasing the number of populations, a Gaussian mixture model can approximate any arbitrary low-rank connectivity distribution. Here we examine how the number of populations and their structure determine and limit the resulting collective dynamics in the network.

We first derive three general properties of Gaussian-mixture low-rank networks: (i) in an autonomous network of rank R, dynamics are characterized by R collective variables that form a dynamical system; (ii) the dynamics are determined by an effective circuit description, where collective variables interact through gain-modulated effective couplings; (iii) the resulting low-dimensional dynamics can approximate any arbitrary R-dimensional dynamical system if the number of populations is large enough. We then proceed to illustrate how increasing the number of populations in a network extends its dynamical range. For that, we specifically focus on fixed points of the dynamics. While a network consisting of a single population can generate at most a pair of stable fixed points, independently of its rank, we show that adding populations allow the network to implement arbitrary numbers of stable fixed points embedded in a subspace determined by the rank of the connectivity matrix. Finally, we propose a general algorithm to approximate a given R-dimensional dynamical system with a multi-population network of rank R, and show one example network that is designed to implement complex temporal dynamics.

2.2 Model class: Gaussian mixture low-rank networks

In this section, we introduce the class of models we study, and define the key underlying quantities.

We consider a recurrent neural network of N rate units. The dynamics of the input x_i to the i-th unit are given by

$$\tau \frac{dx_i}{dt} = -x_i + \sum_{j=1}^{N} J_{ij} \phi\left(x_i\right) + I_i^{ext}\left(t\right) \tag{2.1}$$

where τ corresponds to the membrane time constant, the matrix element J_{ij} is the synaptic strength from unit j to unit i and $I_i^{ext}\left(t\right)$ is the external input received by the i-th unit. The non-linear function $\phi\left(x\right)$ maps the input of a neuron to its firing rate activity. Throughout this study, we use the non-linear activation function $\phi\left(x\right) = \tanh\left(x\right)$, although the theoretical results in Section 2.3 hold for any non-polynomial activation function.

We restrict the connectivity matrix to be of low rank, i.e. the number of non-zero singular values of the matrix J is $R \ll N$. Using singular value decomposition, any connectivity matrix of this type can be expressed as the sum of R unit rank terms,

$$J_{ij} = \frac{1}{N} \sum_{r=1}^{R} m_i^{(r)} n_j^{(r)}. \tag{2.2}$$

The connectivity is therefore characterized by a set of R N-dimensional vectors, or connectivity patterns, $\mathbf{m^{(r)}} = \left\{m_i^{(r)}\right\}_{i=1\ldots N}$ and $\mathbf{n^{(r)}} = \left\{n_i^{(r)}\right\}_{i=1\ldots N}$ for $r = 1, \ldots, R$ where $\mathbf{m^{(r)}}$ are the left singular vectors of the connectivity matrix, and $\mathbf{n^{(r)}}$ correspond to the right singular vectors multiplied by the corresponding singular values (see Fig. 2.1A for an example of a rank-two connectivity matrix). The vectors $\mathbf{m^{(r)}}$ (resp. $\mathbf{n^{(r)}}$) for $r = 1, \ldots, R$ are mutually orthogonal. Without loss of generality, we fix the norm of the left singular vectors $\mathbf{m^{(r)}}$ to be equal to N. This decomposition is unique, up to a change in sign of the set of vectors $\mathbf{m^{(r)}}$ and $\mathbf{n^{(r)}}$.

The external input can be expressed as the sum of N_{in} time-varying terms

$$I_i^{ext}\left(t\right) = \sum_{s=1}^{N_{in}} I_i^{(s)} u_s\left(t\right), \tag{2.3}$$

which are fed into the network through a set of orthonormal input patterns $\mathbf{I^{(s)}} = \left\{I_i^{(s)}\right\}_{i=1\ldots N}$ for $s = 1, \ldots, N_{in}$. In this study, we focus on the dynamics of autonomous networks or networks with a constant external input.

Each neuron in the network is therefore characterized by its $2R + N_{in}$ components on the connectivity patterns $\mathbf{m^{(r)}}$ and $\mathbf{n^{(r)}}$ and input patterns $\mathbf{I^{(s)}}$. By analogy with factor analysis, we refer to these components as pattern loadings, and denote the set of loadings for neuron i as

$$\left(\left\{m_i^{(r)}\right\}_{r=1\ldots R}, \left\{n_i^{(r)}\right\}_{r=1\ldots R}, \left\{I_i^{(s)}\right\}_{s=1\ldots N_{in}}\right) := \left(\underline{m}_i, \underline{n}_i, \underline{I}_i\right). \tag{2.4}$$

Each neuron can thus be represented as a point in the loading space of dimension $2R + N_{in}$, and the connectivity of the full network can therefore be described as a set of N points in this pattern loading space (see Fig. 2.1B).

We assume that for each neuron, the set of pattern loadings is generated independently from a multi-variate probability distribution $P\left(\underline{m}, \underline{n}, \underline{I}\right)$. We moreover restrict ourselves to a specific class of loading distributions, mixtures of multi-variate Gaussians. This choice

is motivated by the fact that Gaussian mixtures can approximate any arbitrary multivariate distribution, afford a natural interpretation in terms of populations, and allow for a mathematically tractable and transparent analysis of the dynamics as shown below.

In this Gaussian mixture model, each neuron is assigned to a population p with probability α_p, $p = 1 \dots P$, so that the connectivity matrix J is a block matrix (Aljadeff et al., 2015a,b). Within population p, the joint distribution $P^{(p)}(\underline{m}, \underline{n}, \underline{I})$ is a multivariate Gaussian defined by (i) its mean $\boldsymbol{a}^{(\mathbf{p})}$, a vector of dimension $2R + N_{in}$, given by the set of means of each pattern loading within population p

$$\boldsymbol{a}^{(\mathbf{p})} = \left(a^{(p)}_{m_1}, \dots, a^{(p)}_{m_R}, a^{(p)}_{n_1}, \dots, a^{(p)}_{n_R}, a^{(p)}_{I_1}, \dots, a^{(p)}_{I_{N_{in}}} \right), \tag{2.5}$$

and (ii) its covariance $\Sigma^{(p)}$, a matrix of dimension $(2R + N_{in}) \times (2R + N_{in})$, whose elements are the pairwise covariances

$$\Sigma^{(p)}_{xy} = E\left[\left(x^{(p)} - a^{(p)}_x \right) \left(y^{(p)} - a^{(p)}_y \right) \right] \tag{2.6}$$

where $E[\cdot]$ indicates the expected value, and x and y represent any pair of connectivity or input components. Within the loading space, each population therefore corresponds to a cluster centered at $\boldsymbol{a}^{(\mathbf{p})}$, and of shape specified by the connectivity matrix $\Sigma^{(p)}_{xy}$ (see Fig. 2.1B).

The geometrical arrangement between patterns is a key feature to understand the behavior of low-rank networks (Mastrogiuseppe and Ostojic, 2018). The connectivity and input patterns are N-dimensional vectors. To quantify the geometrical configuration between two patterns, we define the overlap, or normalized scalar product:

$$O(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} x_i y_i \tag{2.7}$$

where $\mathbf{x}$ and $\mathbf{y}$ are any two patterns in the set given by $\mathbf{m}^{(\mathbf{r})}, \mathbf{n}^{(\mathbf{r})}$ and $\mathbf{I}^{(\mathbf{s})}$. The overlap is the projection of pattern $\mathbf{x}$ onto $\mathbf{y}$, so that two patterns are orthogonal if and only if their overlap is zero.

An important property of rank-R matrices, such as the connectivity matrix J, is that their non-zero eigenvalues coincide with the eigenvalues of the overlap matrix J^{ov} (Nakatsukasa, 2019) that is defined by the overlaps between pairs of connectivity patterns:

$$J^{ov}_{rs} = O\left(\mathbf{m}^{(\mathbf{s})}, \mathbf{n}^{(\mathbf{r})} \right), \tag{2.8}$$

for $r, s = 1, \dots, R$. The eigenvalues of the connectivity matrix, and therefore of the overlap matrix, are an essential property to understand the dynamics of low-rank networks, as we show in Section 4. It is often more convenient to calculate the eigenspectrum of the overlap matrix J^{ov}, of size $R \times R$, than of the connectivity matrix J, of size $N \times N$.

In a network with P populations, any pattern $\mathbf{x}$ of length N can be represented as a set of P sub-patterns $\mathbf{x}^{(\mathbf{p})}$, for $p = 1, \dots, P$, where each sub-pattern has length $\alpha_p N$ and includes the components of neurons belonging to population p. Fig. 2.1 shows an example of a rank-two network with two populations, where the connectivity patterns can be split into two different sub-patterns of equal size (green and purple). The overlap between two patterns can then be expressed as a weighted average of the overlaps between sub-patterns:

$$O(\mathbf{x}, \mathbf{y}) = \sum_{p=1}^{P} \alpha_p O\left(\mathbf{x}^{(\mathbf{p})}, \mathbf{y}^{(\mathbf{p})} \right). \tag{2.9}$$

Even if the sub-patterns are not orthogonal to each other, i.e. the overlap between two sub-patterns is not zero, the patterns can be orthogonal to each other when the sub-pattern

overlaps cancel out. In the limit of large networks, the overlap between two sub-patterns $\mathbf{x^{(p)}}$ and $\mathbf{y^{(p)}}$ is given by the expected value over the distribution of the loadings in the population:

$$O\left(\mathbf{x^{(p)}}, \mathbf{y^{(p)}}\right) = E\left[x^{(p)} y^{(p)}\right] = a_x^{(p)} a_y^{(p)} + \Sigma_{xy}^{(p)}. \tag{2.10}$$

In order to define the overlap matrix in terms of the statistics of the different Gaussian populations, we define the matrix

$$\sigma_{n_r m_s}^{(p)} = \Sigma_{m_s n_r}^{(p)}. \tag{2.11}$$

The matrix $\boldsymbol{\sigma_{mn}}^{(p)}$ is a $R \times R$ whose entries contain the covariance between the connectivity patterns $\mathbf{m^{(r)}}$ and the $\mathbf{n^{(r)}}$ in population p. We call this matrix $\boldsymbol{\sigma_{mn}}^{(p)}$ a (reduced) covariance matrix, in an abuse of notation, because it is a subset of the covariance matrix $\Sigma^{(p)}$, and therefore it is not symmetric nor positive definite. For example, for a rank-one network, $\boldsymbol{\sigma_{mn}}^{(p)}$ is just a scalar, that can take any real value. For a rank-two network, $\boldsymbol{\sigma_{mn}}^{(p)}$ is a 2×2 matrix, whose entries are given by the four covariances $\sigma_{m_1 n_1}^{(p)}$, $\sigma_{m_1 n_2}^{(p)}$, $\sigma_{m_2 n_1}^{(p)}$, and $\sigma_{m_2 n_2}^{(p)}$.

Using Eqs. (2.9) and (2.10), we can characterize the overlap matrix J^{ov} as a function of the statistics of the connectivity sub-patterns:

$$J^{ov} = \sum_{p=1}^{P} \alpha_p \left(\boldsymbol{a_n^{(p)}} \boldsymbol{a_m^{(p)}}^T + \boldsymbol{\sigma_{mn}^{(p)}} \right), \tag{2.12}$$

where $\boldsymbol{a_n^{(p)}}$ and $\boldsymbol{a_m^{(p)}}$ are R dimensional vectors whose entries correspond to the corresponding subset of elements in $\boldsymbol{a^{(p)}}$ (Fig. 2.1C).

Similarly to the covariance matrix $\boldsymbol{\sigma_{mn}}$ that measures the correlations between connectivity patterns $\mathbf{m^{(r)}}$ and $\mathbf{n^{(r)}}$, we define the covariance $\sigma_{\mathbf{nI}}$ between the connectivity patterns $\mathbf{n^{(r)}}$ and the constant external input $\mathbf{I}$, as a vector of length R, where each component is defined as

$$\sigma_{n_r I}^{(p)} = \Sigma_{n_r I}^{(p)} \tag{2.13}$$

for $r = 1, \ldots, R$. We assume that the input loadings and loadings of the left connectivity patterns are uncorrelated, $\sigma_{m_r I}^{(p)} = 0$.

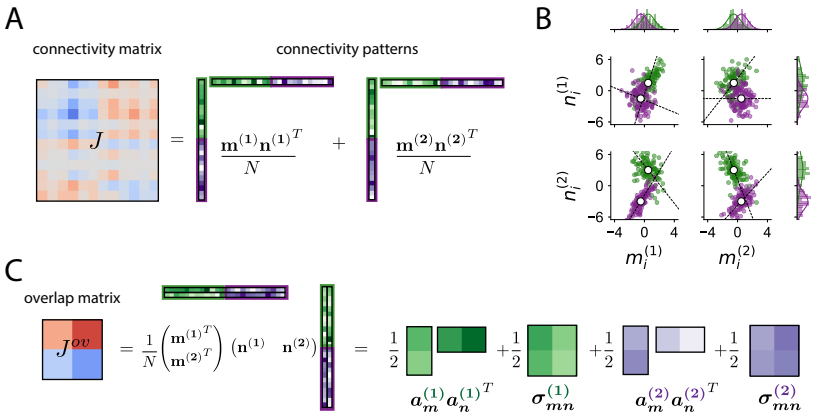

Figure 2.1: **Low-rank connectivity with Gaussian populations**. **A** The connectivity matrix J, rank-two in this illustration, is decomposed into the sum of two rank-one terms given by the outer product of the connectivity patterns $\mathbf{m}^{(\mathbf{r})}$ and $\mathbf{n}^{(\mathbf{r})}$, $r = 1, 2$. The components of the connectivity patterns – the pattern loadings – are grouped into two different sub-patterns (green and purple) with different population statistics. For visual purposes, the connectivity is shown only for 12 neurons in each population, the first 12 neurons belong to population 1 and the last 12 neurons belong to population 2. **B** Scatter plot of the distribution of pattern components in the four-dimensional loading space. Each dot corresponds to one neuron, and each neuron is characterized by its four values on the patterns $\mathbf{m}^{(\mathbf{r})}$ and $\mathbf{n}^{(\mathbf{r})}$, $r = 1, 2$. The color indicates whether the neuron belongs to the first population (green) or the second population (purple). The different populations are defined by different multivariate Gaussian statistics, means (white dots) and covariances (dashed lines), and define separate clusters. Population size $N = 200$, $\alpha_p = 0.5$. **C** Overlap matrix given by the inner product between connectivity patterns. The overlap matrix is a square matrix of size given by the rank of the connectivity, in this case 2×2. Its eigenvalues coincide with the non-zero eigenvalues of the $N \times N$ connectivity matrix. The overlap matrix can be expressed as a weighted sum over the overlaps of the different populations, as shown in Eq. (2.12).

2.3 Dynamics in Gaussian mixture low-rank networks

In this section, we summarize the three main properties of dynamics in mixture of Gaussian low-rank networks: (i) in a network of rank R, dynamics can be characterized by R collective variables that form a dynamical system; (ii) for loadings drawn from Gaussian mixture distributions, the dynamics can be further described as an effective circuit in which collective variables interact through gain-modulated effective couplings; (iii) with a sufficient number of populations, the resulting low-dimensional dynamics can approximate an arbitrary R-dimensional dynamical system.

Details of the derivations are provided in appendices 2.7.1 and 2.7.2.

2.3.1 Low-dimensional dynamics

In recurrent networks with low-rank connectivity, the dynamics of the trajectories $\mathbf{x}(t)$ are embedded in a linear subspace of dimension $R + N_{in}$ spanned by the left singular vectors $\mathbf{m}^{(\mathbf{r})}$ and the external input patterns $\mathbf{I}^{(\mathbf{s})}$, and can therefore be expressed as

$$x_i\left(t\right)=\sum_{r=1}^{R}\kappa_r m_i^{(r)}+\sum_{s=1}^{N_{in}}\kappa_{I_s}I_i^{(s)}. \tag{2.14}$$

Here κ_r and κ_{I_s} are collective variables that are obtained by projecting the activity $\mathbf{x}\left(t\right)$ on the patterns $\mathbf{m^{(r)}}$ and $\mathbf{I^{(s)}}$, that we assume orthogonal to each other. Introducing the trajectory $\mathbf{x}\left(t\right)$ expressed in this new basis into Eq. (2.1), the dynamics of the collective variables are then given by the following dynamical system:

$$\tau\frac{d\kappa_r}{dt}=-\kappa_r+\kappa_r^{rec} \tag{2.15}$$

$$\tau\frac{d\kappa_{I_s}}{dt}=-\kappa_{I_s}+u_s\left(t\right)$$

$$\kappa_r^{rec}=\frac{1}{N}\sum_{i=1}^{N}n_i^{(r)}\phi\left(\sum_{s=1}^{N_{in}}I_i^{(s)}\kappa_{I_s}+\sum_{l=1}^{R}m_i^{(l)}\kappa_l\right). \tag{2.16}$$

We focus in the following on networks receiving a constant input, so that there is only one collective variable κ_I along the input dimension, the value of which is constant. The recurrent connectivity contributes to the dynamics of κ_r through the term κ_r^{rec}.

The dynamics of collective variables in Eq. (2.15) are valid for any finite-size low-rank network, without any assumption on the values of pattern loadings. We next turn to networks where the pattern loadings are generated from specific distributions.

2.3.2 Dynamics in multi-population networks

For low-rank networks in which pattern loadings are generated for each neuron from a Gaussian mixture distribution, in the limit of large N the dynamics in Eq. (2.15) can be expressed in terms of the statistics of pattern loadings over the populations, and become (see appendix 2.7.1):

$$\tau\frac{d\kappa_r}{dt}=-\kappa_r+\kappa_r^{rec} \tag{2.17}$$

$$\kappa_r^{rec}=\sum_{p=1}^{P}\alpha_p\left[a_{n_r}^{(p)}\left\langle\phi\left(\mu^{(p)},\Delta^{(p)}\right)\right\rangle+\left(\sigma_{n_rI}^{(p)}\kappa_I+\sum_{s=1}^{R}\sigma_{n_rm_s}^{(p)}\kappa_s\right)\left\langle\phi'\left(\mu^{(p)},\Delta^{(p)}\right)\right\rangle\right]. \tag{2.18}$$

Here $\mu^{(p)}$ and $\Delta^{(p)}$ are the mean and variance of input to population p, given by

$$\mu^{(p)} = a_I^{(p)}\kappa_I+\sum_{s=1}^{R}a_{m_s}^{(p)}\kappa_s \tag{2.19}$$

$$\Delta^{(p)} = \sigma_{I^2}^{(p)}\kappa_I^2+\sum_{r=1}^{R}\sigma_{m_r^2}^{(p)}\kappa_r^2. \tag{2.20}$$

In Eq. 2.18, we used the Gaussian integral notation:

$$\left\langle f\left(\mu,\Delta\right)\right\rangle=\int dx\ \left(2\pi\right)^{-\frac{1}{2}}e^{-x^2/2}f\left(\mu+\sqrt{\Delta}x\right). \tag{2.21}$$

The factor $\left\langle\phi'\left(\mu^{(p)},\Delta^{(p)}\right)\right\rangle$ in Eq. (2.18) corresponds to the average gain of neurons in population p in a given state, specified by the mean $\mu^{(p)}$ and variance $\Delta^{(p)}$ of the inputs to

the population p. For each population, this average gain multiplies the covariances $\sigma^{(p)}_{m_l n_r}$ and $\sigma^{(p)}_{n_r I}$, and the corresponding average over populations defines an effective connectivity

$$\tilde{\sigma}_{xy} = \sum_{p=1}^{P} \alpha_p \sigma^{(p)}_{xy} \left\langle \phi' \left(\mu^{(p)}, \Delta^{(p)} \right) \right\rangle . \tag{2.22}$$

The contributions of the first-order statistics $a^{(p)}_{n_r}$ to the recurrent dynamics are modulated by the average firing rate in population p, and define an effective input

$$\tilde{a}_{n_r} = \sum_{p=1}^{P} \alpha_p a^{(p)}_{n} \left\langle \phi \left(\mu^{(p)}, \Delta^{(p)} \right) \right\rangle . \tag{2.23}$$

Introducing the effective connectivity and inputs into Eq. (2.17), the dynamics of a low-rank network with uncorrelated constant input take the simple form of an effective circuit of interacting collective variables:

$$\tau \frac{d\kappa_r}{dt} = -\kappa_r + \tilde{a}_{n_r} + \sum_{l=1}^{R} \tilde{\sigma}_{n_r m_l} \kappa_l . \tag{2.24}$$

Note that Eq. (2.24) describes the full non-linear dynamics in the limit $N \to \infty$. Although the collective variables interact linearly through the effective connectivity and inputs, those depend implicitly on κ_r. The overall dynamics are therefore non-linear, the non-linearity being fully encapsulated in the effective inputs and couplings.

2.3.3 Universal approximation of low-dimensional dynamical systems

By mapping the dynamics in Eqs. (2.17) and (2.24) to a feed-forward network with a single hidden layer, and exploiting the universal approximation theorem (Cybenko, 1989; Leshno et al., 1993), we can show that a Gaussian mixture network of rank R receiving a constant input is a universal approximator of R-dimensional dynamical systems (Appendix 2.7.2). More precisely, for a sufficient number of populations, the low-rank dynamics in Eq. (2.18) and (2.24) can approximate with arbitrary precision any R-dimensional dynamical system

$$\frac{d\boldsymbol{\kappa}}{dt} = G\left(\boldsymbol{\kappa}\right), \tag{2.25}$$

defined by a vector field

$$G\left(\{\kappa_r\}_{r=1\ldots R}\right) := \left(G_1\left(\{\kappa_r\}_{r=1\ldots R}\right), \ldots, G_R\left(\{\kappa_r\}_{r=1\ldots R}\right)\right) \tag{2.26}$$

over an arbitrary finite domain $\{\kappa_r\}_{r=1\ldots R} \in \left[\kappa_r^{min}, \kappa_r^{\max}\right]$. More specifically, this result requires that the vector field G is bounded and piecewise continuous, and the transfer function is not a polynomial (Appendix 2.7.2).

Alternatively, if the transfer function is bounded and monotonic, a rank-R network with multiple populations can approximate any vector field $G\left(\{\kappa_r\}_{r=1\ldots R}\right)$ over the full domain of the collective variables, $\{\kappa_r\}_{r=1\ldots R} \in [-\infty, +\infty]$, with the restriction that the vector field follows asymptotic leaky dynamics for large input values:

$$\lim_{\kappa_s \to \pm\infty} \frac{\partial G_r}{\partial \kappa_{r'}} \left(\kappa_1, \ldots, \kappa_r\right) = -\delta_{rr'} \tag{2.27}$$

for any values $s, r, r' = 1, \ldots, R$, where G_r represents the r-th component of the vector field as in Eq. (2.26), and δ_{ij} is the Kronecker delta. This stems from the fact that for large values of κ_r, the recurrent dynamics (Eq. 2.18) saturate to a constant value.

Note that the universal approximation theorem does not state how many populations P are required to implement a given dynamical system, and does not provide an algorithm for finding the statistics of the different populations.

2.4 Dynamics in networks with a single population

Having shown that a rank R network with an arbitrary number of populations can approximate any R-dimensional dynamical system, we now illustrate how having a small number of populations in contrast limits the possible dynamics.

We focus first on the case of networks consisting of a single Gaussian population. This case was previously studied for connectivities of rank one and two (Mastrogiuseppe and Ostojic, 2018; Schuessler et al., 2020a). Here we provide an overview of these results, and extend them to single-population networks of arbitrary rank. Specifically, we show that, independently of their rank, the range of dynamics such networks can implement is restricted. For simplicity, we focus on autonomous networks, with zero-mean connectivity patterns.

In vectorial form, assuming zero-mean connectivity patterns, the collective dynamics in Eq. (2.17) for one population read

$$\tau \frac{d\boldsymbol{\kappa}}{dt} = -\boldsymbol{\kappa} + \left\langle \phi' \left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa}\right) \right\rangle \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa}, \tag{2.28}$$

where we used the vector of collective variables $\boldsymbol{\kappa} \in \mathcal{R}^R$, and the $R \times R$ covariance matrix $\boldsymbol{\sigma_{mn}}$ as defined in Eq. (2.11), which is equal to the overlap matrix (Eq. 2.12) in the case of zero-mean connectivity patterns. Therefore, the eigenvalues of the covariance matrix $\boldsymbol{\sigma_{mn}}$, which for $N \to \infty$ are equivalent to the eigenvalues of the connectivity matrix, determine the dynamics in collective space (Schuessler et al., 2020a), as we review in the following analysis.

Fixed points The fixed points of Eq. 2.28 are given by

$$\boldsymbol{\kappa}_0 = \left\langle \phi' \left(0, \boldsymbol{\kappa}_0^T \boldsymbol{\kappa}_0\right) \right\rangle \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa}_0. \tag{2.29}$$

For $\phi(x) = \tanh(x)$, the trivial point $\boldsymbol{\kappa_0} = 0$ is always a solution. There might however be non-trivial fixed points depending on the eigenvalues of the covariance matrix $\boldsymbol{\sigma_{mn}}$. The covariance matrix can have up to R eigenvalues, that we denote λ_r, with associated eigenvector $\boldsymbol{u_r}$. Each real and non-degenerate eigenvalue λ_r of the covariance $\boldsymbol{\sigma_{mn}}$ generates a fixed point $\boldsymbol{\kappa_0^{(r)}} = \rho_r \boldsymbol{u_r}$, where ρ_r is the radial location of the fixed point along the direction set by the eigenvector $\boldsymbol{u_r}$. Introducing this parametrization of the fixed in Eq. 2.29, we obtain the following implicit equation for the value ρ_r:

$$1 = \lambda_r \left\langle \phi' \left(0, \rho_r^2\right) \right\rangle. \tag{2.30}$$

The gain factor $\left\langle \phi' \left(0, \rho_r^2\right) \right\rangle$ is bounded between 0 and 1 for the transfer function $\phi(x) = \tanh x$. Therefore, eigenvalues $\lambda_r > 1$ generate two non-trivial fixed points, symmetrically located around the origin (see Fig. 2.2 A-D, bottom row, for a rank-one example). Smaller eigenvalues do not generate any non-trivial fixed point (Fig. 2.2 A-D, first row).

In order to determine the stability of the fixed points, we linearize the dynamics and obtain the Jacobian S_r at the fixed point corresponding to the eigenvalue λ_r of $\boldsymbol{\sigma_{mn}}$ (see appendix 2.7.3)

$$S_r = -\boldsymbol{I} + \frac{1}{\lambda_r} \boldsymbol{\sigma_{mn}} + \left\langle \phi''' \left(0, \rho_r^2\right) \right\rangle \lambda_r \rho_r^2 \boldsymbol{u_r} \boldsymbol{u_r}^T, \tag{2.31}$$

where $\boldsymbol{I}$ denotes the $R \times R$ identity matrix. The eigenvalues of S_r determine the stability of the fixed points: if any positive eigenvalue exists, the dynamics will diverge away from the fixed point in the direction of the corresponding eigenvector. Negative eigenvalues correspond to attractive modes of the dynamics around the fixed point. If all eigenvalues of the stability matrix are negative, the fixed point is stable.

When the eigenvectors of the matrix $\boldsymbol{\sigma_{mn}}$ are orthogonal to each other (Fig. 2.3 A-D), the R eigenvalues of the matrix S_r, denoted as $\gamma_{r'}$ for $r' = 1 \dots R$, can be calculated analytically as shown in (Schuessler et al., 2020a). The eigenvalues $\gamma_{r'}$ have associated eigenvectors equal to the eigenvectors $\boldsymbol{u_{r'}}$ of the covariance matrix $\boldsymbol{\sigma_{mn}}$, and read

$$\gamma_{r'} = -1 + \frac{\lambda_{r'}}{\lambda_r} + \left\langle \phi'''\left(0, \rho_r^2\right)\right\rangle \lambda_r \rho_r^2 \delta_{rr'}. \tag{2.32}$$

Remarkably, the eigenvalues of the Jacobian around any non-trivial fixed point are therefore directly determined by the eigenvalues of connectivity and covariance matrices (Schuessler et al., 2020a). If $r' = r$, the two first terms cancel out, and the third term is always negative (see appendix 2.7.3). This implies that all non-trivial fixed points are stable in the direction $\boldsymbol{u_r}$ that points towards the origin. However, if there are other non-trivial fixed points corresponding to eigenvalues $\lambda_{r'} > \lambda_r$ of $\boldsymbol{\sigma_{mn}}$, the fixed point $\boldsymbol{\kappa_0^{(r)}}$ is destabilized in the directions of the eigenvectors with larger eigenvalue. When the eigenvectors are not orthogonal (Fig. 2.3 E-H), the eigenvectors of $\boldsymbol{\sigma_{mn}}$ are not necessarily eigenvectors of the linear stability matrix S_r. However, the same stability properties appear to hold: every fixed point is stable in the direction towards the origin, and the fixed point in the direction given by the largest eigenvalue is stable, while the other ones become unstable.

In summary, if all eigenvalues of the covariance matrix are real and non-degenerate, only the pair of non-trivial fixed points corresponding to the largest eigenvalue is stable. All the other non-trivial fixed points of the dynamics are saddle points. This implies that low-rank networks consisting of a single Gaussian population can have at most two stable fixed points independently of their rank.

Limit cycles Complex eigenvalues of the covariance matrix $\boldsymbol{\sigma_{mn}}$, if they exist, always appear in conjugate pairs. They lead to spiral dynamics around the origin, in the plane spanned by the real and imaginary part of the corresponding eigenvectors. If the real part of the complex eigenvalues is smaller than unity, $\mathrm{Re}\left(\lambda_r\right) < 1$, the spiral dynamics decay back to the origin. Otherwise, if $\mathrm{Re}\left(\lambda_r\right) > 1$, there is a limit cycle on the plane, around the origin. Similarly to the case with only real eigenvalues of the covariance matrix, if the real part of the complex eigenvalue is larger than the real part of any other eigenvalue of $\boldsymbol{\sigma_{mn}}$, any trajectory will converge to the plane defined by the real and imaginary parts of the corresponding eigenvectors. On this plane, we then find that the limit cycle is stable.

To illustrate this case, we consider a rank two network with a covariance matrix of the form

$$\boldsymbol{\sigma_{mn}} = \begin{pmatrix} \sigma & -\sigma_\omega \\ \sigma_\omega & \sigma \end{pmatrix}, \tag{2.33}$$

which has eigenvalues $\sigma \pm i\sigma_\omega$. Fig. 2.4 A-B shows an example of a network with such connectivity.

We can then write the equations for a rank-two network in polar form. Using the mapping to polar coordinates $\kappa_1 := \rho\cos\theta$ and $\kappa_2 := \rho\sin\theta$, the dynamics in Eq. (2.28) become

$$\tau\frac{d\rho}{dt} = -\rho + \rho\sigma\left\langle \phi'\left(0, \rho^2\right)\right\rangle \tag{2.34}$$

$$\tau\frac{d\theta}{dt} = \sigma_\omega\left\langle \phi'\left(0, \rho^2\right)\right\rangle. \tag{2.35}$$

When the real part σ of the eigenvalues is larger than one, the flow in the radial direction cancels at a value ρ_0 given by Eq. (2.30), which yields

$$\sigma^{-1} = \left\langle \phi'\left(0, \rho_0^2\right)\right\rangle. \tag{2.36}$$

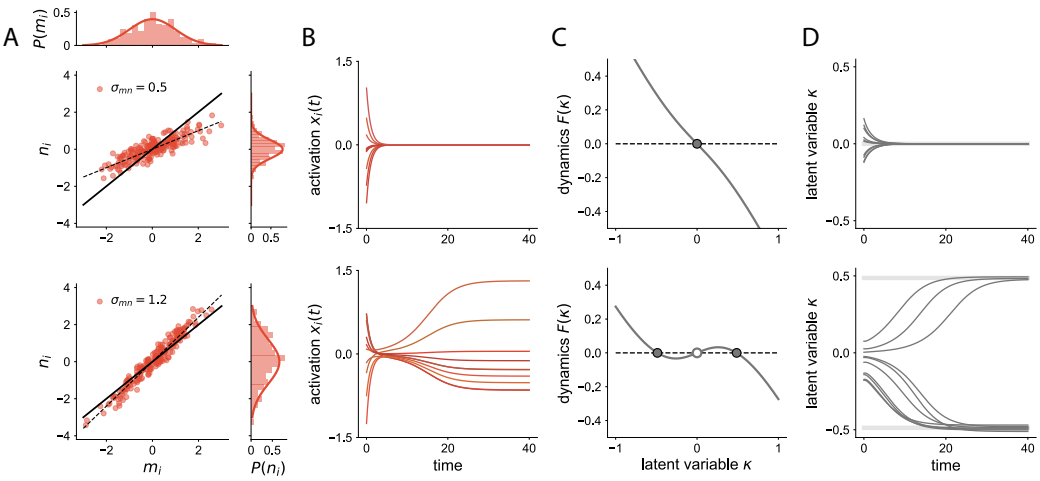

FIGURE 2.2: **Dynamics in rank-one networks with a single Gaussian population**. **A** Scatter plot of the loadings of left singular vectors $m_i^{(r)}$ and right singular vectors $n_i^{(r)}$. Top: Covariance σ_{mn}, indicated by the slope of the dashed line, below the critical value for non-trivial fixed points (solid line). Bottom: Covariance σ_{mn} beyond the critical value. **B** Dynamics of the activation variable $x_i(t)$ of ten units in the network for the two different networks initialized at random values. The network with σ_{mn} larger than 1 (bottom) converges to a heterogeneous fixed point, while the other one decays to zero. **C** One dimensional dynamics corresponding to the right hand side of Eq. (2.28). Filled dots correspond to stable fixed points. For a weak covariance between connectivity patterns (top), the trivial fixed point is the only fixed point. For a strong covariance (bottom), the recurrent connectivity generates two non-trivial stable fixed points. **D** Evolution of the collective variable κ as a function of time in a finite-size network, defined as the projection of the activity $\mathbf{x}(t)$ onto the connectivity pattern $\mathbf{m}$. Each curve corresponds to a different realization of the random connectivity matrix. $N = 1000$, top row: $\sigma_{n^2} = 0.34$, bottom row $\sigma_{n^2} = 1.52$.

Based on Eq. (2.34), we observe that any perturbation in the plane away from the limit cycle makes the radial component ρ go back to ρ_0. The limit cycle is therefore stable, as shown in Fig. 2.4 C.

Introducing this result into Eq. (2.35), we obtain that the oscillations of the limit cycle are generated at a frequency

$$\omega_{LC} = \frac{\sigma_\omega}{\sigma}. \tag{2.37}$$

In this analysis, Eq. (2.37) is derived for the particular covariance matrix $\boldsymbol{\sigma_{mn}}$ in Eq. (2.33), which is antisymmetric. However, numerical explorations suggest that this equation is valid more generally, for any connectivity matrix with a pair of complex eigenvalues. When the covariance matrix is not antisymmetric but still has complex eigenvalue, the limit cycle is no longer a circle but resembles an ellipse (see Fig. 2.4 G, grey trajectory, or Mastrogiuseppe and Ostojic (2018), Fig S8).

Figure 2.4 E-H shows an example of a rank-three network, whose connectivity matrix has a real eigenvalue λ_1 and a pair of complex conjugate eigenvalues λ_2 and λ_3. The real part of all eigenvalues is larger than one, so that the real eigenvalue leads to a pair of fixed points, and the complex eigenvalues generate a limit cycle. Given that in this example the real eigenvalue λ_1 is larger than the real part of the other eigenvalues, the fixed points are stable. The limit cycle is marginally stable in the plane spanned by the real and imaginary parts of the complex eigenvector of λ_2, but unstable in any other direction. Therefore,

trajectories starting in the plane converge to the limit cycle in the mean-field equation (see grey trajectory in Fig. 2.4 G). Small perturbations, such as those introduced by finite-size effects, make these trajectories deviate from the limit cycle and converge to one of the two stable fixed points (grey trajectory, Fig. 2.4 H).

Slow manifolds When the covariance matrix $\boldsymbol{\sigma_{mn}}$ has degenerate eigenvalues, low-rank RNNs can lead to other phenomena than discrete fixed points or limit cycles. As an example of degenerate eigenvalues, we study the network dynamics when the covariance matrix $\boldsymbol{\sigma_{mn}}$ is diagonal:

$$\boldsymbol{\sigma_{mn}} = \sigma_{mn}\boldsymbol{I}. \tag{2.38}$$

This covariance matrix has one single real eigenvalue σ_{mn}, which is degenerate, since it has R linearly independent eigenvectors. Introducing the covariance matrix in Eq. (2.38) into the dynamics in Eq. (2.28) we obtain the fixed point equation

$$\boldsymbol{\kappa}_0 = \left\langle \phi'\left(0, \boldsymbol{\kappa}_0^T\boldsymbol{\kappa}_0\right)\right\rangle \sigma_{mn}\boldsymbol{\kappa}_0. \tag{2.39}$$

To solve the fixed point equation, as in the previous section, we use the ansatz $\boldsymbol{\kappa}_0 = \rho_0\boldsymbol{u_{\kappa_0}}$, where $\boldsymbol{u_{\kappa_0}}$ is an arbitrary unitary vector in collective space. Introducing the ansatz in the fixed point equation (Eq. 2.39), we find that there is a non-trivial solution given implicitly by the scalar equation $\left\langle \phi'\left(0, \rho_0^2\right)\right\rangle = \sigma_{mn}^{-1}$, which is independent of the particular direction $\boldsymbol{u_{\kappa_0}}$. Furthermore, we find that the fixed point is stable in the direction $\boldsymbol{u_{\kappa_0}}$. Therefore, in the mean-field limit given by Eq. (2.28), this degenerate connectivity leads to a continuous manifold of attractive states that are at an equal distant ρ_0 away from the origin. In the case of rank-two connectivity, this degenerate covariance matrix leads to a stable ring attractor (Fig 2.3I-K), and in rank-R, to a stable R-spherical attractor.

In finite-size simulations, the sampling of random loadings introduces spurious correlations in the matrix $\boldsymbol{\sigma_{mn}}$, breaking the degeneracy of the eigenvalues. As a consequence, only a small number of points on the continuous attractor predicted by the mean-field theory give rise to actual fixed points. While the rest of the points on the predicted continuous attractor are not fixed points of the finite-size network, the dynamics around them are typically slow. More specifically, any trajectory of activity quickly converges towards the predicted continuous attractor, and then slowly evolves along it until it reaches a fixed point (Fig 2.3L) (Mastrogiuseppe and Ostojic, 2018). In finite-size networks, the continuous attractor predicted by the mean-field analysis therefore gives rise to a low-dimensional manifold in state space, along which the dynamics are slow.

When degenerate and non-degenerate real and complex eigenvalues are combined, the global stability appears to be given by the criterion in Eq. (2.32): each eigenvalue generates its corresponding non-trivial dynamics (fixed points, continuous attractors or limit cycle) independently. The stability of these dynamical phenomena depends on the global eigenspectrum: the eigenvalues with the largest real part generate stable attractors, while the other eigenvalues lead to repellers.

In summary, in a low-rank network consisting of a single Gaussian population, the possible non-trivial steady states are a pair of fixed points, a limit cycle, or a continuous attractor that gives rise to a small number of fixed points in finite networks. On top of these limited range of stable solutions, increasing the rank leads to additional unstable fixed points and limit cycles, that can potentially be used to control the dynamics, a point we do not further explore here. We instead proceed to show that increasing the number of Gaussian populations allows networks to implement a larger range of stable dynamics.

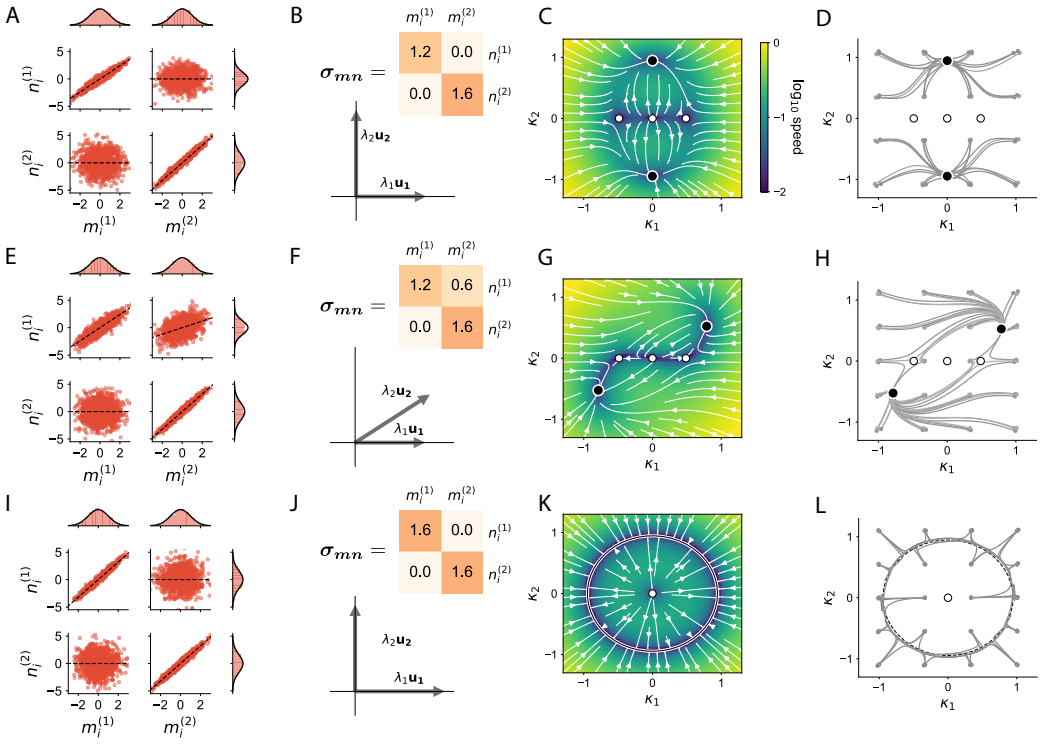

FIGURE 2.3: **Dynamics in rank-two networks with a single Gaussian population - Connectivity matrix with real eigenvalues**. **A** Scatter plot of the loadings of left singular vectors $m_i^{(r)}$ and right singular vectors $n_i^{(r)}$. **B** Covariance matrix $\boldsymbol{\sigma_{mn}}$ of the population (top), and its eigenvectors (bottom). **C** Vector field corresponding to the mean-field dynamics in the plane $\kappa_1 - \kappa_2$ of collective variables (Eq. 2.28). The colormap represents the speed of the dynamics, defined as the norm of vector $\frac{d\boldsymbol{\kappa}}{dt}$, in different points of the collective space. Two non-trivial fixed points are generated in the direction of each eigenvector. Black dots correspond to stable fixed points, while white dots are unstable or saddle points. The pair of fixed points corresponding to the largest eigenvalue is stable. **D** Finite-size simulations of the dynamics. Three different connectivity realizations are shown from each initial condition. $N = 1000$. **E-H** Similar to **A-D** for a network where the eigenvectors of the covariance matrix are not orthogonal (overlap between the connectivity patterns of different rank-one structures $\sigma_{m_2 n_1} \neq 0$). The eigenvector with largest eigenvalue generates a pair of stable fixed points. **I-L** Similar to **A-D** for a network with degenerate eigenvalues: any vector in the plane spanned by vectors $\boldsymbol{m}^{(1)}$ and $\boldsymbol{m}^{(2)}$ is an eigenvector of the connectivity. This symmetry leads to a continuous attractor in the mean-field dynamics. In finite size simulations (one matrix realization shown in **L**) the continuous attractor corresponds to a slow manifold on which usually two stable fixed points lie.

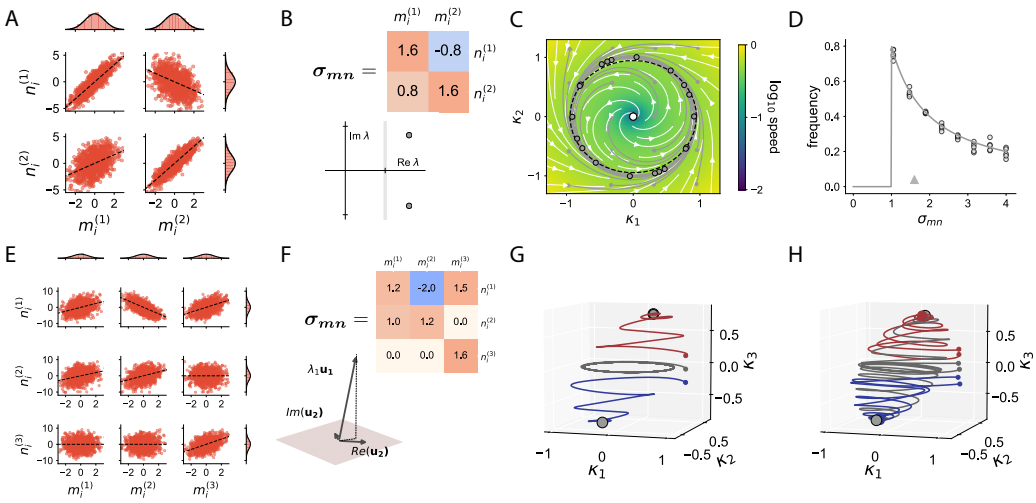

Figure 2.4: **Dynamics in rank-two networks with a single Gaussian population - connectivity matrix with complex eigenvalues**. **A** Scatter plot between the components of connectivity patterns $m_i^{(r)}$ and $n_i^{(r)}$, following the statistics given in Eq. (2.33), $\sigma = 1.6$ and $\sigma_\omega = 0.8$. **B** Covariance matrix of the singular vectors (top) and its eigenvalues in the complex plane, given by $\sigma \pm i\sigma_\omega$. **C** Vector field of the mean-field dynamics (Eq. 2.28). The colormap represents the speed of the dynamics, defined as the norm of vector $\frac{d\boldsymbol{\kappa}}{dt}$. Given that the real part of the eigenvalue is larger than one, a limit cycle (indicated by the dashed line) emerges in collective space. The grey lines correspond to finite-size simulations of the network, starting at different initial conditions with the same connectivity matrix. **D** Frequency of the limit cycle for different values of the symmetric part of the connectivity σ and fixed imaginary part $\sigma_\omega = 0.8$. The dots show the numerically estimated frequency of oscillations in finite-size simulations for five different network realizations. The line corresponds to Eq. (2.37). The triangle indicates the parameter σ used in **A-C**. **E-F** Analogous to **A-B**, for a rank-three network with one pair of complex eigenvalues and one real eigenvalue. The real eigenvalue ($\lambda_1 = 1.6$) is larger than the real part of the complex eigenvalues ($\mathrm{Re}(\lambda_2) = 1.2$). The real eigenvector $\mathbf{u_1}$ and the real and imaginary parts of the complex eigenvector $\mathbf{u_2}$ are plotted in **F**, bottom. The imaginary and real parts of the eigenvector $\mathbf{u_2}$ span the horizontal plane (shaded in grey). **G** Mean-field dynamics (Eq. 2.28) for three trajectories starting at different initial conditions. Each color indicates a different trajectory. When the network is initialized in the horizontal plane (grey trajectory), the activity ends at a limit cycle. Otherwise it converges to one of the two stable fixed points, located in the direction of the eigenvector $\mathbf{u_1}$. **H** Same trajectories as in **G**, in finite-size simulations, for three different connectivity matrices. The trajectories always end up in one of the two stable fixed points, even if initialized in the horizontal plane (grey trajectories). $N = 1000$.

2.5 Dynamics in networks with multiple populations

As described in the previous section, a major limitation of rank-R networks consisting of a single Gaussian population is that they cannot give rise to more than two stable fixed points, symmetrically arranged around the origin. We next show that networks consisting of several Gaussian populations can exhibit a larger number of stable fixed points. We specifically describe two different mechanisms by which multiple fixed points can be generated and controlled.

Non-linear gain control We first consider an autonomous rank-one network with zero-mean connectivity patterns consisting of two populations. We examine how this setup can lead to three stable fixed points, one at the origin, and two symmetrically arranged at non-zero values of the collective variable κ.

Every neuron in the network belongs to one of two populations, each population being defined by different statistics of pattern loadings. Within population p, for $p = 1, 2$, the joint distribution of n and m values over neurons is specified by a 2×2 covariance matrix $\Sigma^{(p)}$, while for simplicity we take the mean of the distribution to be zero. In the two-dimensional loading space defined by m and n, the two populations correspond to different Gaussian clusters, both centered at zero but with different shape and orientations (green and purple dots in Fig. 2.5 A).

Neurons belonging to each population are defined by different statistics of the loadings $n^{(p)}$ and $m^{(p)}$, for populations $p = 1, 2$ which have zero mean. The recurrent dynamics are determined by the overlaps $\sigma^{(1)}_{mn}$ and $\sigma^{(2)}_{mn}$ between the loadings n and m, and by the variance of the m loadings in each population $\sigma^{(1)}_{m^2}$ and $\sigma^{(2)}_{m^2}$. Indeed, the dynamics of the collective variable κ in Eq. (2.24) read:

$$\tau \frac{d\kappa}{dt} = -\kappa + \tilde{\sigma}_{mn}\kappa, \tag{2.40}$$

with the effective feedback $\tilde{\sigma}_{mn}$ defined as

$$\tilde{\sigma}_{mn} = \frac{1}{2}\sigma^{(1)}_{mn} \left\langle \phi' \left(0, \kappa^2 \sigma^{(1)}_{m^2}\right) \right\rangle + \frac{1}{2}\sigma^{(2)}_{mn} \left\langle \phi' \left(0, \kappa^2 \sigma^{(2)}_{m^2}\right) \right\rangle. \tag{2.41}$$

This effective feedback $\tilde{\sigma}_{mn}$ is set by the average of covariances $\sigma^{(p)}_{mn}$ for each population p, weighted by the gain of the population. If the two populations have different variances $\sigma^{(p)}_{m^2}$, their gains will vary differently with κ. If moreover the different populations have covariances $\sigma^{(p)}_{mn}$ of different signs, the total effective feedback will vary strongly with κ, while this is not the case in networks with uniform populations or a single one.

This network can have three stable fixed points (the origin and a pair of symmetrical non-trivial fixed points) if the effective feedback $\tilde{\sigma}_{mn}$ has a different sign in different regions of the collective space. First, the origin $\kappa = 0$ is always a fixed point of dynamics in Eq. (2.40). The origin is moreover a stable fixed point if the effective feedback at zero, which is given by $\frac{1}{2}\left(\sigma^{(1)}_{mn} + \sigma^{(2)}_{mn}\right)$, is smaller than 1. Therefore, one of the populations, which we define to be the first one ($p = 1$), must have a strong negative overlap, $\sigma^{(1)}_{mn} < 2 - \sigma^{(2)}_{mn} < 0$. Second, at large values of κ the effective feedback $\tilde{\sigma}_{mn}$ should be positive to cancel the contribution of the leaky term $-\kappa$ and generate a non-trivial fixed point. Given Eq. 2.41, this implies that the gain of the positively correlated population two should be large, whereas the gain of the negatively correlated population one should be close to zero. A small gain is achieved in the first population by having a large value $\sigma^{(1)}_{m^2}$, so that the second condition reads $\sigma^{(1)}_{m^2} \gg \sigma^{(2)}_{m^2}$. Fig. 2.5B-C shows the dynamics of such a network given by the mean-field equation and in finite-size networks.

More generally, with more than two populations this mechanism can be extended to produce a larger number of stable fixed points in rank-one networks. The key principle of this mechanism is to control independently the gain of the different populations, so that the contribution of each population to the effective feedback takes place at different ranges of the collective variable κ, and to have covariances $\sigma_{mn}^{(p)}$ of different signs, so that the effective feedback can flexibly take both positive and negative values in different ranges of κ. These mechanisms can also be applied to networks with rank higher than one. In that case, the overlap between loadings is given by a matrix $\boldsymbol{\sigma_{mn}^{(p)}}$ instead of a scalar, while the gain of each population is a scalar value. Populations with different covariance matrices and gains that vary at different ranges of the collective variables are able to generate multiple fixed points in different regions of the collective space, or combinations between stable limit cycles and stable fixed points (Dubreuil et al., 2020).

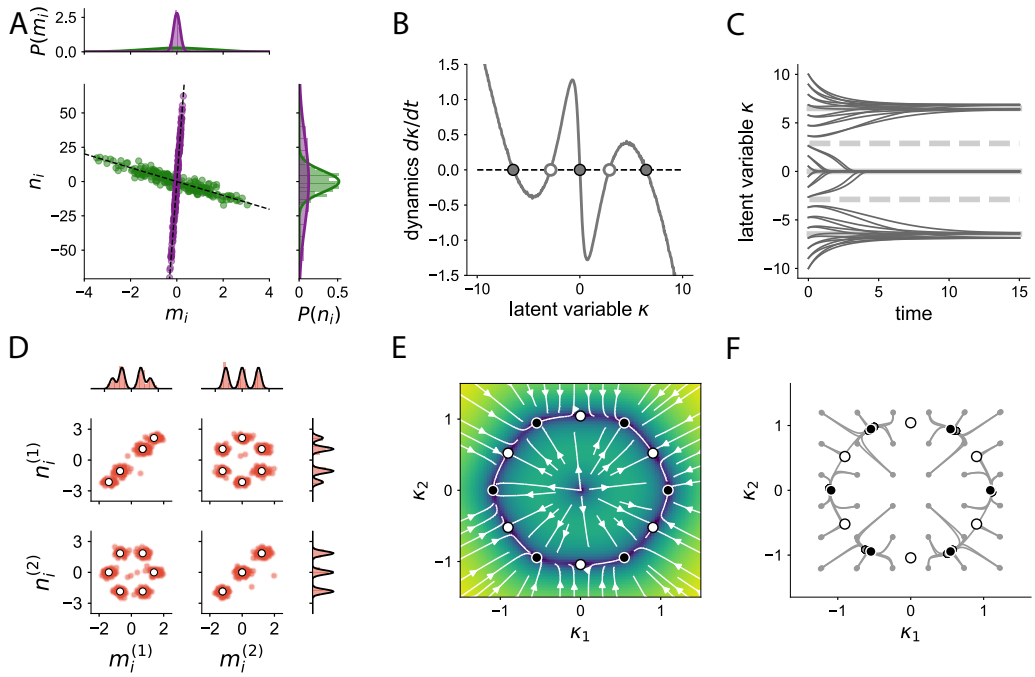

FIGURE 2.5: **Dynamics in low-rank networks with multiple populations**. **A** Scatter plot between the components of the connectivity patterns m_i and n_i in a rank-one network with two Gaussian populations, shown in green (negatively correlated population) and purple (positively correlated population). **B** Mean-field dynamics generated by the two-population statistics. Three stable fixed points (filled grey dots) emerge in the 1D recurrent dynamics. **C** Dynamics of the collective variable κ in a network with $N = 1000$ units, initiated at different initial values. The dynamics converge to one of the three stable fixed points. **D** Similar to **A**, for a rank-two network consisting of six statistical populations, with centers located on the vertices of a regular hexagon. **E** Mean-field dynamics of the network, the colormap represents the speed of the dynamics, defined as the norm of vector $\frac{d\boldsymbol{\kappa}}{dt}$ (blue: slow dynamics, yellow: fast dynamics). The hexagonal symmetry in the loadings produces a solution with hexagonal symmetry, with six stable fixed points (black dots) symmetrically arranged along a ring. Saddle points (white dots) appear between the stable fixed points. **F** Trajectories of the collective variables in finite-size simulations, initiated at different initial conditions. All trajectories converge to one of the six stable fixed points. Two different network realizations are shown for each initial condition. Parameters in **A-C**: $\sigma_{mn}^{(1)} = -10, \sigma_{mn}^{(2)} = 4.5, \sigma_{m^2}^{(1)} = 1.98, \sigma_{m^2}^{(2)} = 0.02$, and $\alpha_1 = \alpha_2 = 0.5$. Parameters in **D-F**: centers arranged as in Eqs. (2.42) and (2.43) where $p = 6$ and $R_n = 1.5$. Variance $\sigma_{m^2} = 0.3$. Network size $N = 1000$.

Symmetries in loading space In low-rank networks, a second mechanism for generating multiple fixed points is to exploit symmetries in the distribution of loadings $P(\underline{m}, \underline{n})$. Indeed a symmetry in the distribution of loadings $P(\underline{m}, \underline{n})$ implies a symmetry in the dynamics of the collective variables. In consequence, if a network with symmetry generates a non-trivial stable fixed point, symmetric points in the collective space will also correspond to stable fixed points. Classical Hopfield networks (Hopfield, 1982) are a prominent instance of this mechanism, where multiple stable fixed points are generated based on symmetries in the connectivity.

In this section, we first illustrate how symmetries in connectivity lead to multiple symmetric fixed points. We then explicitly show that Hopfield networks in the limit of a small number of stored patterns correspond to a special case of Gaussian mixture low-rank networks with symmetric connectivity. Throughout this section, we focus on networks where the overlap between the connectivity patterns is given by the non-zero means of the loadings, which is complementary to the previous section where the connectivity patterns had zero mean and the recurrent dynamics is determined by the covariances between the loadings.

As an illustration, we consider first a rank-two network, with units evenly split into P populations. In each population, the loadings $m_1^{(p)}, m_2^{(p)}, n_1^{(p)}, n_2^{(p)}$ have a different set of means $a_{m_1}^{(p)}$, $a_{m_2}^{(p)}$, $a_{n_1}^{(p)}$, $a_{n_2}^{(p)}$ and the covariances $\sigma_{m_r n_s}^{(p)}$ are zero. The variance of the loadings, σ_{m^2} and σ_{n^2}, are identical in all populations. As a consequence, different populations correspond to clusters of identical spherical shape, but centered at different points in the four-dimensional loading space.

We specifically arrange the means of the different populations (centers of the different clusters) symmetrically at the vertices of a regular polygon in the planes of loadings $m_1 - m_2$ and $n_1 - n_2$:

$$a_{m_1}^{(p)} = R_m \cos\left(\frac{2\pi p}{P}\right), \quad a_{m_2}^{(p)} = R_m \sin\left(\frac{2\pi p}{P}\right); \tag{2.42}$$

$$a_{n_1}^{(p)} = R_n \cos\left(\frac{2\pi p}{P}\right), \quad a_{n_2}^{(p)} = R_n \sin\left(\frac{2\pi p}{P}\right); \tag{2.43}$$

where p is the population index, $p = 1 \dots P$. The radial distance R_m is fixed so that the patterns $\mathbf{m^{(1)}}$ and $\mathbf{m^{(2)}}$ have unit variance, while the free parameter R_n controls the overlap between the connectivity patterns. Figure 2.5D shows an example with six populations, $P = 6$. This distribution has a discrete rotational symmetry of order P, since rotations of angle $2\pi/P$ in the planes $m_1 - n_2$ and $m_2 - n_1$ leave the distribution unchanged.

Using the mean-field description in Eq. (2.17), the dynamics of the two collective variables now read

$$\tau \frac{d\kappa_1}{dt} = -\kappa_1 + \frac{1}{P}\sum_{p=1}^{P} a_{n_1}^{(p)} \left\langle \phi\left(a_{m_1}^{(p)}\kappa_1 + a_{m_2}^{(p)}\kappa_2, \sigma_m^2\left(\kappa_1^2 + \kappa_2^2\right)\right)\right\rangle \tag{2.44}$$

$$\tau \frac{d\kappa_2}{dt} = -\kappa_2 + \frac{1}{P}\sum_{p=1}^{P} a_{n_2}^{(p)} \left\langle \phi\left(a_{m_1}^{(p)}\kappa_1 + a_{m_2}^{(p)}\kappa_2, \sigma_m^2\left(\kappa_1^2 + \kappa_2^2\right)\right)\right\rangle. \tag{2.45}$$

Given the symmetry in the distribution, if we identify one non-trivial stable fixed point, there will be at least $P - 1$ other fixed points with the same stability. Focusing on the direction given by $\kappa_2 = 0$, the velocity in the κ_2 direction, given by Eq. (2.45), is always zero due to the symmetry in the distribution. Therefore, we obtain a fixed point equation for κ_1 on the $\kappa_2 = 0$ direction using Eq. (2.44):

$$\kappa_1 = \frac{1}{P}\sum_{p=1}^{P} R_n \cos\left(\frac{2\pi p}{P}\right) \left\langle \phi\left(R_m \cos\left(\frac{2\pi p}{P}\right)\kappa_1, \sigma_m^2 \kappa_1^2\right)\right\rangle. \tag{2.46}$$

The r.h.s. is a sum of P monotonically increasing bounded functions of κ_1. If the slope at the origin is larger than one, then, the r.h.s. will intersect with the function κ_1 at a non-trivial point. The slope of the r.h.s at the origin, obtained by differentiating the r.h.s. with respect to κ_1 and evaluating at $\kappa_1 = 0$, is $\frac{1}{2}R_n R_m$, so that a condition for a non-trivial fixed point is

$$R_n R_m > 2. \tag{2.47}$$

Because of the symmetry, if $R_m R_n > 2$, there are at least P stable fixed points arranged symmetrically on a circle (Fig 2.5 E-F). If the number of population pairs is odd, there are $2P$ stable fixed points symmetrically arranged on a circle, because there is also a symmetry with respect to the origin, imposed by the symmetry in the transfer function. Otherwise, if P is even, P stable fixed points are generated by the network.

Symmetrical arrangements of multiple populations can also be used in higher R-rank networks to obtain multiple stable fixed points located on a R-dimensional sphere. For example, in rank-three networks, we consider eight populations whose centers are arranged at the vertices of a cube. The centers of the eight populations in the three-dimensional space of loadings $m^{(r)}$, for $r = 1, 2, 3$, correspond to the vertices of a cube with side $2R_m$, so that

$$\left(a_{m_1}^{(p)}, a_{m_2}^{(p)}, a_{m_3}^{(p)}\right) = (\pm R_m, \pm R_m, \pm R_m)\,. \tag{2.48}$$

Populations $p = 1, \ldots, 8$ correspond to one of the eight different possible combinations of the sign. The variances of the loadings, σ_{m^2} is identical in all populations. The value of R_m is fixed so that the norm of each connectivity pattern $\mathbf{m^{(r)}}$ is N.

The centers of the $n^{(r)}$ loadings follow the same configuration, at the vertices of a cube of side $2R_n$:

$$\left(a_{n_1}^{(p)}, a_{n_2}^{(p)}, a_{n_3}^{(p)}\right) = (\pm R_n, \pm R_n, \pm R_n)\,, \tag{2.49}$$

where each population p correspond to the same combination of signs as for the m loadings, so that

$$\operatorname{sgn}\left(a_{m_r}^{(p)}\right) = \operatorname{sgn}\left(a_{n_r}^{(p)}\right), \tag{2.50}$$

with the collective index $r = 1, 2, 3$ and the population index $p = 1 \ldots 8$. The value R_n is, as in the previous case, a free parameter that controls the overlap between connectivity patterns. This configuration is shown in Fig. 2.6 D-E and G-H, for two different values of R_n. This distribution exhibits a cubic symmetry in the loading space $m_1 - m_2 - m_3$ and in space $n_1 - n_2 - n_3$. Thus, if we identify a non-trivial fixed point, these symmetries require the existence of symmetric solutions in the collective space. Inspecting the direction $\kappa_2 = \kappa_3 = 0$ in the dynamics, we obtain a criterion for having a non-trivial stable fixed point:

$$\kappa_1 = \frac{1}{8}\sum_{p=1}^{8} a_{n_1}^{(p)} \left\langle \phi\left(a_{m_1}^{(p)}\kappa_1, \sigma_m^2\kappa_1^2\right)\right\rangle \tag{2.51}$$

Eq. 2.51 has a non-trivial solution, which is always stable, if $R_n R_m > 1$. When this solution exists, applying a rotation of $\pi/2$ in the $m_1 - m_2$ plane and in the $m_1 - m_3$, it is possible to determine the other five stable fixed points that are generated by the symmetry (Fig. 2.6F). These stable fixed points are arranged in the collective space at the vertices of an octahedron, the dual polyhedron of the cube (the dual of a polyhedron A is the polyhedron B where the vertices of A correspond to the edges of B). Applying symmetry principles, the middle point of each triangular face of the octahedron is also a fixed point. However, the stability of this fixed point depends on the overlap $R_n R_m$. If $R_n R_m$ is larger than one but low, these fixed points are saddle points (Fig. 2.6F). Beyond a critical value of $R_n R_m$, these fixed points become also stable. This second set of fixed points consists of eight points arranged on a cube (Fig. 2.6I, blue dots).

In general, any K-dimensional discrete symmetry in the loadings (centers arranged within a regular polytope -generalization of a polyhedron in more than three dimensions-, symmetric with respect to the origin), will generate a dynamical system with stable fixed points on a K-dimensional sphere, arranged with the symmetry of the dual polytope.

Hopfield networks storing $R \ll N$ patterns can be seen as a particular limit of symmetric Gaussian-mixture low-rank networks. A Hopfield network is designed to store R binary

patterns $m_i^{(r)} = \pm m$, where for every neuron the sign of the entry in each pattern generated randomly, and m is a scalar parameter. A Hopfield network storing these R patterns is defined as a recurrent network with connectivity matrix

$$J_{ij}^{Hopfield} = \sum_{r=1}^{R} m_i^{(r)} m_j^{(r)} \tag{2.52}$$

Such a configuration creates two symmetric fixed points around the origin in the direction of each pattern $\mathbf{m^{(r)}}$, for large enough m.

Hopfield networks (Hopfield, 1982) correspond to a specific type of low-rank matrix, and can be mapped onto Gaussian-mixture low-rank networks. One of the specific properties of Hopfield networks (Eq. 2.52) is that the connectivity is symmetric, so that the left and right connectivity patterns are proportional to each other

$$\mathbf{m^{(r)}} = c\mathbf{n^{(r)}} \tag{2.53}$$

where c is a positive constant. Secondly, the loadings of the patterns $\mathbf{m^{(r)}}$ and $\mathbf{n^{(r)}}$, for $r = 1, \ldots, R$, are binary and of equal sign, so that each neuron is characterized by $2R$ loadings that can only differ from each other in their signs. Therefore, each neuron in a Hopfield network belongs to one of the 2^R sign combinations allowed. In terms of the low-rank framework, Hopfield networks can therefore be described as low-rank networks with 2^R deterministic populations, which have means

$$\left(a_{m_1}^{(p)}, \cdots, a_{m_R}^{(p)}\right) = R_m \left(\pm 1, \ldots, \pm 1\right), \tag{2.54}$$

$$\left(a_{n_1}^{(p)}, \cdots, a_{n_R}^{(p)}\right) = R_n \left(\pm 1, \ldots, \pm 1\right), \tag{2.55}$$

$$\operatorname{sgn}\left(a_{m_r}^{(p)}\right) = \operatorname{sgn}\left(a_{n_r}^{(p)}\right), \tag{2.56}$$

and where there is no dispersion around the mean of each population, so that $\sigma_m^{(p)} = \sigma_n^{(p)} = 0$.

A rank-two network with four populations $P = 4$ –characterized by Eq. (2.42), see Fig. 2.6 A-C,– is therefore equivalent to a two-pattern Hopfield network in the limit of no dispersion around the mean of each cluster, $\sigma_{m^2}^{(p)} = 0$. In this limit, saddle points are located at the midpoints between neighbouring stable fixed points. In the more general rank-two networks in Eq. (2.42) where $\sigma_{m^2}^{(p)} > 0$, the saddle points between stable fixed points move further away from the origin (such as in Fig. 2.6 B, where $\sigma_{m^2}^{(p)} = 0.3$), but the four stable fixed points remain on the vertices of a square along the axes $\kappa_1 = 0$ and $\kappa_2 = 0$. In the limit of very large $\sigma_{m^2}^{(p)}$ the saddle points between stable fixed points approach the circle that circumscribes the stable fixed points.

The rank-three network presented in Eqs. (2.49) and (2.50) also becomes a classical Hopfield network in the limit of $\sigma_{m^2}^{(p)} \to 0$. Allowing for values $\sigma_{m^2}^{(p)} > 0$, as illustrated in Fig. 2.6 D and G, does not change the number of fixed points generated by the Hopfield network nor their direction in collective space. These networks generate pairs of stable fixed points along the directions $\mathbf{m^{(1)}}$, $\mathbf{m^{(2)}}$, and $\mathbf{m^{(3)}}$. The additional fixed points along directions $\pm\mathbf{m^{(1)}} \pm \mathbf{m^{(2)}} \pm \mathbf{m^{(3)}}$, that become stable when $R_m R_n$ is large, correspond to well known spurious mixture states in Hopfield networks (Amit et al., 1987).

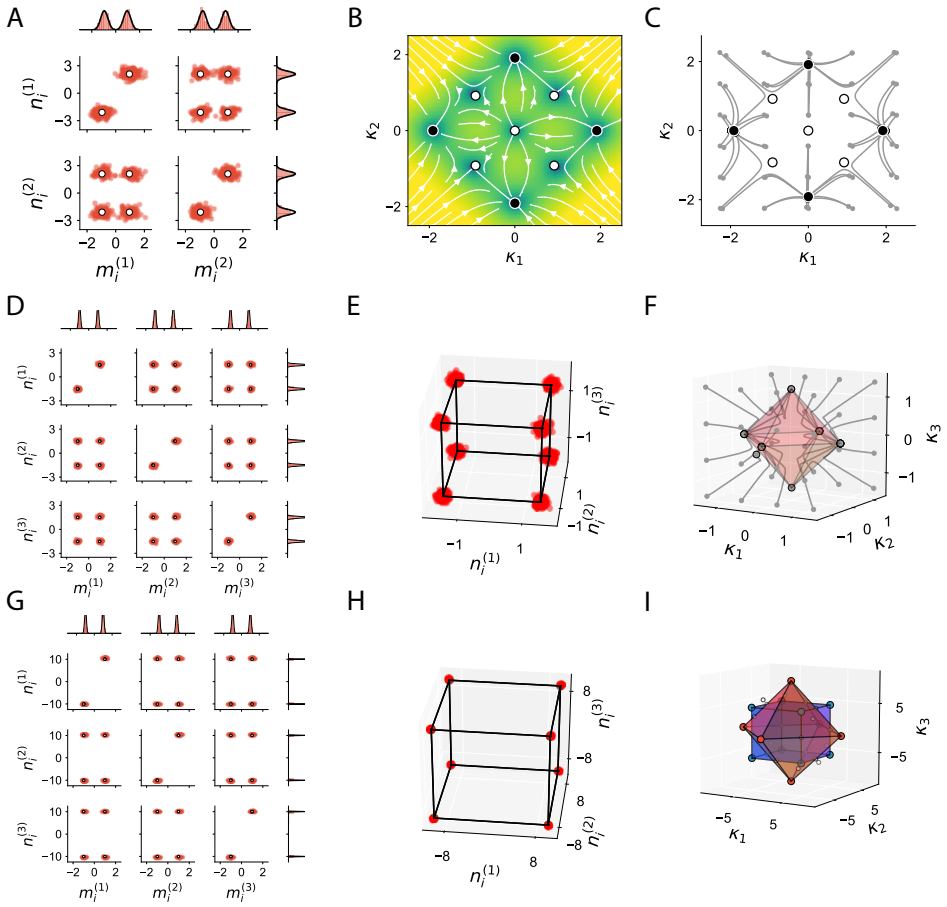

FIGURE 2.6: **Multiple populations in rank-R networks**. **A** Scatter plot between the entries of left singular vectors m_i and right singular vectors n_i in a rank-two network with four populations following Eqs. (2.42) and (2.43), with $P = 2$. Standard deviation of 0.3 around the mean of each population. **B** Corresponding mean-field dynamics. The colormap represents the speed of the dynamics, defined as the norm of vector $\frac{d\boldsymbol{\kappa}}{dt}$ (blue: slow dynamics, yellow: fast dynamics). Four stable fixed points emerge, arranged in a square. **C** Trajectories starting at different initial conditions in a finite-size network. Each initial condition shows trajectories for two network realizations. **D** Analogous to **A** in a rank-three network with loadings arranged as in Eqs. 2.48 and 2.49. **E** The populations are arranged at the vertices of a cube. $R_n = 2.1$. **F** Dynamics of the collective variables. Six stable fixed points (grey dots) emerge, arranged at the vertices of a dodecahedron (dual polygon of the cube, highlighted in red for visual purposes). Grey lines correspond to the trajectories of finite-size networks, initialized at different points in state-space. **G-I** Same as in **D-F**, but for a network whose populations have larger mean values, $R_n = 7$. For such large values, spurious fixed points that are proportional to the combinations of the three stored patterns $(\pm \boldsymbol{m}_1 \pm \boldsymbol{m}_2 \pm \boldsymbol{m}_3,)$ also become stable. Therefore, apart from the six fixed points in a octahedron (red polygon), eight other spurious fixed points appear arranged in a cube (blue polygon). Network size $N = 1000$.

2.6 Approximating dynamical systems with Gaussian-mixture low-rank networks

In the previous section, we focused on generating multiple fixed points in an autonomous network by means of a few Gaussian populations in the connectivity. More generally, as shown in Section 2.3, multi-population rank-R networks can approximate any R-dimensional dynamical system. In this section, we propose an algorithm to do so.

Previous works have developed algorithms for training recurrent networks to implement given dynamics that effectively used low-rank connectivity (Paulin, 2004; Pollock and Jazayeri, 2020; Rivkind and Barak, 2017). These methods rely on tuning the loadings $n_i^{(r)}$ of individual neurons, given fixed external inputs $I_i^{(s)}$ and connectivity loadings $m_i^{(r)}$. Here we focus instead on mixtures of Gaussian populations rather than individual units, and extend previous methods to find the first and second order moments of multiple Gaussian populations that approximate a given dynamical system.

Our goal is to approximate the R-dimensional dynamics specified by a vector field $G(\boldsymbol{\kappa})$:

$$\frac{d\boldsymbol{\kappa}}{dt} = G(\boldsymbol{\kappa}). \tag{2.57}$$

Our algorithm proceeds as follows. We first fix the number of Gaussian populations in the network and the fraction of neurons included in each population, α_p. Depending on the complexity of the approximated dynamics, a smaller or larger number of populations is required. Second, we set the mean and variance of the $\mathbf{m^{(r)}}$ vectors in each population, $a_{m_r}^{(p)}$ and $\sigma_{m_r^2}^{(p)}$, together with the mean and variance of the external input, $a_I^{(p)}$ and $\sigma_{I^2}^{(p)}$. Finally, we determine the statistics of the $\mathbf{n^{(r)}}$ vectors, the only unknown in the network, using linear regression.
We define a number of set points $\{\boldsymbol{\kappa}_k\}_{k=1\ldots K}$ on which we impose that the effective flow in the low-rank network given by Eq. (2.17) be equal to the target vector field

$$G(\boldsymbol{\kappa}_k) = -\boldsymbol{\kappa}_k + \sum_{p=1}^{P} \alpha_p \left(\boldsymbol{a_n^{(p)}} \left\langle \phi\left(\mu^{(p)}(\boldsymbol{\kappa}_k), \Delta^{(p)}(\boldsymbol{\kappa}_k)\right)\right\rangle + \boldsymbol{\sigma_{nm}^{(p)}} \boldsymbol{\kappa_k} \left\langle \phi'\left(\mu^{(p)}(\boldsymbol{\kappa}_k), \Delta^{(p)}(\boldsymbol{\kappa}_k)\right)\right\rangle\right). \tag{2.58}$$

These $k = 1\ldots K$ set points should be relevant points of the vector field $G(\boldsymbol{\kappa})$; they can be fixed points, but can also be chosen within a grid in collective space or based on sampled trajectories of the target system (Eq. 2.57). For simplicity, in Eq. (2.58) we are considering that the input pattern $\mathbf{I}$ is orthogonal to the connectivity patterns $\mathbf{n^{(r)}}$. It is possible to extend the algorithm to account for non-zero values of the parameters $\sigma_{n_r I}$.

Note that $\mu^{(p)}$ and $\Delta^{(p)}$ depend on the statistics of patterns $\mathbf{I}$ and $\mathbf{m^{(r)}}$ that are fixed (see Eq. (2.19)), but not on $a_{n_r}^{(p)}$ and $\sigma_{m_r n_r}^{(p)}$ which we aim to determine. Eq. (2.58) can therefore be written as a linear system of the form

$$\boldsymbol{G} = \boldsymbol{W}^T \mathbf{X} \tag{2.59}$$

where, for one single set point, $\boldsymbol{G}$ is a vector of length R, $\boldsymbol{G} = G(\boldsymbol{\kappa_k}) + \boldsymbol{\kappa_k}$, the vector

$$\boldsymbol{X} := \left[a_{n_1}^{(1)}, \ldots a_{n_R}^{(1)}, \sigma_{m_1 n_1}^{(1)}, \ldots \sigma_{m_1 n_R}^{(1)}, \ldots \sigma_{m_R n_1}^{(1)}, \ldots \sigma_{m_R n_R}^{(1)}, \ldots a_{n_1}^{(P)} \ldots \sigma_{m_R n_R}^{(P)}\right] \tag{2.60}$$

has length $R(R+1)P$ and the corresponding matrix $\boldsymbol{W}$ of size $R(R+1)P \times R$. For the K set points $\boldsymbol{\kappa}_k$ on which we want to approximate the dynamics, we concatenate the vector $\boldsymbol{G}$ and matrix W of each point, so that they will be of size $R \cdot K$ and $R \cdot K \times (R(R+1)P)$ respectively.

The unknown values of vector $\mathbf{X}$ can now be obtained by standard linear regression as

$$\boldsymbol{X} = \left(\boldsymbol{W}\boldsymbol{W}^T\right)^{-1} \boldsymbol{W}\boldsymbol{G}. \tag{2.61}$$

Often, it is convenient to regularize the regression algorithm to avoid the entries of $\boldsymbol{X}$ being exceedingly large, at the cost of increasing the error in the approximation of the dynamics. Solutions with very large values of $\boldsymbol{X}$ are less robust, because they produce stronger finite-size effects when sampling from the found mixture of Gaussians, potentially affecting the stability of the solution. One standard possibility amongst many is to use ridge regression to find the unknown values

$$\boldsymbol{X} = \left(\boldsymbol{W}\boldsymbol{W}^T + \beta^2 \boldsymbol{I}\right)^{-1} \boldsymbol{W}\boldsymbol{G} \tag{2.62}$$

where β is the ridge parameter that controls the amount of regularization.

The number of populations, together with the distributions chosen to fix the mean and covariance values $a_{m_r}^{(p)}$, $\sigma_{m_r^2}^{(p)}$, $a_I^{(p)}$ and $\sigma_{I^2}^{(p)}$ are hyperparameters of the algorithm. These hyperparameters can be tuned progressively by running several iterations of the algorithm. For example, a possible goal is to search for the the minimal number of populations required for approximating a given dynamical system within some accuracy limits.

To illustrate the algorithm, we use a rank-two network to approximate a Van der Pol oscillator. The Van der Pol oscillator is a two-dimensional non-linear dynamical system that generates non-harmonic oscillations. It is defined as

$$\frac{dx}{dt} = y \tag{2.63}$$

$$\frac{dy}{dt} = \mu\left(1 - x^2\right) y - x \tag{2.64}$$

where μ is a scalar parameter that controls the strength of the non-linearity. For this example, we set $\mu = 1$ (Fig. 2.7 A). We set the number of populations in the network to 50. Secondly, we determine the statistics for the left connectivity patterns and the external input, by drawing random values for the mean values in each population $a_I^{(p)}$ and $a_{m_r}^{(p)}$ from a zero-mean uniform distribution, and the variances $\sigma_{m_r^2}^{(p)}$ and $\sigma_{I^2}^{(p)}$ from an exponential distribution, all values of order one. As set points, we use a $K = 30 \times 30$ grid for values x and y ranging between -3 and 3.

Applying linear regression, we find that such a network can flawlessly approximate the Van der Pol oscillator in collective space using the mean-field equations (Fig. 2.7 B-D). However, it comes with the cost that the found parameters in $\boldsymbol{\Sigma}$ are orders of magnitude larger than the parameter values for $\sigma_{m_r^2}^{(p)}$ (Fig. 2.7 E). To reduce the norm of the solutions, we added ridge regression to the least square algorithm. Regularized solutions are able to decrease strongly the order of magnitude of the found parameters $\sigma_{n_r^2}^{(p)}$, while still producing limit cycles, although the approximation error is increased (Fig. 2.7 F-H).

This algorithm can be applied to generate any given dynamics in collective space within a finite domain. Beyond this finite domain sampled through the chosen set points, if the target vector field does not follow the required asymptotic behavior (Eq. 2.27), as it is the case for the Van der Pol oscillator, the network will not extrapolate to the target dynamics (region outside square of set points in Fig. 2.7 D and F). However, in practice, it may produce qualitatively similar dynamics: in the example of the Van der Pol oscillator, if the network is initialized at a point outside the limit cycle, the resulting trajectories still converge to the limit cycle.

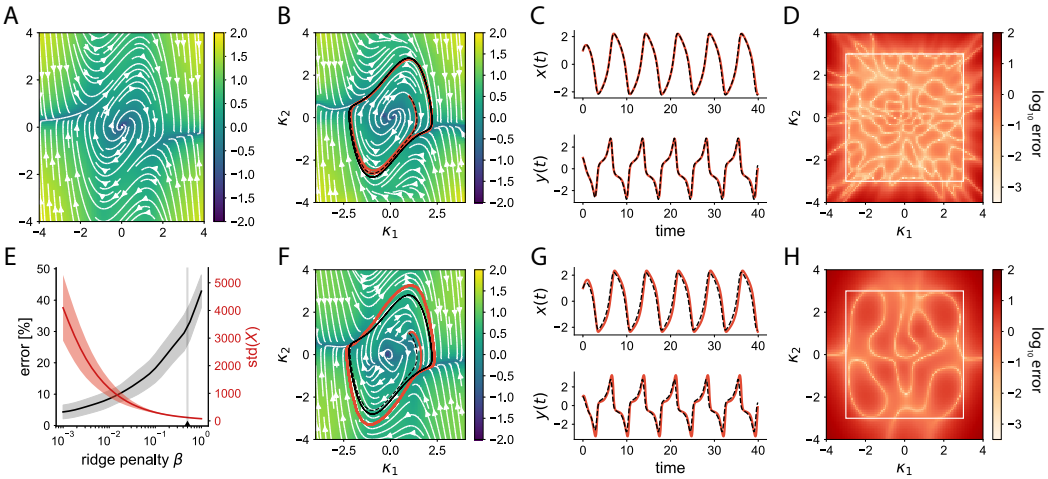

Figure 2.7: **Approximation of a Van der Pol oscillation with low-rank networks**. **A** Dynamics of a Van der Pol oscillator ($\mu = 1$). **B** Approximated dynamics by a low-rank network of 50 populations, calculated with no regularization. The trajectory of a Van der Pol oscillator initialized at $(1, 1)$ is shown in the black line. The corresponding trajectory in the low-rank network is shown in red. **C** Trajectories for the Van der Pol oscillator (black dashed line) and the low-rank network (mean-field, red line) as a function of time. **D** Heatmap of the logarithm of the error of approximation by the low-rank network (mean-field equations). The low-rank network is approximated on a grid spanned by the white square. **E** Approximation error (black) and standard deviation of the found parameters Σ as a function of the ridge parameter β. The shaded region corresponds to the standard deviation estimated from 10 different simulations. The triangle shows the regularization parameter chosen for **F-H**. **F-H** Same as **B-D** for a network with regularization parameter of $\beta = 0.5$. The network is able to produce stable limit cycles, with similar shape and frequency to those of the Van der Pol oscillator, although there is a larger approximation error. Note that the dynamics shown in **B-C** and **F-G** correspond to a mixture of Gaussians low-rank network, described by the statistics of the fifty populations, and are not the dynamics of a particular realization of a finite-size $N \times N$ connectivity matrix.

2.7 Discussion

In this manuscript, we have examined the dynamics in Gaussian-mixture low-rank recurrent neural networks, a class of models in which the connectivity is defined by a low-rank matrix, with connectivity patterns consisting of several populations with distinct Gaussian statistics. In these networks, the collective dynamics can be described by $R + N_{in}$ collective variables, where R is the rank of the connectivity matrix and N_{in} the dimensionality of the input patterns. These collective variables form a dynamical system, the evolution of which is determined by the connectivity statistics of the populations forming the network. The rank of the network, and the population structure therefore play complementary roles: the rank of the network sets the internal dimensionality of the dynamics and defines the corresponding collective variables, while individual populations shape the dynamics of these collective variables, but do not contribute new ones. We specifically showed that, in the limit of a large number of populations, this class of network displays a universal approximation property, and can therefore implement a large range of dynamical systems. Having a small number of populations instead imposes constraints and limits the achievable range of dynamics.

We have focused here on a specific family of distributions for the connectivity patterns,

mixtures of multi-variate Gaussians. This choice was motivated by several considerations. First, this family of distributions can be used to approximate any multi-variate distribution for the pattern loadings. Second, this family of distributions leads to a particularly simple form of dynamics for the collective variables, where the time-evolution is formulated in terms of a simple effective circuit (Eq. 2.24). Remarkably, in this description of the dynamics, which is exact and non-linear, the collective variables appear to interact linearly through effective couplings and effective inputs, that fully encapsulate the non-linearities. This allows for a particularly transparent interpretation of dynamics in terms of gain modulation. Several of our results are however independent of the specific assumption for the type of distribution; this is in particular the case for the influence of symmetry in the connectivity on the dynamics. When a large number of populations is needed to approximate the connectivity structure, other parametric distributions may be more suitable, and the interpretation in terms of discrete populations may not be appropriate.

Low-rank networks with arbitrary pattern distributions form a rich and versatile framework that encompasses a number of previously studied types of recurrent neural networks. As shown in the last part of the results, Hopfield networks storing $R \ll N$ patterns can be seen as a particular limit of Gaussian-mixture low-rank networks, in which pattern loadings are binary and exhibit a specific type of symmetry. The Neural Engineering Framework (Paulin, 2004) and the Manifold Embedding approach (Pollock and Jazayeri, 2020) provide algorithms that implement specific low-dimensional dynamics by controlling the structure of fixed points and Jacobians using linear-regression methods. These approaches generate recurrent networks with low-rank connectivity, in which the pattern loadings are however not a priori restricted to belong to a specific type of distribution. Approximating the obtained distributions by Gaussian mixtures might provide additionnal control of the generated dynamics.

Our framework is also closely related to Echo-state (Jaeger, 2001) and FORCE networks (Sussillo and Abbott, 2009), which rely on randomly connected recurrent networks controlled by feedback loops. Each feedback loop is mathematically equivalent to adding a unit-rank component to the connectivity matrix. Echo-state and FORCE networks therefore correspond to low-rank networks with an additionnal full-rank, random term in the connectivity (Mastrogiuseppe and Ostojic, 2018, 2019). Because the feedback loops are trained to produce specific outputs, the low-rank part of the connectivity is typically correlated to the random connectivity term (but see Mastrogiuseppe and Ostojic (2019)). Such correlations increase the dimensionality and the range of the dynamics (Schuessler et al., 2020a; Logiaco et al., 2019), although the low-rank connectivity structure and the number of populations still generate strong constraints. For instance, for rank-one networks with a random term in the connectivity, but consisting of a single population, the fixed points are restricted to lie on a one-dimensional, but non-linear manifold, and typically at most two non-trivial stable fixed points can be generated (Schuessler et al., 2020a). More generally, random components in the connectivity can strongly influence learning dynamics during training (Schuessler et al., 2020b).

Gaussian-mixture low-rank networks, the Neural Engineering Framework, and Echo-state networks all exhibit universal approximation properties (Eliasmith, 2005; Maass et al., 2002). It is however important to distinguish between several variants of this property. In our case, in analogy with the NEF, we started from an R-dimensional dynamical system fully specified by its flow function, and showed that Gaussian-mixture low-rank networks can approximate this flow function, provided a large number of populations is available and the flow function satisfied specific constraints. Echo-state and FORCE networks instead start by specifying a target readout, and universal approximation means that any such readout can be generated by training the feedback (Maass et al., 2007). This readout corresponds to a low-dimensional projection of a large dynamical system, and Echo-state networks are free to implement any dynamical system consistent with the specified output projection. This is a major distinction with our, and the NEF approach, where the overall dynamical

system is more tightly constrained.

In this work, we have examined only networks with fixed inputs. Varying the inputs instead modifies the low-dimensional dynamics, an effect that can be understood through modulations of effective couplings that govern the interactions between collective variables. In a companion paper (Dubreuil et al., 2020), we have used Gaussian-mixture low-rank RNNs to reverse-engineer networks trained on a range of neuroscience tasks, and found that gain modulation through input control underlies complex computations, such as flexible input-output mappings (Fusi et al., 2016). Varying inputs while keeping connectivity fixed therefore has the potential of implementing a large range of dynamical systems and computations (Pollock and Jazayeri, 2020), but the full capacity of this mechanism still remains to be understood.

Acknowledgements

The project was supported by the Ecole de Neurosciences de Paris, the ANR project MORSE (ANR-16-CE37-0016), the CRCNS project PIND, the program "Ecoles Universitaires de Recherche" launched by the French Government and implemented by the ANR, with the reference ANR-17-EURE-0017. There are no competing interests. We thank Mehrdad Jazayeri and Eli Pollock for discussions.

Code availability

Code and trained models will be made available upon publication.

2.7.1 Appendix A: Dynamics in multi-population networks

In this appendix, we derive the equation for the dynamics of a multi-population low-rank network, Eq. (2.17). We consider a low-rank network that consists of P populations, where each population is defined by different statistics of the probability distribution $P_{(p)}\left(\underline{m}, \underline{n}, \underline{I}\right)$. We assume that the external input is constant in time and uncorrelated with the left connectivity patterns. Each neuron in the network is assigned to a population according to the probability α_p. In the following, we set the statistics of each population to be drawn from a multivariate Gaussian with mean vector $\boldsymbol{a^{(p)}}$, as defined in Eq. (2.5), and covariance matrix $\Sigma^{(p)}$ (Eq. 2.6).

The recurrent dynamics in a low-rank network are determined by Eq. (2.16): it consists of a sum over the N units in the network. In the limit of large networks with defined statistics, by means of the law of large numbers, this sum over N i.i.d. elements corresponds to the empirical average over the distribution of its elements. Therefore, we can replace the sum over network units for $i = 1, \ldots, N$ of loadings $\left\{n_i^{(r)}\right\}, \left\{m_i^{(r)}\right\}$ and I_i, by an integral over their probability distribution $P\left(\underline{m}, \underline{n}, I\right)$. Using this probability distribution, the recurrent dynamics in Eq. (2.16) can be expressed as

$$\kappa_r^{rec} = \sum_{p=1}^{P} \alpha_p \int d\underline{m}\, d\underline{n}\, dI\, P_{(p)}\left(\underline{m}, \underline{n}, I\right) n_r^{(p)} \phi\left(I^{(p)}\kappa_I + \sum_{l=1}^{R} m_l^{(p)}\kappa_l\right). \tag{2.65}$$

Note that we refer to the input loadings I as a single Gaussian variable, instead of a set of Gaussian variables $\underline{I}$, because, since the input is constant in time, there is only one input pattern. We then separate the contribution of the mean a_{n_r} and the fluctuations of n_r around its mean into two different terms:

$$\kappa_r^{rec} = \sum_{p=1}^{P} \alpha_p \int dI\, d\underline{m}\, P_{(p)}\left(\underline{m}, I\right) a_{n_r}^{(p)} \phi\left(I^{(p)}\,\kappa_I + \sum_{l=1}^{R} m_l^{(p)}\kappa_l\right) \tag{2.66}$$

$$+ \sum_{p=1}^{P} \alpha_p \int dn_r\, dI\, d\underline{m}\, P_{(p)}\left(\underline{m}, n_r, I\right)\left(n_r^{(p)} - a_{n_r}^{(p)}\right) \phi\left(I^{(p)}\,\kappa_I + \sum_{l=1}^{R} m_l^{(p)}\kappa_l\right). \tag{2.67}$$

Using Stein's lemma in the second term, and making use of the fact that the sum of Gaussian variables is itself a Gaussian variable, we can express the dynamics as

$$\kappa_r^{rec} = \sum_{p=1}^{P} \alpha_p a_{n_r}^{(p)} \int \mathcal{D}x\, \phi\left(a_I^{(p)}\kappa_I + \sum_{s=1}^{R} a_{m_s}^{(p)}\kappa_s + x\sqrt{\sigma_{I^2}^{(p)}\kappa_I^2 + \sum_{s'=1}^{R} \sigma_{m_{s'}^2}^{(p)}\kappa_{s'}^2}\right) \tag{2.68}$$

$$+ \sum_{p=1}^{P} \alpha_p \left(\sigma_{n_r I}^{(p)}\kappa_I + \sum_{l=1}^{R} \sigma_{n_r m_l}^{(p)}\kappa_l\right) \int \mathcal{D}x\, \phi'\left(a_I^{(p)}\kappa_I + \sum_{l=1}^{R} a_{m_l}^{(p)}\kappa_l + x\sqrt{\sigma_{I^2}^{(p)}\kappa_I^2 + \sum_{l'=1}^{R} \sigma_{m_{l'}^2}^{(p)}\kappa_{l'}^2}\right) \tag{2.69}$$

where $\mathcal{D}x = dx\,(2\pi)^{-\frac{1}{2}}\, e^{-\frac{x^2}{2}}$. Finally, using the Gaussian integral notation in Eq. (2.21), we retrieve Eq. (2.18).

2.7.2 Appendix B: Universal approximation of low-dimensional dynamics

The universal approximation theorem for artificial neural networks (Hornik et al., 1989; Funahashi, 1989; Cybenko, 1989) states that any piecewise-continuous bounded function

$G(\boldsymbol{x})$, where $\boldsymbol{x}$ is a d-dimensional vector, can be approximated to arbitrary precision by a finite linear combination of non-linear units having the same transfer function but different gain and thresholds. More precisely, it is possible to build an approximation $\hat{G}(\boldsymbol{x})$ of $G(\boldsymbol{x})$

$$\hat{G}(\boldsymbol{x}) = \sum_{i=1}^{N} \boldsymbol{v}_i \phi\left(\boldsymbol{w}_i^T \boldsymbol{x} + b_i\right), \tag{2.70}$$

with finite integer N, and real values for $\boldsymbol{v}_i \in \mathcal{R}^{d'}$, $\boldsymbol{w}_i \in \mathcal{R}^d$ and $b_i \in \mathcal{R}$, so that $\left|G(x) - \hat{G}(x)\right| < \epsilon$, for any $\epsilon > 0$, given that the activation function $\phi(x)$ is a piecewise-continuous non-constant bounded function (Leshno et al., 1993).

There is a direct mapping between the second term of Eq. (2.70) and the recurrent dynamics of a low-rank RNNs. The recurrent dynamics in Eq. (2.16) can be directly mapped to Eq. (2.70): the variables $\frac{1}{N}\boldsymbol{n}_i$ correspond to $\boldsymbol{v}_i$, $\boldsymbol{m}_i$ to $\boldsymbol{w}_i$, and $\kappa_I I_i$ to b_i. This implies that the recurrent dynamics can approximate any flow function within a finite domain.

The dynamics of low-rank networks with multiple Gaussian populations can also be mapped to the universal approximation theorem. The mean term contribution to the dynamics in Eq. (2.17) reads

$$\sum_{p=1}^{P} \alpha_p \boldsymbol{a_n}^{(p)} \left\langle \phi\left(\boldsymbol{a_m}^T \boldsymbol{\kappa} + a_I^{(p)}, \sigma_{I^2}^{(p)} + \boldsymbol{\kappa}^T \sigma_{m^2}^{(p)} \boldsymbol{\kappa}\right)\right\rangle, \tag{2.71}$$

so that $\alpha_p \boldsymbol{a_n}^{(p)}$ maps to $\boldsymbol{v_i}$, $\boldsymbol{a_m}^{(p)}$ maps to $\boldsymbol{w}_i$ and $a_I^{(p)}$ is mapped to the bias term b_i. The transfer function is however different. In Eq. (2.70), the non-linear function used is $\phi(x)$, while in Eq. (2.71), the non-linear function used is $\langle \phi(x, \Delta(x))\rangle$. Both functions are non-linear and non-polynomial, so that the theorem applies. The contribution given by the disorder in the population loadings, $\boldsymbol{\sigma}_{m^2}^{(p)}$ and $\sigma_{I^2}^{(p)}$ are not required for the universal approximation. However, quadratic terms like the one introduced by the variance of loadings improve the approximation in terms of expressibility and efficiency (Fan et al., 2020). Overall, this means that a low-rank network with a finite number of populations can approximate any dynamical system within a bounded domain.

2.7.3 Appendix C: Linear stability matrix at fixed points in networks with single population

The linear dynamics of small perturbations around the fixed point $\boldsymbol{\kappa_0}$ (defined in Eqs. 2.29) read

$$\tau \frac{d\boldsymbol{\kappa}}{dt} = -\boldsymbol{\kappa} + \left[\nabla\left(\left\langle \phi'\left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa}\right)\right\rangle \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa}\right)\right]_{\boldsymbol{\kappa}=\boldsymbol{\kappa_0}} \boldsymbol{\kappa}, \tag{2.72}$$

where ∇ is the vector differential operator. We apply the property $\nabla\left(f(\boldsymbol{\kappa}) A \boldsymbol{\kappa}\right) = f(\boldsymbol{\kappa}) A + A\boldsymbol{\kappa}\left(\nabla f(\boldsymbol{\kappa})\right)^T$, based on the chain rule, to obtain:

$$\tau \frac{d\boldsymbol{\kappa}}{dt} = -\boldsymbol{\kappa} + \left[\left\langle \phi'\left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa}\right)\right\rangle \boldsymbol{\sigma_{mn}} + \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa} \left\langle \nabla \phi'\left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa}\right)\right\rangle^T\right]_{\boldsymbol{\kappa}=\boldsymbol{\kappa_0}} \boldsymbol{\kappa}. \tag{2.73}$$

We then calculate the gradient of the gain factor. To do so, we first write explicitly the Gaussian integral

$$\left\langle \nabla \phi'\left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa}\right)\right\rangle = \int \mathcal{D}x \nabla \phi'\left(\sqrt{\boldsymbol{\kappa}^T \boldsymbol{\kappa}} x\right), \tag{2.74}$$

where $\mathcal{D}x$ is the differential element of a normally distributed variable. Applying the chain rule

$$\left\langle \nabla \phi'\left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa}\right)\right\rangle = \int \mathcal{D}x \phi''\left(\sqrt{\boldsymbol{\kappa}^T \boldsymbol{\kappa}} x\right) \nabla\left(x\sqrt{\boldsymbol{\kappa}^T \boldsymbol{\kappa}}\right) = \int \mathcal{D}x \phi''\left(\sqrt{\boldsymbol{\kappa}^T \boldsymbol{\kappa}} x\right) x \frac{\boldsymbol{\kappa}}{\sqrt{\boldsymbol{\kappa}^T \boldsymbol{\kappa}}}. \tag{2.75}$$

Using Stein's lemma, the gradient of the gain factor reads:

$$\left\langle \nabla\phi'\left(0, \boldsymbol{\kappa}^T\boldsymbol{\kappa}\right)\right\rangle = \int \mathcal{D}x\phi'''\left(\sqrt{\boldsymbol{\kappa}^T\boldsymbol{\kappa}}x\right)\boldsymbol{\kappa} = \left\langle \phi'''\left(0, \boldsymbol{\kappa}^T\boldsymbol{\kappa}\right)\right\rangle \boldsymbol{\kappa}. \tag{2.76}$$

Finally, introducing Eq. (2.76) into Eq. (2.73), and using the fact that $\boldsymbol{\sigma_{mn}}\boldsymbol{\kappa_0} = \lambda_r\boldsymbol{\kappa_0}$, the dynamics of small perturbation around the fixed point read

$$\tau\frac{d\boldsymbol{\kappa}}{dt} = \left[-\boldsymbol{I} + \left\langle \phi'\left(0, \boldsymbol{\kappa_0}^T\boldsymbol{\kappa_0}\right)\right\rangle \boldsymbol{\sigma_{mn}} + \left\langle \phi'''\left(0, \boldsymbol{\kappa_0}^T\boldsymbol{\kappa_0}\right)\right\rangle \boldsymbol{\sigma_{mn}}\boldsymbol{\kappa_0}\boldsymbol{\kappa_0}^T\right]\boldsymbol{\kappa}, \tag{2.77}$$

which leads to the linear stability matrix given by Eq. (2.31).

It is important to analyze the behavior of the function $\langle\phi'''(0,\Delta)\rangle$ to assess the stability. In the limit $\Delta = 0$, the Gaussian integral reduces to the evaluation of the function at zero. For a transfer function $\phi(x) = \tanh(x)$ we obtain:

$$\lim_{\Delta\to 0}\langle\phi'''(0,\Delta)\rangle = \phi'''(0) = -2. \tag{2.78}$$

In the limit of infinite Δ, the Gaussian integral can be expressed as :

$$\lim_{\Delta\to\infty}\langle\phi'''(0,\Delta)\rangle = \int_{-\infty}^{+\infty} dx\,\phi'''(x) = 0. \tag{2.79}$$

Furthermore, it can be shown that it is a monotonically increasing function of Δ, so that its value for any Δ is negative and bounded between -2 and 0.

Summary of Chapter 3

Animals can flexibly control the timing and speed of a given action. Neural recordings in behaving monkeys have shown that flexible timing relies on neural activity that is temporally stretched when the same action is executed at different time intervals (Wang et al., 2018), so that at the level of the neural population, neural activity evolves at different speeds along an identical low-dimensional invariant manifold. In this work, we used networks of recurrently connected units to investigate the mechanisms of neural dynamics that underlie such flexible temporal computations.

We started by training low-rank recurrent neural networks to solve timing tasks, and reverse-engineered them to identify candidate dynamical mechanisms underlying the generated neural trajectories. In a second step, we reproduced these isolated dynamical mechanism in reduced low-rank network models. Finally, we tested the computational role of those mechanisms by implementing the same tasks using the reduced models. This approach allows us to discover, characterize and test novel network mechanisms for performing temporal computations.

We found that recurrent networks perform temporal computations by generating slow manifolds that correspond to continuous attractive regions of neural states with low-speed dynamics. Such manifolds generate slow transient trajectories, store continuous estimates of temporal intervals in working memory and produce temporally scaled output signals. We show that low-rank network connectivities with a simple, quasi-isotropic connectivity structure are sufficient to generate such slow manifolds. Deviations from a perfect isotropic connectivity structure robustly shape the dynamics along the slow manifold, while tonic inputs can modulate the speed along the manifolds. Altogether, we identified a set of novel dynamical mechanisms for temporal flexibility that rely on minimal connectivity structure and can implement a vast range of computations.

The work included in this Chapter was collaboratively supervised by S. Ostojic and M. Jazayeri. Corresponding manuscript in preparation.

Temporal computations through dynamics on neural manifolds 3

3.1 Introduction

Temporal flexibility is a fundamental aspect of animal behavior. A given motor action can be executed at widely varying speeds based on the internal state of the animal and the environmental demands (Safaie et al., 2020), such as urgency (Drugowitsch et al., 2012; Thura and Cisek, 2016), attention (Nobre and Van Ede, 2018) internal motivation and vigor (Manohar et al., 2015) or timescale of relevant information and rewards (Kacelnik and Brunner, 2002). To this effect, it is necessary for the brain to process the temporal structure of external stimuli by estimating the duration of presented stimuli, keep track of time, and understand the structure of sequences and the periodicity of rhythmic events.

Among the cognitive computations performed by the brain, those involving time processing are idiosyncratic in several ways (Buonomano and Maass, 2009). The nervous system generates itself time-dependent activity that is constrained by numerous biophysical constraints; from the time resolution of action potentials, to the diffusion of neuromodulators or the time delays between connected regions. In spite of this, the brain can still produce behavior adapted to a very wide range of timescales (Mauk and Buonomano, 2004). Another specific property of temporal computations is that time is an analog variable. Unlike decision making where discrete actions must be taken or cognitive tasks involving categorizing stimuli into discrete categories, the nature of time is continuous. It remains an open question whether continuous quantities are represented and transformed in the brain as discrete or continuous processes, or a combination of both (Goldman et al., 2003; Ma et al., 2014; Panichello et al., 2019; Tee and Taylor, 2018). For such reasons, sensory-motor computations explicitly involving time have remained relatively understudied by the neuroscience community.

Recent studies have investigated the neural substrate of speed control for motor responses. A key finding is that the profile of neural activity is temporally adjusted, stretched or expanded, to flexibly generate adaptive behavioral responses. This property is present in different experimental paradigms such as in speed-accuracy trade-off studies (Hanks et al., 2014), sensory anticipation tasks (Kilavik et al., 2014) or flexible interval timing tasks (Wang et al., 2018; Remington et al., 2018a; Sohn et al., 2019). The possible mechanisms at the level of neuronal networks that control the speed of such neural responses remain to be fully elucidated.

An emerging approach has proposed to focus on the general motifs in the collective activity of neuronal activity that drive goal-directed behavior. The computation-through-

dynamics framework studies such joint activity of an ensemble of neurons by analyzing the trajectories drawn through time in neural space, a high-dimensional space where each dimension corresponds to the firing rate of one neuron (Buonomano and Maass, 2009; Remington et al., 2018b; Vyas et al., 2020). At each moment, the activity of the neural population corresponds to one point in this high-dimensional space. As time evolves, the neural state moves in high-dimensional space delineating a curve or trajectory. Interestingly, trajectories of cortical activity while performing cognitive tasks have been found to span a low number of dimensions and are constrained to smooth regions of neural space, called neural manifolds (Yu et al., 2009; Kaufman et al., 2014; Elsayed et al., 2016; Gallego et al., 2017; Wang et al., 2018). Applying this framework to a task that requires to flexibly produce a timed motor response, Wang et al. (2018) found that the dimensions of neural manifolds can be classified into two different categories based on their timing properties. Along some dimensions, the manifolds show temporal scaling: trajectories evolve along the same path, but they do so at different speed. These dimensions define the *temporal scaling subspace* of the neural manifold. Simultaneously, neural activity along a different set of dimensions controls the speed at which trajectories evolve. Subsequent studies (Remington et al., 2018a; Sohn et al., 2019) extended this finding to flexible timing tasks with additional cognitive requirements.

In this Chapter, we explore the theoretical basis of speed-control on neural manifolds generated by the recurrent connectivity of neural networks. We first address the question of how neural manifolds can be generated by the recurrent connectivity of cortical networks and what structures allow to control their speed. Secondly, we describe how such manifolds are used to implement specific temporal computations, and finally we discuss the broader functional role of task-related manifolds for generalization to unseen stimuli and learning novel but related timing tasks.

To this end, we use recurrent neural networks as an *in silico* model of local cortical networks (Fig. 3.1 A). We first trained recurrent neural networks to solve flexible timing tasks. Once the trained networks have learned how to solve the tasks, we reverse-engineer them (Sussillo and Barak, 2013; Wang et al., 2018): we identify candidate core mechanisms used by the networks to solve the tasks. We then examine those mechanisms by reproducing them in isolation in simplified network models. Finally, we test these computational mechanisms by implementing the tasks using the reduced network models. It is then possible to close the loop based on the findings and design new experimental tasks to test new hypothesis.

We focus all along the study on networks with random low-rank connectivity (Mastrogiuseppe and Ostojic, 2018; Schuessler et al., 2020b; Beiran et al., 2020; Dubreuil et al., 2020). This specific connectivity structure (Fig. 3.1 B) constrains the network to generate low-dimensional dynamics, which facilitates the identification of dynamical mechanisms in trained network and link them in reduced network models to the relationship between connectivity patterns.

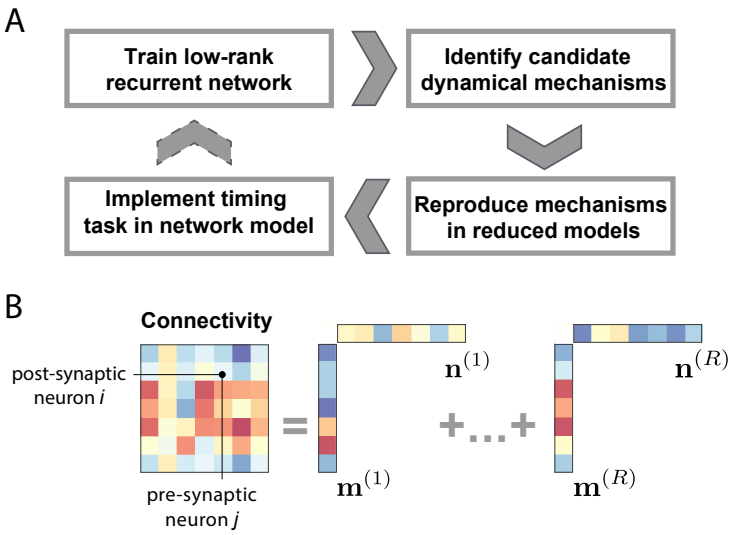

FIGURE 3.1: **Research strategy and recurrent connectivity**. **A** Approach for studying temporal computations in recurrent neural networks. First, recurrent networks with low-rank connectivity are *trained* to solve timing tasks. Secondly, trained networks are *reverse-engineered* to identify the candidate dynamical mechanisms set up to solve the tasks. Then, the identified mechanisms are *reproduced* in isolation by means of reduced network models. Finally, the considered tasks are *implemented* by means of the reduced network models. We can then close the loop by designing novel timing tasks based on the dynamical mechanisms. **B** Trained networks and reduced network models are constrained to have low-rank connectivity. A rank-R connectivity matrix (left) is decomposed using Singular Value Decomposition (SVD) into the sum of R rank-one terms (right), where each term is defined by the outer product of two connectivity patterns $\mathbf{m}^{(r)}$ and $\mathbf{n}^{(r)}$, for $r = 1, \ldots, R$. Low-rank connectivities constrain the dimensionality of the activity of large neural networks to be low, which facilitates the study of the emergent dynamical landscape and its link with the connectivity patterns.

3.2 Flexible timing tasks

3.2.1 Task epochs

We focus on a set of three tasks involving flexible computations with time, that we present here grouped by their cognitive components. These tasks are designed following the principle of compositionality: each task partly builds on previous tasks, and adds a new module that require complementary computations. This approach allows us to understand the mechanisms required to solve each aspect of the task, and how these mechanisms are combined with each other.

Production All tasks require the animal to execute a motor action after some precise time interval following a short input, the 'Set' stimulus. The required time interval that must be produced changes on a trial-by-trial basis and is indicated in various ways in different tasks. We refer to the part of the trial between the input pulse 'Set' and the motor action as the production epoch, and to the self-initiated motor action as the 'Go' event. The first presented task, Cue-Set-Go, focuses on understanding only the flexible production of intervals (Fig. 3.2 A, based on Wang et al. (2018)). The duration of the produced interval t_p is determined by a cue presented at the beginning of the trial. The agent must have

learned over training the association between different cues and different produced intervals to solve this task.

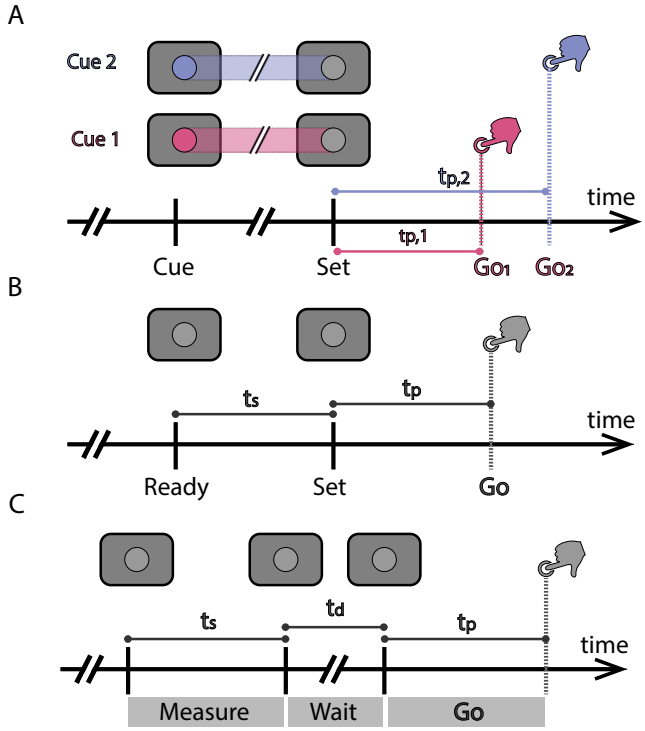

Figure 3.2: **Flexible timing tasks**. **A Cue-Set-Go.** A contextual input at the beginning of the trial, the Cue, determines the length of the interval to be produced, t_p (e.g., blue and red cues, associated with a long and a short interval, respectively). A second input, Set, indicates the beginning of timing. A motor action is required at time t_p after Set, the 'Go' action. This task focuses on the flexible production of a time interval **B Ready-Set-Go.** Two input pulses, Ready and Set, define a time interval t_s. The motor action is expected at time $t_p = t_s$ after Set. This task requires estimating a time interval in addition to motor timing. **C Measure-Wait-Go.** Two input pulses at the beginning of the trial determine the interval to be reproduced, t_s. After the estimation epoch, there is a delay of random duration t_d. A third pulse indicates the beginning of the production epoch. This task builds on Ready-Set-Go and adds a working memory component: the sensory estimate must be stored during the random delay.

Estimation of a temporal interval A second task, the Ready-Set-Go task, demands to estimate the time interval elapsed between two brief stimuli immediately before the production epoch (Fig. 3.2 B, first used in Jazayeri and Shadlen (2010)). The trials start with two input pulses, Ready and Set, separated by a time interval t_s, denoted as the sample interval. The animal is asked to generate a motor response a time $t_p = t_s$ after the second pulse, 'Set'. Therefore, the production epoch of this task demands to produce a rightly timed motor action, as in the Cue-Set-Go. However, the produced interval is here indicated by a previous time interval that needs to be estimated based on the timing of sensory inputs. We refer to the part of the trial where the animal is exposed to the sampled interval as the estimation epoch. The 'Set' pulse in this task indicates both the end of the estimation epoch and the beginning of production. The sample interval t_s is drawn

randomly at every trial from a given distribution, that we call prior distribution. Several variants of this task have been considered in experimental studies. The animal can combine this prior information with the noisy estimation of the sample interval, to optimize the performance. In one experimental study, Sohn et al. (2019) studied the Ready-Set-Go task alternating between two different prior distributions. Remington et al. (2018a) extended the task so that the produced interval t_p is a linear function of the estimated interval, $t_p = gt_s$, with different gain parameters g. In these two studies, the information about the gain or the prior distribution was cued at the beginning of the trial. Finally, Egger et al. (2019) modified the Ready-Set-Go paradigm by including more than one repetition of the sampled interval within each trial. All these tasks require to time a motor action based on an uncertain sensory estimation, due to the noise inherent to the any perceptual task.

Working memory for the estimated interval The set of timing tasks is extended by a novel task, Measure-Wait-Go, which adds a random delay between the estimation epoch and the production epoch (Fig. 3.2 C). This random delay between the estimation and production epochs, not present in the Ready-Set-Go tasks, obliges the animal to hold in memory the estimate of the sampled interval. While the Ready-Set-Go task can be thought of as the production of a third beat after the two first beats of a rhythm, the delay in this task breaks up the rhythm. This task is composed of three different short stimuli (pulses), instead of two. The first two stimuli indicate the beginning and end of the estimation epoch, while the third stimulus marks the beginning of the produced interval.

3.2.2 Task implementation in recurrent networks

We study the neural mechanisms on which recurrent neural networks rely to solve this set of temporal tasks. For that purpose, it is necessary to model specifically the output that we require from the network to correctly solve the task and the inputs that these local networks receive. The mechanisms that recurrent networks use to solve the task will most likely depend on the particular design of inputs and outputs of the local network.

We restrict the output of the network to the time points around the production epoch, since we assume that downstream areas use this time window for the initiation of a motor action. We require a linear combination of the network activity –the readout activity– to ramp linearly from an initial state to a fixed final value, the threshold (Fig. 3.3 A). Our choice is motivated by behavioral models that use accumulator variables that integrate some external variable (e.g., time) until reaching a threshold. A ramping variable towards a fixed threshold can describe the distribution of reaction times in voluntary movement initiation (Carpenter and Williams, 1995; Hanes and Schall, 1996). Additionally, the activity of a subset of cortical neurons has been found to evolve to a well-defined motor initiation state right before executing the motor action (Romo and Schultz, 1987; Roitman and Shadlen, 2002; Maimon and Assad, 2006). More recent studies have found that a wider group of cortical neurons, although showing more complex temporal profiles, also reach a fixed action-triggering state before movement initiation (Churchland et al., 2006; Wang et al., 2018).

In order to flexibly produce different time intervals, a ramping signal could either adjust the bias (the distance to threshold at the beginning of the trial) or the slope of the ramp, in other words, the speed at which the signal evolves. In the last years, a series of studies have found that the speed at which neural activity evolves in flexible timing tasks is modulated based on the length of the produced interval (Wang et al., 2018; Remington et al., 2018a; Sohn et al., 2019). Based on this evidence, we use an output signal that grows towards threshold at different speeds, adjusting the slope of the ramp for different intervals.

An important assumption about the inputs in these tasks is that we only consider low dimensional external inputs with simple dynamics, namely, tonic inputs or brief pulses. More complex inputs could simplify the task without requiring any explicit temporal computation

by the recurrent network. In the Cue-Set-Go task, we assume that the contextual information given by the cue is fed to the network as a tonic input, present during the whole trial. The amplitude of this input, that changes from trial to trial, is associated with different produced intervals (Fig. 3.3 B, left). This choice is consistent with the finding that thalamic inputs to medial frontal cortex show low temporal complexity along the total duration of the trial and its strength is modulated monotonically by the contextual information (Wang et al., 2018).

For all the other external stimuli, indicating the beginning of the estimation/production epoch, or the end of the estimation epoch, we assume that the inputs are short pulses that can immediately change the network state. Mathematically, we describe these brief stimuli in time as proportional to a Dirac delta function (Fig. 3.3 B). They do not modify the dynamical landscape generated by the network, but trigger instantaneous changes in the neural trajectories. It remains an open question whether the inputs to the cortical network responsible for the timing computations are received along distinct or the same spatial patterns. On one hand, stimuli corresponding to different events are generally shown in different locations of the visual field, and have different shapes and colors. On the other hand, some upstream area between the sensory cortices and the cortical network performing the temporal computations might already integrate the different visual stimuli and generate a stimulus-invariant response. From a computational perspective, restricting all short external inputs to the same spatial pattern imposes a more restrictive constraint, since the network can only tell stimuli apart based on the temporal order of presentation. In this study, we consider both possibilities for the input patterns of different pulses.

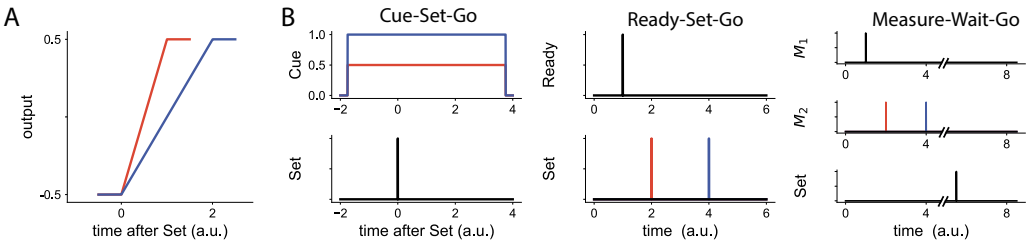

Figure 3.3: **Output and inputs to the recurrent network in flexible timing tasks**. **A Output.** Example of two different readout activity, producing a short interval (red) and a long interval (blue). The readout activity ramps from an initial state, -0.5, towards a threshold value, 0.5. The motor action is performed when the readout signal reaches threshold. Thus, the speed/slope of the ramp controls the timing. **B Inputs**. Inputs fed to the recurrent network in the three flexible tasks presented in Fig. 3.2. Every line corresponds to the temporal profile of an input along a different spatial pattern, and different colors correspond to trials with different produced interval t_p. In the Cue-Set-Go task (left), the amplitude of the tonic input provides the information about the interval to be produced after Set. In the Ready-Set-Go task, the produced interval must equal the sampled time between Ready and Set. In the Measure-Wait-Go task, each trial is composed of three pulses: the first two pulses bound the sampled interval t_s, while the third pulse, Set, which appears after a random delay, initializes production.

3.3 Analyses of trained recurrent networks

3.3.1 Strategy

We trained different recurrent neural networks to perform each of the flexible timing tasks described in Section 3.2.2, with the particularity that we constrained the rank of the network

connectivity matrix a priori. Once a network is trained successfully, we reverse-engineered it, that is, we studied the network and its dynamics to understand how they solve the task (see Dubreuil et al. (2020) for a similar approach on other cognitive tasks). The goal of training recurrent networks with minimal rank is two-fold: it allows to identify the minimal number of collective variables required to perform the task, and it also simplifies the analysis of the network dynamics, since we can apply the theory of low-rank networks to study the network dynamics.

The dynamics of rank-R recurrent networks read

$$\tau \frac{d\mathbf{x}}{dt} = -\mathbf{x} + \sum_{r=1}^{R} \mathbf{m}^{(\mathbf{r})} \mathbf{n}^{(\mathbf{r})^T} \phi(\mathbf{x}) + \sum_{s=1}^{s} u_s(t) \mathbf{I}^{(\mathbf{s})} \tag{3.1}$$

where $\mathbf{x}(t)$ is an N-dimensional vector representing the input received by each of the N network units, τ is the membrane time constant, and $\mathbf{m}^{(\mathbf{r})}$ and $\mathbf{n}^{(\mathbf{r})}$ are the left and right connectivity patterns that determine the rank-R connectivity matrix. In analogy to factor analysis, we refer to the entries of the connectivity patterns, $m_i^{(r)}$ and $n_i^{(r)}$, as pattern *loadings*. The vectors $\mathbf{I}^{(\mathbf{s})}$ correspond to the different spatial patterns of the external inputs, and $u_s(t)$ to their temporal profile. The firing rate of each neuron is obtained by applying a non-linear function ϕ to the input. In this study, we use a sigmoidal transfer function $\phi(x) = \tanh x$. The readout or output signal of the recurrent network is a linear combination of the firing rate of single units, defined as

$$z(t) = \sum_{i=1}^{N} w_i \phi(x_i(t)). \tag{3.2}$$

The readout can be interpreted as a one-dimensional projection of the firing rates of all network units along vector $\mathbf{w}$.

For each task, we used the temporal profile of inputs $u_s(t)$ shown in Fig. 3.3 B. The goal of training was to generate the readout $z(t)$ shown in Fig. 3.2 A: a ramp with the required slope during the production epoch (see Methods, Section 3.7.2). We first fixed the rank and trained the following parameters using backpropagation through time: the left and right connectivity patterns $\mathbf{m}^{(\mathbf{r})}$ and $\mathbf{n}^{(\mathbf{r})}$, and the initial conditions $\mathbf{x}(t=0)$. Unless stated otherwise, the input and readout patterns are fixed at the beginning of training, and only the overall scale of the patterns is trained (see Methods, Section 3.7.1, for details about the training procedure)

The intrinsic dimensions of recurrent network dynamics is determined by the rank of the connectivity matrix plus the dimensionality of the external input. The state vector $\mathbf{x}(t)$ can be expressed in the basis spanned by the input and (right) connectivity patterns $\mathbf{m}^{(\mathbf{r})}$,

$$\mathbf{x} = \sum_{r=1}^{R} \kappa_r \mathbf{m}^{(\mathbf{r})} + \sum_{s=1}^{S} \kappa_{I_s} \mathbf{I}_{\perp}^{(\mathbf{s})}, \tag{3.3}$$

so that the whole dynamics can be described by a few collective variables κ_r (for $r = 1, \ldots R$) and κ_s, for $s = 1, \ldots S$. We assumed for simplicity that the input patterns $\mathbf{I}_{\perp}^{(\mathbf{s})}$ are orthogonal to the left connectivity patterns $\mathbf{m}^{(r)}$. The R collective variables span the so-called *recurrent space*: the dimensions of the dynamics spanned by the connectivity, while the variables κ_{I_s} correspond to the collective variables in the *input subspace* (Wang et al., 2018).

The dynamics of the low-rank network shown in Eq. (3.1), which correspond to an N-dimensional system of equations, can be rewritten in terms of the collective variables as the $R + S$-dimensional dynamical system:

$$\tau \frac{d\kappa_r}{dt} = -\kappa_r + \kappa_r^{rec} \tag{3.4}$$

$$\tau \frac{d\kappa_{I_s}}{dt} = -\kappa_{I_s} + u_s(t), \tag{3.5}$$

where

$$\kappa_r^{rec} = \frac{1}{N} \mathbf{n}^{(\mathbf{r})^T} \phi \left(\sum_{r=1}^{R} \kappa_r \mathbf{m}^{(\mathbf{r})} + \sum_{s=1}^{S} \kappa_{I_s} \mathbf{I}^{(\mathbf{s})} \right). \tag{3.6}$$

External inputs in the studied tasks have very simple dynamics, they are either delta pulses or constant in time. Therefore, the dynamics in the input subspace determined by the variables κ_{I_s} are also simple. For transient pulses, neural trajectories converge quickly (at the timescale of the membrane time constant) to the recurrent subspace. For that reason, we restrict the analysis of trained networks to the recurrent subspace, determined by collective variables κ_r, for $r = 1, \ldots, R$ (Eq 3.4). In the following, we refer to the value of all collective variables κ_r using the vector notation $\boldsymbol{\kappa}$.

The low-dimensional recurrent dynamics are visualized by means of two different tools: the speed at every point in the recurrent subspace and the streamlines. Similar to Sussillo and Barak (2013), we define speed as the scalar function

$$Q(\boldsymbol{\kappa}) = \sqrt{\sum_{r=1}^{R} \left(\frac{d\kappa_r}{dt} \right)^2}. \tag{3.7}$$

States $\boldsymbol{\kappa}^*$ in the recurrent subspace where the speed is zero, $Q(\boldsymbol{\kappa}^*) = 0$, correspond to fixed points of the dynamics, so that if a trajectory reaches that point, it remains at that state in the absence of inputs. In rank-one networks, speed is represented as a one-dimensional function of the collective variable κ. In rank-two networks, the speed is a function of two variables, κ_1 and κ_2, so that it can be displayed as a colormap, where the color at every point indicates the speed value Q.

The streamlines are a set of curves distributed across the recurrent subspace which correspond to trajectories with different initial conditions. They can be graphically represented in networks up to rank-three. Streamlines provide complementary information to the speed function. For instance, they indicate the stability of fixed points. If a fixed point is stable, all streamlines in its vicinity lead towards the fixed point, while if streamlines point away from it in one or more directions, the fixed point is unstable. Streamlines can also inform about other phenomena such as limit cycles (closed curves in which the speed is always different from zero) or orbits (trajectories starting and finishing in fixed points).

We refer to the collection of dynamical phenomena that can be produced by a given recurrent neural network as its *dynamical landscape*. This includes the fixed points of the network (location and stability), cycles, orbits and the way they are positioned with respect to each other in collective space. The dynamical landscape in networks with rank higher than two might also include other dynamical phenomena such as chaotic attractors. As the rank of a neural network increases, and therefore the dimensionality of the dynamics, it becomes more challenging to identify the full dynamical landscape of networks.

The first step in our approach to reverse-engineer trained networks is to inspect the dynamical landscape of the recurrent subspace, or the recurrent subspaces, if several tonic inputs are considered. Tonic inputs shift the recurrent subspace along a given direction in neural space, so that the dynamical landscape also changes. As a second step, we look at the regions of the dynamical landscape explored by the neural trajectories that solve the task, in order to understand the dynamical components used for temporal computations.

One observation common to the analysis of all trained networks, as we show in the next section, is that they generate low-dimensional and compact sets of states $\{\boldsymbol{\kappa}^*\}$ in the

recurrent subspace where the speed is very slow:

$$Q\left(\{\boldsymbol{\kappa}^*\}\right) < \epsilon, \tag{3.8}$$

where $\epsilon > 0$ is a threshold value, much smaller than the speed given by the membrane time constant τ^{-1}. We refer to these elements of the dynamical landscape as *slow manifolds* (see Methods, 3.7.3). The slow manifolds we found are smooth, often ring-shaped or spherically-shaped. One property of these slow manifolds is that they are attractive, in the sense that when the neural activity is initiated at a random state, the trajectories quickly reach the slow manifold and then evolve slowly on the manifold, until they reach a fixed point, if there is one within the manifold. Furthermore, we found that neural trajectories in all studied flexible timing tasks go along these internally-generated manifolds, to produce different computations such as interval estimation, storage and production. In the following section, we detail how trajectories evolve along these slow neural manifolds at different epochs of the tasks.

3.3.2 Trained networks on timing tasks

We found that the minimum rank necessary for flexibly generating a time interval (Cue-Set-Go task) or estimating an interval and reproducing it immediately after (Ready-Set-Go task) is two. Therefore, the recurrent dynamics during any given trial evolve on a plane in neural space, determined by the collective variables κ_1 and κ_2. For the Measure-Wait-Go task, where the neural dynamics must also store in working memory the temporal estimate, we found that the recurrent dynamics require the use of a third dimension, so that the minimal rank is three and there are three collective variables κ_1, κ_2 and κ_3.

Production We analyze the production epochby focusing on trained networks solving the Cue-Set-Go task (inputs and readout of trained network in Fig. 3.4 A-B). Inspecting the dynamical landscape generated by the network, we find that there are four non-trivial fixed points: two stable ones and two saddle points. These points are connected by a closed trajectory where dynamics are slow, the *slow manifold* (blue region, Fig. 3.4 E). The ramping output of the network is generated while the neural trajectories evolve along this slow manifold. The path of neural trajectories is described as follows: first, trajectories are initialized at a stable fixed point (left point on the manifold, Fig. 3.4 C). Once the 'Set' input is presented, the trajectories quickly move above a saddle point and unfold along the slow manifold towards a final fixed point. For different intervals, the neural trajectories overlap in the two-dimensional space of collective variables.

It is also possible to visualize the neural trajectories in the three-dimensional space (given by the two recurrent dimensions and the direction of the external cue input, Fig. 3.4 D). In that 3D space, the slow manifold corresponds to the surface of a cylinder, along which trajectories evolve in parallel. Different heights on this cylinder correspond to different speeds of the manifold, because the network is able to temporally adapt its readout. Indeed, the tonic cue modulates the speed along the manifold, without affecting its shape, as shown in Fig. 3.4 E. For different values of the cue, the shape of the slow manifold remains largely unchanged, while its speed is decreased.

Experimental recordings in the Cue-Set-Go task showed that neural trajectories corresponding to different produced intervals evolve along some scaling dimensions (Wang et al., 2018). In this subspace, trajectories overlap, but they evolve at different speeds. At the same time, there is an orthogonal subspace, where the initial conditions at the beginning of production set the speed along the scaling subspace. These results are consistent with the neural trajectories of trained rank-two networks (Fig. 3.4 D). From trial to trial, the initial conditions along the input dimension $\boldsymbol{I_{cue}}$ change, depending on the amplitude of the tonic input. Simultaneously, the amplitude of the tonic input controls the speed along

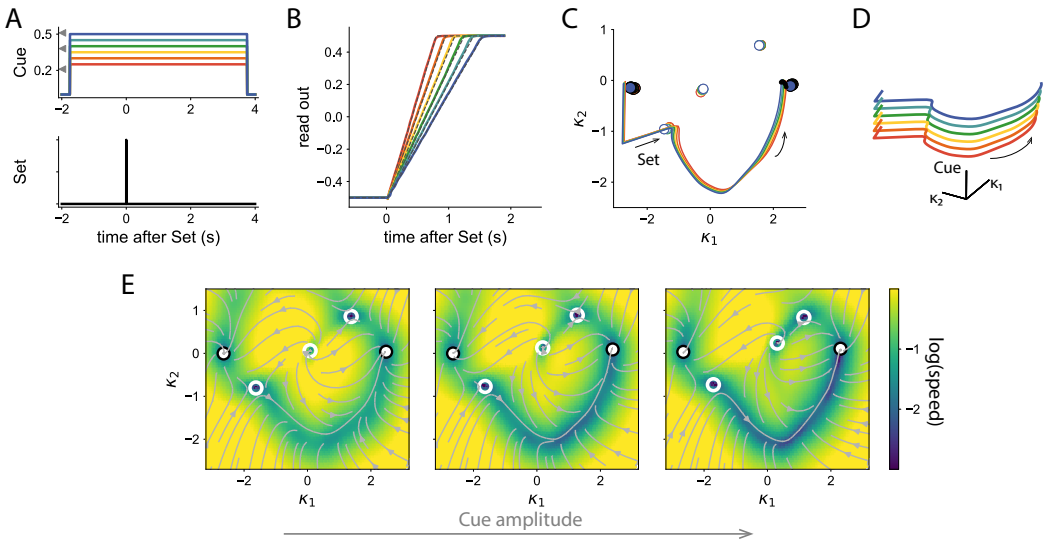

Figure 3.4: **Interval production in rank-two network trained on Cue-Set-Go task**. **A** Inputs received by the network to perform the task. We consider six different cue values, corresponding to six different time intervals to be produced, so that on each trial, the tonic input (Cue, top) takes one of the six shown values. **B** Network readout of the trained rank-two network for different cues. The dotted lines are the corresponding target function. **C** Neural trajectories for different Cues, in the two-dimensional recurrent space given by collective variables κ_1 and κ_2. The two collective variables are the projection of the state vector $\boldsymbol{x}$ onto the connectivity patterns $\mathbf{m}^{(1)}$ and $\mathbf{m}^{(2)}$. Colored dots: stable fixed points, white dots: saddle points. The trajectories start on the fixed point on the left. The Set input quickly moves the neural state beyond the saddle bottom in the bottom. Trajectories then evolve along the same neural manifold towards the stable fixed point on the right. **D** Neural trajectories in the three dimensional space given by the collective variables and the tonic input (vertical axis). The trajectories evolve in parallel at different levels of the speed. **E** Dynamics in collective space for three different amplitudes of the cue input (amplitude values indicated with triangles in panel **A**, top). The dynamics are shown by plotting the flow field (grey lines, indicating in which direction in neural space trajectories evolve if initiated at different states) and the speed at which they evolve, represented by the colormap (the log-speed of the dynamics is defined as $\log_{10} Q$). As the cue is increased, the speed along the bottom part of the manifold becomes slower (darker in the colorscale), while not changing its shape. This mechanism produces different time intervals. Parameters: $N = 150$, membrane time constant $\tau = 30$ms. Cue values shown in E: $0.2, 0.38, 0.5$ (triangles in panel **A**, top).

the manifold, which lies on an orthogonal plane spanned by patterns $\boldsymbol{m}^{(1)}$ and $\boldsymbol{m}^{(2)}$. This plane, which corresponds to the recurrent subspace, coincides in this task with the temporal scaling subspace. Trajectories projected on the temporal scaling subspace overlap with each other, but they unroll at different speeds. Overall, the trained network constitutes an in-silico model that reproduces the salient features of the data while giving access to the broader dynamical landscape, such as the generation of a slow manifold.

The mechanism we described for production in the Cue-Set-Go task is based on the tonic external input: the different states on the cue axis determine the speed. However, in the other two tasks (Ready-Set-Go and Measure-Wait-Go), there is no external input that can set the different initial states when the Set signal arrives. Therefore, the different initial conditions at the beginning of production must be internally generated by the network,

based on the history of the neural trajectories. We study this more in detail, by analyzing how networks estimate and store a temporal interval.

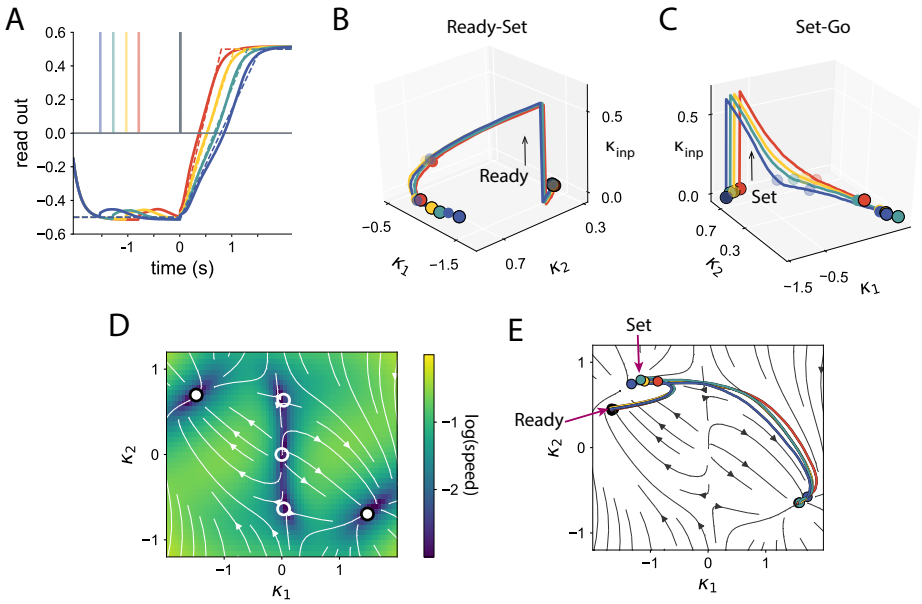

FIGURE 3.5: **Production and estimation in Ready-Set-Go task**. **A** Inputs received by the network to perform the task (shaded pulses), aligned to the 'Set' pulse. Each color corresponds to a different sampled interval. The curves are the readout signal of a trained rank-two network (dashed lines are the target signal). This rank-two network is able to produce a ramping signal with the duration of a previously estimated interval. **B** Neural trajectory during the estimation epoch, in the 3D space spanned by the recurrent dynamics (collective variables κ_1 and κ_2) and by the input vector, κ_{inp}. The first pulse elicits a slow trajectory, so that at the end of the estimation epoch, the neural state is different for different sampled intervals. **C** Neural trajectory during the production epoch. The different initial states when the Set is perceived produce trajectories that reach the slow manifold for production at different stages, so that they reach the final state at different time points. **D** Dynamics in the recurrent subspace. There are two stable fixed points (dots filled in white), two saddle points and one unstable fixed point at the origin (empty dots). **E** Neural trajectories projected onto the recurrent subspace. Parameters: $N = 1000$, membrane time constant $\tau = 200$ms. The two input pulses, Ready and Set, are received along the same spatial pattern.

Estimation A trained rank-two network is able to solve the Ready-Set-Go task as well. In this case, the only inputs to the recurrent network are two time pulses, e.g., two rapid changes in the neural state, so that the dynamics generated by the recurrent connectivity must account for estimating and then producing the interval (Fig. 3.5 A). The trial starts with a first pulse that generates a slow transient trajectory. In the example we are showing, this trajectory would decay slowly back to the initial fixed point if there were no second pulse (Fig. 3.5 B). The essential feature is that this transient trajectory is slow enough so that the neural state when the 'Set' pulse arrives is still not at a fixed point, even for the longest trained interval. Under this condition, when the second pulse is received, different intervals correspond to different points in the trajectory and are then mapped onto different

neural states. This serves as the basis for the different initial conditions required to produce the corresponding time intervals (Fig. 3.5 C).

We found that the 2D dynamics generated by the network are similar to those in the Cue-Set-Go task: a slow ring-like manifold, that connects two stable fixed points separated by two saddle points (Fig. 3.5 D). The network is initialized at one of the stable fixed points, when the first pulse elicits a transient response that decays back to the initial fixed point, following the slow manifold. The second pulse then sends the trajectories beyond the saddle point, so that they evolve towards the right stable fixed (Fig. 3.5 E). The readout signal ramps up at distinct speeds due to the fact that neural trajectories after the Set is received are at different positions on the ring manifold. Neural trajectories then evolve behind each other along the slow manifold towards the final state.

This mechanism, found in trained rank-two networks solving Ready-Set-Go, is not extendable to tasks where the time interval estimate must be stored in memory. The information about the estimated interval is mapped onto different neural states when the 'Set' pulse arrives, and it is immediately used to produce trajectories that take different times to reach a final state. The information of the estimate (the neural states on the first transient trajectory) is only used at the specific time that separates the estimation and production epoch.

This result is partly at odds with the analysis of neural recordings: in the data, the activity after the first pulse moves towards a neural state different from the initial one. However, this is not an essential feature used by trained networks for solving the task. The second point of disagreement is that trajectories do not seem to follow one behind each other along the same path towards a final state; but they evolve in parallel trajectories at different speeds towards a final fixed point. This main point of disagreement suggests to look for alternative classes of solutions. We found that when the temporal estimate has to be kept in working memory, trained low-rank recurrent neural networks show a geometry in state-space similar to the one found in the neural data.

Storage Recurrent neural networks must be at least rank three to solve the Measure-Wait-Go task, that requires estimating an interval, holding it in memory for a random delay, and then producing it (Fig. 3.6 A-B). First, we explored the dynamical landscape of the trained rank-three networks, by initializing the dynamics at a random state, and letting the trajectories evolve autonomously. We observe that the trajectories move quickly towards a sphere, and then evolve slowly on its surface (Fig. 3.6 C). Given that there are no tonic inputs in this network, the spherical manifold must be generated by the trained recurrent connectivity.

Then, we analyzed the dynamics on the surface of this 3D manifold by using two parameters: the altitude (an angle bounded between 0 and π) and the azimuth or latitude (an angle bounded between 0 and 2π). On the spherical manifold there are two regions, opposed to each other, where the speed Q is almost zero (Fig. 3.6 D, blue areas), so that trajectories at that state will barely move. These regions extend along a short segment of the manifold, and can function locally like a line attractor, because they can store a continuum of neural states over a time period much longer than the trial duration.

As a second step, we look at the path that neural trajectories take in collective space to solve the task. We found that the trajectories are constrained most of the trial duration to the surface of the spherical manifold. For that reason, we project the neural trajectories on the spherical manifold to describe how they solve the task (Fig. 3.6 E). At the beginning of the estimation epoch, the response is similar to the Ready-Set-Go task: the first pulse generates a slow transient trajectory (Fig. 3.6 E, left). The second pulse, indicating the end of the period to be estimated, is received when the network is at different states on this transient trajectory. The key point is that after the second pulse, the different trajectories decay to the one-dimensional region of the manifold where the speed is almost zero. This

region in neural space functions as a line attractor (Fig. 3.6 E, middle panels). Different intervals are mapped to different states in this region of state-space with extremely slow dynamics, and remain there during the variable delay period. This line attractor implements the memory requirement of the task, since it can hold a continuum of network states over a random period of time, as a function of the previously shown sample interval.

The production epoch shows strong similarities with the production in the Cue-Set-Go task. When the third pulse is perceived, indicating the start of the production epoch, the different trajectories evolve in parallel curves but at different speeds towards a final state (Fig. 3.6 F, right). The common subspace in which trajectories are parallel to each other, which is a two-dimensional subspace, constitutes the temporal scaling subspace. The orthogonal dimension within the recurrent subspace corresponds to the input subspace. The main difference with respect to the Cue-Set-Go task is that different initial conditions correspond to different states on some region of neural space that functions as a line attractor, whereas in the Cue-Set-Go task, the initial conditions are determined by the external input.

We can conclude that this spherical manifold underlies three computational components: (i) generating a transient slow trajectory after the first pulse pulse, (ii) storing the neural state over a random delay and (iii) producing almost parallel trajectories that go from one side of the sphere towards the opposite side at different speeds.

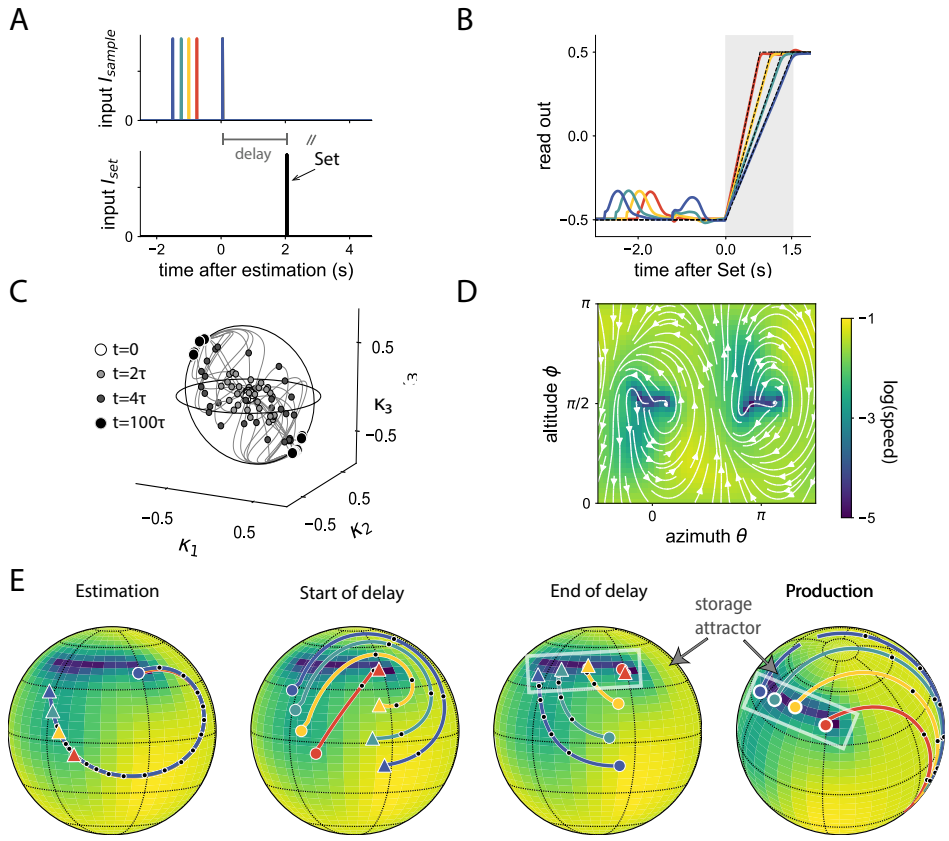

Figure 3.6: **Production, storage and estimation in Measure-Wait-Go task**. **A** Inputs received by the network to perform the Measure-Wait-Go task, temporally aligned at the end of the estimation epoch. The first two pulses are fed through the same spatial pattern, $\mathbf{I}_{sample}$. The third pulse, indicating the beginning of production, is fed through an orthogonal pattern $\mathbf{I}_{set}$. We show the results for four different sampled intervals, corresponding to different colors. The delay period, between the sampled intervals and the 'Set' input varies randomly from trial to trial. **B** Network readout of the trained rank-three network for different intervals. The shaded area indicates the target region used for training in trials with the longest interval. **C** Neural trajectories on the 3D recurrent subspace, when the trained network is initialized at random initial conditions. Initial states correspond to white dots, which are located around the origin of the state-space. The trajectories quickly move towards the surface of a sphere (see light grey dots, for state after two membrane time constants), and evolve slowly on the sphere towards one of two stable regions (see dark grey and black dots for states after 4 and 100 membrane time constants). **D** Dynamics projected on the surface of the sphere, parameterized by the altitude and azimuth of the sphere. The dynamics have been rotated, so that the equator (altitude $= \pi/2$) coincides with the plane of the two attractive regions. Instead of fixed points, this network created segments in neural space where the activity is very slow. **E** Neural trajectories projected on the spherical manifold during the three epochs of the task: estimation, delay period (divided into two halves) and production. The color map indicates the speed of the dynamics (scale as in **D**). During estimation, the first pulse produces a slow transient trajectory on the sphere. The dots correspond to the state at the beginning of the epoch, diamonds to the end of the epoch. At the beginning of the delay, when the second pulse is perceived, the activity decays quickly towards one of the slow attractive regions on the sphere. During the rest of the delay, the different sampled intervals stay almost constant at different positions on this region. For production, trajectories evolve in parallel trajectories from one side of the sphere towards the opposite state on the sphere. Parameters: $N = 1000$, membrane time constant $\tau = 100$ms. The delay is fixed to 2 s in **A** and **E** and to 1.2s in **B**.

3.4 Dynamical components

The inspection of the dynamics in trained recurrent neural networks revealed the following candidate dynamical components used to solve temporal tasks:

1. The recurrent connectivity generates ring- or spherically-shaped slow manifolds.
2. Fixed points and saddle points are located on these slow manifolds.
3. Ramping signals are generated by neural trajectories evolving from one side of the manifold to the other.
4. The speed of dynamics along these manifolds can be modulated by tonic inputs.
5. Higher-dimensional manifolds can consist of a hierarchy of lower-dimensional manifolds.

We tested the mechanistic role of these dynamical computations by building simplified networks that solve the temporal tasks based on them. In this section, we describe how these dynamical components can be generated in simplified network models. In the next section, we use these simplified models to construct networks that perform the temporal task.

Simplified network models In order to generate the dynamical components described above, we look for the minimal structure in the connectivity using the low-rank neural network models presented in Section 3.3.1. The networks are composed of a large number N of units that are recurrently connected through a connectivity matrix that has low rank R. The dynamics of the N units in the network follow Eq. (3.1). Due to the low-rank structure of the connectivity, the neural state of all units in the network is fully determined by the state of R collective variables, that we denote as κ_r for $r = 1, \ldots, R$. Furthermore, the dynamics of these collective variables can be expressed through a low-dimensional dynamical system of the form

$$\tau \frac{d\boldsymbol{\kappa}}{dt} = \mathbf{F}\left(\boldsymbol{\kappa}\right), \tag{3.9}$$

which is detailed in Eqs (3.4) and (3.6). The vector field $\mathbf{F}$ represents a function that maps an R-dimensional vector to an R-dimensional output vector.

Although large low-rank recurrent networks are reduced in this way to a low-dimensional dynamical system, the parameter space remains high dimensional. The generated dynamical landscape depends on the loadings of the connectivity patterns $m_i^{(r)}$ and $n_i^{(r)}$, for $i = 1, \ldots, N$ and $r = 1, \ldots R$; that is, $N \times R$ parameters.

We therefore further simplify the low-rank network by assuming that the loadings of individual units are sampled from the same fixed probability distribution, in particular, a zero-mean multivariate Gaussian distribution (see Methods, Section 3.7.3). This assumption implies that all neurons in the network are statistically equivalent, as they are samples of the same population. The theoretical framework of low-rank networks with multiple population was developed in Beiran et al. (2020). We focus in this study on the case of one single neural population. As a consequence of this assumption, the only parameters that are relevant for shaping the neural dynamics are the parameters of the generative Gaussian distribution: the correlations between connectivity pattern loading.

We denote the correlation between two connectivity pattern loadings, $m^{(s)}$ and $n^{(r)}$, by the symbol $\sigma_{m_s n_r}$. We can consider the correlations between all pattern loadings as a matrix $\boldsymbol{\sigma_{mn}}$, where each element is defined as $\sigma_{m_s n_r}$, for $s, r = 1, \ldots, R$. We refer to this matrix as the covariance matrix or correlation matrix of the network connectivity, because it is composed of pairwise covariances of Gaussian variables, although it is not necessarily a

symmetric or positive definite matrix. The dynamics of collective variables simplify in the mean-field limit of large networks to (see Methods, Schuessler et al. (2020b)):

$$\tau \frac{d\boldsymbol{\kappa}}{dt} = -\boldsymbol{\kappa} + \left\langle \phi'\left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa}\right)\right\rangle \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa}, \tag{3.10}$$

where $\langle \phi'(0, \Delta)\rangle$ is the activity-dependent gain, defined by the Gaussian integral $\int dx\, (2\pi)^{-\frac{1}{2}}\, e^{\frac{-x^2}{2}}\, \phi\left(\sqrt{\Delta} x\right)$.

The only parameters that determine the dynamics in Eq. (3.10) are the entries of the $R \times R$ correlation matrix $\boldsymbol{\sigma_{mn}}$. Based on this, we design the correlation matrices of low-rank networks with Gaussian connectivity that implement the dynamic features observed in trained networks:

#1. Generating slow manifolds The first common feature found in trained networks is the generation by the recurrent connectivity of smooth manifolds in neural space, where the dynamics are much slower than the membrane time constant of single neurons. Trajectories on the recurrent space initiated at a random state quickly converge to this region. Furthermore, we found that manifolds correspond to regions with a fixed radius in recurrent space, or regions whose radial distances are constrained around a fixed radius. In rank-two networks, slow manifolds display a ring shape (Fig 3.4 E), whereas in rank-three networks, manifolds are topological spheres (Fig 3.6 C).

To see how such manifolds occur in our simplified networks, it is useful to write the dynamics in Eq. (3.10) in polar coordinates. In networks of any rank R, we define the radial coordinate r of the recurrent space as

$$r = \sqrt{\boldsymbol{\kappa}^T \boldsymbol{\kappa}}. \tag{3.11}$$

The dynamics in the radial component, calculated as $r\dot{r} = \boldsymbol{\kappa}^T \dot{\boldsymbol{\kappa}}$, read

$$\tau \frac{dr}{dt} = -r + \frac{1}{r} \left\langle \phi'\left(0, r^2\right)\right\rangle \boldsymbol{\kappa}^T \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa}. \tag{3.12}$$

Slow manifolds can be simply generated using an isotropic correlation matrix, i.e., the matrix is proportional to the identity matrix:

$$\boldsymbol{\sigma_{mn}} = \sigma_{mn} \mathbf{I}, \tag{3.13}$$

where σ_{mn} is a scalar parameter, and $\mathbf{I}$ is the identity matrix. Such a correlation matrix of the connectivity patterns implies that the generative Gaussian distribution from which loadings are sampled is isotropic with respect to all connectivity patterns. Left and right connectivity patterns corresponding to the same rank term r have equal correlations,

$$\sigma_{m_r n_r} = \sigma_{mn} \qquad \text{for} \quad r = 1, \ldots, R, \tag{3.14}$$

while patterns of different rank terms are uncorrelated

$$\sigma_{m_r n_s} = 0 \qquad \text{for any} \quad r \neq s. \tag{3.15}$$

The correlations in Eq. (3.14) correspond to the diagonal entries of the correlation matrix $\boldsymbol{\sigma_{mn}}$, and the off-diagonal entries (Eq. 3.15) represent the correlations between connectivity patterns of different rank-one terms.

Using the correlation matrix in Eq. (3.13), the dynamics in the radial direction (Eq. 3.12) simplify to

$$\tau \frac{dr}{dt} = -r + \left\langle \phi'\left(0, r^2\right)\right\rangle \sigma_{mn} r. \tag{3.16}$$

Consequently, the radial component of the dynamics cancels out if $\sigma_{mn} > 1$ at a radius r_0 determined by

$$\sigma_{mn}^{-1} = \left\langle \phi' \left(0, r_0^2\right) \right\rangle . \tag{3.17}$$

We can show that if the correlation matrix is isotropic there is no other flow in non-radial directions (see Methods 3.7.3), so that all states at a distance r_0 with respect to the origin are fixed points. Since a fixed point is generated at the same distance in every direction of the recurrent subspace, the dynamics produce a continuum of fixed points. Furthermore, this continuum of fixed points is stable (see Methods 3.7.3). Therefore, we conclude that rank-R networks with a strong isotropic correlation between connectivity patterns lead to R-dimensional spherical attractors. In rank-two, a ring attractor is generated (see Fig. 3.7 A, left and center), and in rank-three, a spherical attractor.

The above analysis is valid only in the mean-field limit of very large networks. In finite-size networks, where the loadings are randomly sampled from a multivariate Gaussian distribution, there is always some variability due to the finite number of samples. In practice, these finite-size effects introduce spurious correlations between connectivity patterns, so that, even if the spurious correlations are weak and decrease with the network size, the correlation matrix $\boldsymbol{\sigma_{mn}}$ is not exactly isotropic. As a consequence, not all points on the attractive spherical manifolds are fixed points (Fig. 3.7 A, right). The general properties of the dynamics however stay unaffected: trajectories are quickly attracted to the manifold in the radial direction, because the speed function Q is high away from the manifold. In the mean-field limit, the speed Q on the manifold is zero. However, in finite networks, the manifold speed is low, but different from zero. Thus, once trajectories reach the spherical manifold, trajectories evolve slowly along its surface. Usually, the spurious correlations generate one pair of stable fixed points somewhere on the spherical manifold, symmetrically positioned at opposite sides of the origin. Additionally, two unstable fixed points (saddle points) appear on the manifold, separating the stable fixed points. The orientation of these fixed points on the manifold is however random, it changes unpredictably at every new sampling of the connectivity patterns. Therefore, when the correlation matrix is close in parameter space to the correlation matrix producing a *spherical attractor* in the mean-field limit, a *slow spherical manifold* is generated.

We conclude that when the covariance matrix $\boldsymbol{\sigma_{mn}}$ is close to isotropic, as in Eq. (3.13), slow spherical manifolds are generated in the vicinity of the corresponding mean-field attractor.

#2. Controlling fixed points on ring manifold We explain here how the location of stable fixed points and saddle points on the slow manifold can be controlled. We focus on rank-two networks that generate slow ring manifolds. Their correlation matrix $\boldsymbol{\sigma_{mn}}$, of size 2×2, must be close to isotropic. We further constrain the network by adding weak perturbations to this correlation matrix to fix the location of fixed points on the manifold. Such perturbations shape the dynamics so that the number, stability, and average location of fixed points is constant for different samplings of networks with common statistics.

One simple possibility is to use correlation matrices of the form

$$\boldsymbol{\sigma_{mn}} = \begin{pmatrix} \sigma_{mn} + \Delta & 0 \\ 0 & \sigma_{mn} - \Delta \end{pmatrix}, \tag{3.18}$$

where the perturbations $\Delta > 0$ are much smaller than σ_{mn}, to preserve the slow ring manifold, but strong enough so that $|\Delta|$ is larger than spurious correlations in finite networks (see Fig. 3.7 B left for an example).

The fixed points of the network are located on the axes κ_1 and κ_2 (Fig. 3.7 B, center). The ones on the κ_1 direction are stable, while the fixed points on the κ_2 direction are saddle points; they are stable in all directions except for the direction along the manifold, which is

unstable (Methods, section 3.7.3). The proximity to the ring attractor in parameter space guarantees the generation of a slow manifold that connects these four fixed points.

We can express the mean-field dynamics in polar coordinates, using the radial distance r (defined in Eq. 3.12) and the angle with respect to the κ_1-axis, θ, so that any state (κ_1, κ_2) is parameterized as $(r\cos\theta, r\sin\theta)$. The dynamics in polar coordinates, using the correlation matrix in Eq. (3.18) read

$$\tau\frac{dr}{dt} = -r + \left\langle \phi'\left(0, r^2\right)\right\rangle \sigma_{mn} r + \left\langle \phi'\left(0, r^2\right)\right\rangle \Delta r \cos 2\theta \tag{3.19}$$

$$\tau r\frac{d\theta}{dt} = -\left\langle \phi'\left(0, r^2\right)\right\rangle \Delta r \sin 2\theta \tag{3.20}$$

The direction of the dynamics along the slow manifold is given by the angular speed (Eq. 3.20). In the first and fourth quadrant of collective space ($|\theta| < \pi/2$), the angular speed is negative, i.e., pointing clockwise, so that the trajectories initiated in this region will evolve slowly along the manifold towards the fixed point in the positive κ_1 direction. In the second quadrant, the angular speed changes sign, so that trajectories initiated in this region will move towards the fixed point at negative κ_1 (see Fig. 3.7 B center). Therefore, the trajectories on the slow manifold always move from a saddle point (attractive in all directions except along the manifold) towards a stable fixed point.

From a dynamical systems perspective, the slow manifold described here corresponds to a heteroclinic cycle (a series of trajectories that start and end at different fixed points, and define a closed curve). Furthermore, this heteroclinic cycle is stable, because all the fixed points are stable in directions orthogonal to the manifold. The saddle points are unstable only in the direction tangential to the slow manifold. Such a combination of a saddle point that attracts trajectories in all but one directions and then redirects them towards a stable fixed point is a novel instance of a *stable heteroclinic channel*. Stable heteroclinic channels have been hypothesized as a general dynamical principle for robust transient behavior in the brain (Rabinovich et al., 2006, 2008, 2015).

By adding specific small perturbations to the correlation matrix $\boldsymbol{\sigma_{mn}}$, we are therefore able to generate slow manifolds where we control the location of the fixed points. Finite-size simulations of these networks qualitatively reproduce the dynamical landscape predicted by mean-field theory (Fig. 3.7 B right). As a result, the generated slow manifolds consist of a sequence of heteroclinic orbits that are robust to different samplings of the random connectivity.

#3. Controlling trajectories along ring manifolds We explain in this section how to adjust the dynamics of trajectories along the slow manifold. We have studied above how to fix stable fixed points separated by saddle points on the slow manifold. However, the saddle points appear consistently in the middle points on the manifold between the two stable fixed points. We describe here how to adjust the position of saddle points on the manifold.

So far, we have not included any correlations between connectivity patterns corresponding to different rank components, $\sigma_{m_r n_s}$ for $r \neq s$. (Eq. 3.15). Including such non-zero correlations allows us to control the location of the saddle points.

We consider the correlation matrix

$$\boldsymbol{\sigma_{mn}} = \begin{pmatrix} \sigma_{mn} + \Delta & \epsilon \\ 0 & \sigma_{mn} - \Delta \end{pmatrix}, \tag{3.21}$$

where we assume again that the value of ϵ, the novel parameter, is small compared to σ_{mn}, so that the dynamics generate a ring manifold.

In this case, the stable fixed points are still at the intersection between the κ_1 axis and the ring manifold. The saddle points however are located in the direction that forms an

angle $\theta = \arctan(-2\Delta/\epsilon)$ with respect to the κ_1 axis (see Methods, section 3.7.3). When $\epsilon = 0$, the saddle points are in the κ_2 direction, orthogonal to the stable equilibria. As ϵ increases, the saddle points are rotated along the slow manifold towards the κ_1 direction.

In the limit case $\Delta = 0$, the saddle points coincide with the stable fixed points in the κ_1 axis. In that case, shown in Fig. 3.7 C, the saddle points and stable fixed points merge into one pair of fixed points. These fixed points are half-stable: they are stable if perturbed in one direction along the manifold, and unstable to perturbations in the opposite direction. Trajectories generated by unstable perturbations evolve along the ring, towards the opposite fixed point. In finite size-networks, instead of producing half-stable fixed point, the network generates a saddle point and a stable fixed point close to each other, around the κ_1 axis (Fig. 3.7 C, right).

In sum, we described a robust mechanism for producing long, stable trajectories in response to a small input in a specific direction.

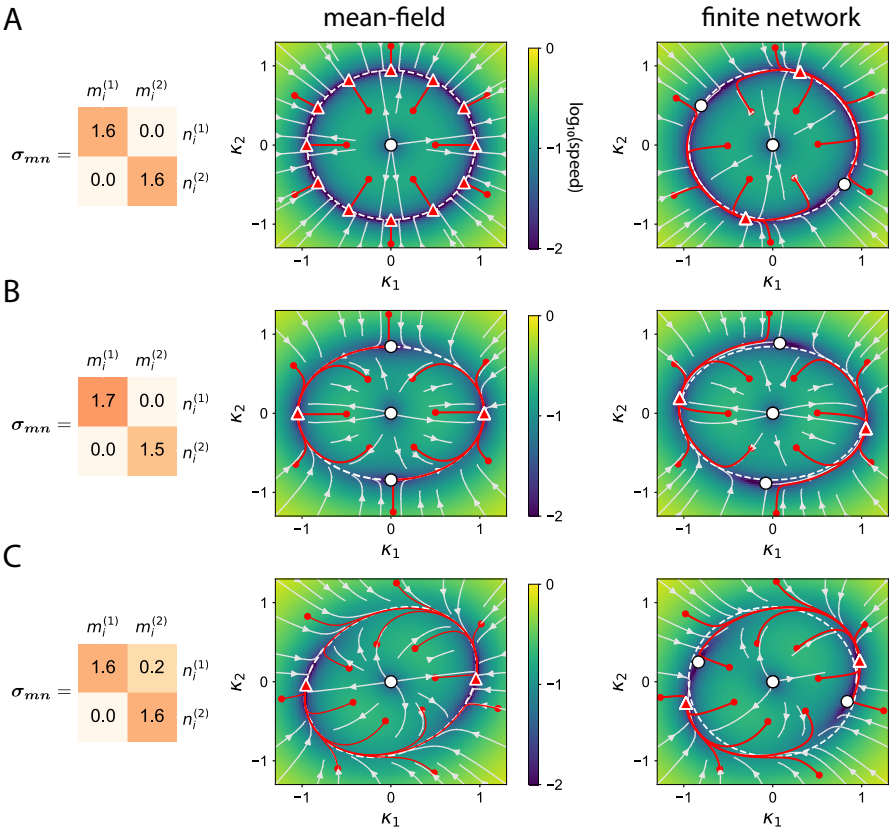

Figure 3.7: **Shaping dynamics on ring manifolds**. Left column: correlation matrix $\sigma_{\mathbf{mn}}$ of the considered rank-two network. Middle column: Mean-field dynamics generated by the network. The colormap indicates the logarithm of the speed Q at every state of the recurrent space. Streamlines are plotted as grey oriented curves. The dashed line corresponds to the slow manifold. In red, mean-field trajectories initiated at the red dots, that evolve towards the diamond symbols (fixed points). White dots correspond to repellers or saddle points. Right column: Dynamics generated by one random sampling of a network with $N = 1000$ units. The colormap, streamlines and unstable fixed points (white dots) illustrate the dynamics of the finite-size network. The dashed line represents the slow manifold from the mean-field description (the same shown in the middle column, for comparison). Red lines, starting at the red circles and finishing at the diamonds correspond to trajectories of the finite size network with different initial states. **A**: Rank-two network generating a slow manifold (correlation matrix given by Eq. 2.33). In the mean-field description, the network produces a continuous ring attractor. In finite networks, the ring attractor turns into a slow manifold, with two fixed points and two saddle points on it. **B**: Rank-two network generating a slow manifold, with two stable fixed points in the κ_1 direction and two saddle points in the κ_2 direction (Eq. (3.18)). The slow manifold and location of the fixed point remains approximately unaltered in finite-size networks. **C** Rank-two network generating a slow manifold with half-stable fixed points (correlation matrix given by Eq. (3.21), with $\Delta = 0$) The mean-field dynamics generate a slow manifold, which is a cycle with two half-stable fixed points along the κ_1 direction (red diamonds). Clockwise perturbations around the fixed point will make the trajectory move along the slow manifold towards the other fixed point. Counterclockwise perturbations decay back to the initial fixed point. In finite networks, similar dynamics are obtained. Instead of obtaining half-stable fixed point, the network generates either a stable fixed point close to a saddle point as shown in the right panel, or a clockwise limit cycle, with very slow dynamics close to $\kappa_2 = 0$.

#4. Controlling the speed along the manifold with a tonic input We address now the question of how the amplitude of a tonic input can control the speed along neural manifold, without modifying considerably the shape and position of the manifold. We extend the dynamics in Eq. (3.10) to account for a tonic input of amplitude u_I along the input pattern $\mathbf{I}$:

$$\tau \frac{d\boldsymbol{\kappa}}{dt} = -\boldsymbol{\kappa} + \left\langle \phi' \left(0, u_I^2 + \boldsymbol{\kappa}^T \boldsymbol{\kappa}\right) \right\rangle \left(\boldsymbol{\sigma_{mn}} \boldsymbol{\kappa} + u_I \boldsymbol{\sigma_{nI}}\right), \tag{3.22}$$

We assume that the input pattern $\mathbf{I}$ is a vector whose loadings are drawn from a normal distribution, which can be correlated with the $\mathbf{n^{(r)}}$ connectivity patterns, but not $\mathbf{m^{(r)}}$. The R-dimensional vector $\boldsymbol{\sigma_{nI}}$ measures the covariances between connectivity patterns $\mathbf{n^{(r)}}$ and vector I. Therefore, within this framework, modeling the input pattern reduces to modeling the correlations between the right connectivity patterns and the external input, $\boldsymbol{\sigma_{nI}}$.

We show how the input affects the two-dimensional network with correlation matrix $\boldsymbol{\sigma_{mn}}$ in Eq. (3.18) (Fig. 3.8 A-B). In the limit case of zero amplitude, $u_I = 0$, we retrieve the slow ring manifold already studied (see Fig. 3.7 B). In the opposite limit case, for very large input amplitude u_I, the network generates one single fixed point in the direction given by $\boldsymbol{\sigma_{nI}}$ and there is no ring manifold (Methods, section 3.7.3). Between these two limit cases, when the amplitude u_I is weak, the unstable node at the origin ($r = 0$) moves in the direction given by $-\boldsymbol{\sigma_{nI}}$. At the same time, the ring manifold remains almost unchanged except for the region closest to the unstable node, which bends in towards this central unstable node (Fig. 3.8 C). Crucially, the speed of the dynamics along the manifold is also affected by the input. We observe that on some regions of the manifold the amplitude increases the speed on the manifold, while on the opposite regions it decreases the speed of the flow along the manifold (Fig. 3.8 D, modulation by more than a factor two).

The speed modulation by the amplitude of an external input is shown here for a particular choice of the correlation matrix $\boldsymbol{\sigma_{mn}}$. However, we found that the mechanism also applies to other spherical manifolds (not shown). An external input correlated with the connectivity patterns modulates the speed of dynamics on the surface of a spherical manifold, without strongly affecting the location of most regions of the manifold.

#5. Combining ring manifolds in higher rank networks Different slow ring manifolds can be generated simultaneously by recurrent networks with rank higher than two. We describe here an example rank-three network that combines a ring manifold with two stable fixed points and two saddle points (Fig. 3.7 B) with a ring manifold with two half-stable fixed points, as in Fig. 3.7 C.

The network dynamics are now embedded in three dimensions, given by collective variables κ_1, κ_2 and κ_3 and determined by the 3×3 correlation matrix $\boldsymbol{\sigma_{mn}}$. Nevertheless, the dynamics restricted to each of the planes κ_1—κ_3, κ_2—κ_3 and κ_1—κ_3 are only determined by the corresponding 2×2 block matrices of $\boldsymbol{\sigma_{mn}}$, so that it is possible to combine different ring manifolds within a sphere. The general rule is that the dynamics within the plane containing the low dimensional manifold remain unperturbed when new rank-one terms are added to the connectivity matrix. However, there are specific constraints: it is possible that stable fixed points in the low-dimensional manifold are unstable in the novel directions introduced by the new rank-one terms.

We focus on a rank-three network with correlation matrix

$$\boldsymbol{\sigma_{mn}} = \begin{pmatrix} \sigma_{mn} + \Delta & 0 & \epsilon \\ 0 & \sigma_{mn} - \Delta & 0 \\ 0 & 0 & \sigma_{mn} + \Delta \end{pmatrix}. \tag{3.23}$$

We assume that $|\Delta| \ll 1 < \sigma_{mn}$ so that the network generates a spherical manifold, since its correlation matrix is close to being isotropic (see Fig. 3.8 E). Within the plane κ_1—κ_2,

the dynamics are determined by the corrsponding 2×2 matrix given by the correlations between connectivity patterns $\mathbf{m^{(1)}}, \mathbf{m^{(2)}}, \mathbf{n^{(1)}}$, and $\mathbf{n^{(2)}}$. This matrix corresponds to the upper left block matrix of $\boldsymbol{\sigma_{mn}}$. Since this block matrix is equal to Eq. (3.18), the dynamics on this plane include to saddle points along the κ_2 direction, and two fixed points along the κ_1 direction that are stable to perturbations within the considered plane (Fig. 3.8 F).

In the orthogonal plane κ_1—κ_3, the corresponding 2×2 matrix that determines the dynamics is given by the elements at the corners of $\boldsymbol{\sigma_{mn}}$:

$$\begin{pmatrix} \sigma_{mn} + \Delta & \epsilon \\ 0 & \sigma_{mn} + \Delta \end{pmatrix}. \tag{3.24}$$

Hence, there are two fixed points along the direction κ_1 which are half-stable in the κ_1—κ_3 plane. Small perturbations along the κ_3 direction will either decay back to the fixed point or evolve along the spherical manifold towards the opposite fixed point, depending on the direction of the perturbations (clockwise or counterclockwise, Fig. 3.8 G).

Finally, in the remaining plane κ_2—κ_3, the correlation structure is given by the bottom left block matrix

$$\begin{pmatrix} \sigma_{mn} - \Delta & 0 \\ 0 & \sigma_{mn} + \Delta \end{pmatrix}. \tag{3.25}$$

Therefore, there are two saddle points in the κ_2 direction, and two stable fixed points in the κ_3 direction within the plane. However, we have seen that in the plane κ_1—κ_3, there are no fixed points along the κ_3 direction, so that these points that are stable to perturbations along the κ_2—κ_3 plane are not fixed points of the dynamics.

Putting all these results together, we found that this network generates a spherical manifold with the following properties: there are two fixed points in the κ_1 direction, that are stable within the κ_1—κ_2 plane, and produce a slow and long trajectory between fixed points within the κ_1—κ_3 plane. In the κ_2 direction, there are two saddle points (Fig. 3.8 H). This network combines two previously studied ring manifolds inlaid in orthogonal planes on the surface of the spherical manifold.

In general, it is possible to combine in such a way ring manifolds within a higher dimensional spherical manifold. It is necessary however to check that the stability properties along the desired planes prevail when the remaining dimensions of the recurrent space are considered.

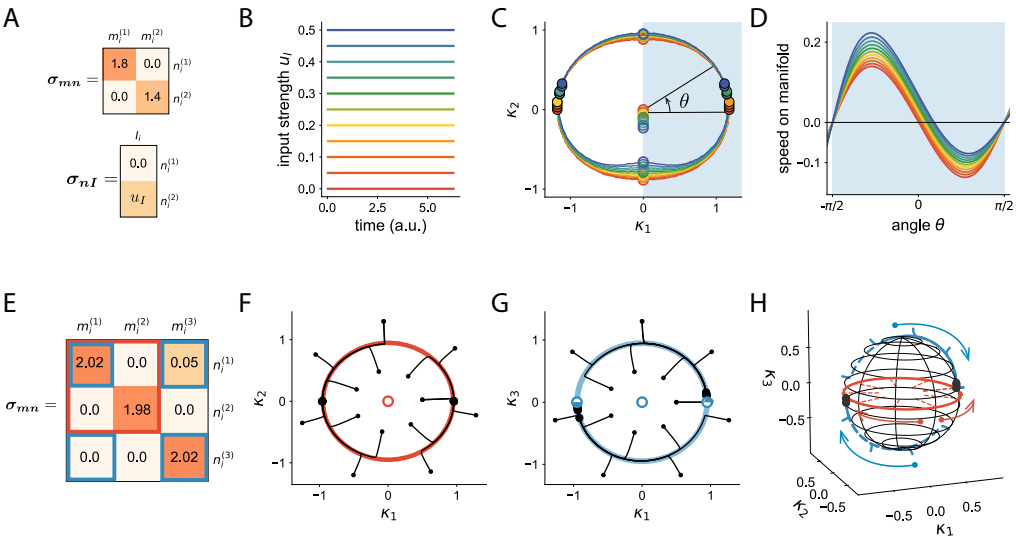

FIGURE 3.8: **Speed control along the manifold with tonic input and combination of ring manifolds on a sphere**. **A** Correlation matrix $\boldsymbol{\sigma_{mn}}$, leading to a slow ring manifold, and correlation between the tonic input and connectivity patterns. **B** Amplitudes of the tonic inputs. **C** Slow manifold generated by the network at different amplitudes of the input. Filled dots are stable fixed points, empty dots are saddle points or unstable saddles. As the input's amplitude is increased, the unstable node at the origin moves in the direction $-\boldsymbol{\sigma_{nI}}$, and the manifold bends in that direction. The location of the manifold far from the unstable node remain mostly unchanged. To study the speed of the dynamics along the manifold, we define the intrinsic variable θ. **D** Speed of the dynamics along the manifold as a function of the angle θ. The speed is defined as $\left(\dot{\kappa}_1^2 + \dot{\kappa}_2^2\right)^{-\frac{1}{2}}$. The speed on the manifold is scaled by the input amplitude. When the input is zero (red curve), the four quadrants produce the same speed profile. As the input is increased, the third and fourth quadrants (lower half of the manifold) increase the speed along the manifold, while the first and second quadrant (upper half of the manifold) reduces the speed. **E** Correlation matrix of a rank-three matrix (following Eq. 3.23) that generates a slow spherical manifold because the matrix is close to isotropic. The dynamics of different planes can be studied by analyzing the 2×2 block matrices of $\boldsymbol{\sigma_{mn}}$. The red block matrix determines the dynamics in the plane κ_1–κ_2, while the blue highlighted entries determine the dynamics on the plane κ_1–κ_3. **F** Mean-field dynamics on the κ_1–κ_2 plane. Similar to Fig. 3.7 B, there is a ring manifold in this plane, with two saddle points in the κ_2 direction, and two stable fixed points in the κ_1 direction. Black curves represent trajectories with initial state indicated by small dots, and final states with large dots. The red curve approximates the ring manifold. **G** Mean-field dynamics on the κ_1–κ_3 plane, which are similar to Fig. 3.7 C. There is a ring manifold with two half-stable fixed points along the κ_1 direction. **H** Three-dimensional representation of the spherical manifold generated by the correlation matrix in **E**, containing the two studied ring manifolds (blue and red curves) on orthogonal planes.

3.5 Implementing temporal computations with reduced network models

In this section, we present solutions to the flexible timing tasks using simplified network models with the dynamical components identified from the analysis of trained recurrent networks.

The dynamical components constrain the relations between connectivity patterns of the simplified low rank-network models that solve the task. For implementing a task, it is additionally required to determine the relations between input patterns and connectivity patterns. There are many possible ways of implementing the correlations of the inputs with the connectivity in these tasks, leading to a degenerate set of possible solutions. For instance, we show in Appendix A that input pulses correlated only with left connectivity patterns $\mathbf{m}^{(\mathbf{r})}$ and input pulses correlated only with right patterns $\mathbf{n}^{(\mathbf{r})}$ generate very similar trajectories in the recurrent subspace. We chose therefore to use transient inputs that are correlated with the $\mathbf{m}^{(\mathbf{r})}$ connectivity patterns, so that neural trajectories remain on the recurrent subspace during the whole trial.

Overall, the goal is to determine the general constraints on recurrent neural networks required to solve the timing tasks, in terms of the input and connectivity patterns. We then provide one minimal implementation of each of the timing tasks.

Cue-Set-Go task The minimal rank of a recurrent neural network required to implement this task is two. The network connectivity must generate dynamical components #1 (create a slow ring manifold), #2 (two stable fixed points and two saddle points lie on the manifold) and #4 (the speed along the manifold is adjusted with a tonic input). The two stable fixed points serve as the initial network state at the beginning of the trial, and the final state to which trajectories evolve during production. The cue input controls then the speed along the slow manifold. The additional constraint is that the 'Set' pulse sends the neural state from the initial stable fixed point passed one of the saddle points, so that trajectories then evolve at controlled slow speed along the manifold towards the final stable fixed point.

In all task implementations, as a proxy for the readout activity, which is a linear combination of the firing rates of individual neurons, we project the neural trajectory $\mathbf{x}(t)$ along a readout direction in collective space. For the Cue-Set-Go task, the only condition is that the readout be not orthogonal to the region of the slow manifold along which trajectories evolves towards the final state.

Fig. 3.10 shows one minimal implementation of the task following these constraints. Trajectories evolve from the initial stable fixed point (circles, Fig. 3.10 A), towards the opposite side of the slow manifold thanks to the action of the Set input that moves the network states beyond the separatrix of the two stable points. Projecting the activity onto the read-out direction (black dashed line, Fig. 3.10 A), the output of the network ramps at different speeds towards the final state (Fig. 3.10 B). We then defined the produced interval as the time elapsed between the Set pulse and the crossing of a given threshold (black line, Fig. 3.10 B). Effectively, the recurrent network flexibly maps the amplitude of the input cue to the produced interval t_p. This mapping is a monotonically increasing function of the amplitude (Fig. 3.10 C).

The core mechanism that controls the produced interval is the speed along the manifold. The speed modulation affects every state on the slow manifold; it is not restricted to the neighboring regions of the final state (see Fig. 3.10 D). The linearized dynamics around the final state alone are not able to explain the temporal stretching shown here. It is necessary to take into account the whole extent of the slow manifold explored by neural trajectories to generate this mechanism.

Finally, we can consider the geometrical structure of trajectories in neural space. Taking all trials into account, neural trajectories are constrained to the surface of a cylinder during

most of the trial duration (see Fig. 3.10 E for two different 3D views of the trajectories in the production epoch). The longitudinal axis of this cylinder corresponds to the input cue direction. The location of the trajectories along this axis therefore changes on a trial-by-trial basis. On the other hand, within a given trial, trajectories evolve in planes orthogonal to the cylinder axis. This is the recurrent subspace, which coincides here with the temporal scaling subspace. This configuration is consistent with the geometry found in neural recordings of non-human primates performing the task (Wang et al., 2018).

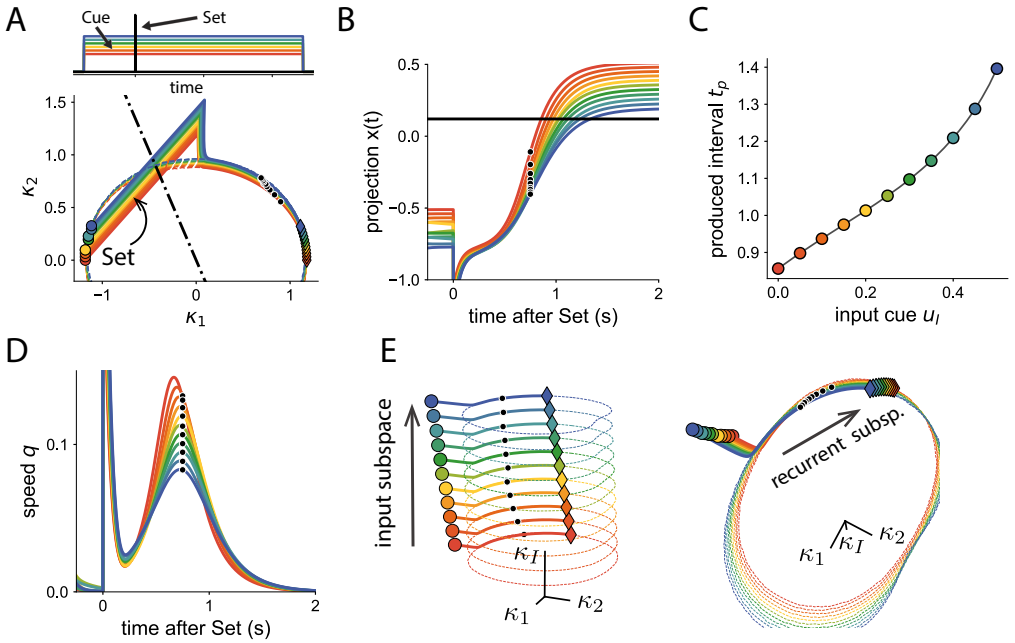

FIGURE 3.9: **Reduced model of rank-two network performing the Cue-Set-Go task**. **A** Bottom: Trajectories in the recurrent subspace solving the task (circles: initial state, diamonds: final state). Top: Inputs fed to the network. Different colors correspond to trials with different input cue amplitude u_I. Black dots indicate the neural state 750ms after the Set pulse is received. The black dashed line corresponds to the readout direction. The trajectories start on a stable fixed point. The Set pulse shift the trajectories to the first quadrant of recurrent space. From there, trajectories reach the slow manifold and move towards the final stable fixed point. **B** Projection of neural trajectories along the readout direction. The projected activity ramps up towards a final state after the 'Set' pulse is received. We define the produced interval as the time it takes to reach the threshold (black line). **C** Relation between the input cue amplitude u_I and the produced interval t_p. The produced interval is longer as the cue increases. **D** Speed profile of different trajectories during the production epoch. Right after Set, the speed is fast, until reaching the slow manifold where the speed goes through a local minimum. Then, on the slow manifold, the speed is scaled according to the background cue. Blue trajectories (larger cues) evolve at a smaller speed than red trajectories (weaker cues). **E** Three-dimensional projections of the neural trajectories during production. The neural space consists of the two-dimensional recurrent subspace, which shows temporal scaling -trajectories overlap-, and the orthogonal input cue direction. The level on this cylinder sets the speed along the recurrent subspace. Parameters: $\tau = 50$ ms, $\sigma_{m_1 n_1} = 1.8$, $\sigma_{m_2,n_2} = 1.4$, Set pulse: $\boldsymbol{\sigma_{mI}} = (1,1)$. Input cue $\sigma_{n_2 I} = u_I$, $\sigma_{I^2} = u_I^2$, $\sigma_{n_1 I} = 0$, orientation of projection vector $\theta = 0.65\pi$, threshold value 0.12.

Ready-Set-Go task The Ready-Set-Go can be solved in a rank-two network, generating a slow manifold (component #1), with two stable fixed points (component #2) separated by saddle points. The constraint on the first input pulse, 'Ready', is that the network generates a robust transient trajectory, slow enough such that it does not reach a fixed point before the end of the sampled interval. The second pulse must then move the neural trajectories towards the basin of attraction of a stable fixed point, in a way such that the shortest sampled interval is closer to the final state following the ring manifold, and the longest sampled intervals is further away. The read-out direction must be not orthogonal to the trajectories evolving on the slow manifold towards the final state.

In the minimal implementation shown in Fig. 3.10, the 'Ready' pulse generates a transient trajectory on the slow manifold that would decay back to the initial stable fixed if no other inputs were fed to the network. Before decaying back to the initial state, the second pulse, 'Set', sends the network to the opposite side of collective space. Trajectories quickly reach the slow manifold and then evolve at slow speed towards the final fixed point. The Set pulse sends the trajectories after a short sampled interval to a point on the slow manifold closer to the final state (red curve Fig. 3.10 A) than for a long sampled interval (blue curve). Therefore, the projected activity on the readout reaches the threshold at different intervals t_p based on the sampled interval t_s (Fig. 3.10 A-B).

Although not strictly necessary, it is be possible to rotate the saddle points along the slow manifold closer to the stable fixed points (dynamical component #3), so that the transient trajectory during estimation and during production explores a longer path of the collective state.

Measure-Wait-Go task A network of rank at least three is required to solve this task, that involves estimating a sample interval, holding the estimate in working memory during a random delay and then reproducing the interval. We use a spherical manifold (dynamical component #1), that contains two orthogonal ring attractors (dynamical component #5), one with very slow speed close to the stable fixed points used for storing the sample interval (dynamical component #2) and the other one used to produce slow parallel trajectories from one side to the other of the sphere (dynamical component #3). The only constraint on the first pulse, that indicates the beginning of the sampled interval, is that it create a slow transient trajectory that does not reach a fixed point before the second pulse is received. The second pulse must elicit a response that sets the neural trajectories at different states close to the stable fixed point of the storage attractor. Neural trajectories then stay during the remaining delay period at different states on this local line attractor. The third pulse, indicating the end of the delay, perturbs the trajectories in the unstable direction of the half-stable fixed point, so that trajectories evolve slowly during production from one side of the sphere to the other. The read-out direction must be co-linear with the parallel trajectories generated during the production epoch.

We use the network presented in Fig.3.8 E-H, that generates a spherical manifold. On the surface of the sphere there is one ring manifold with very low speed that can be used as a line attractor close to its stable fixed points. There is also a second ring manifold, orthogonal to this storage attractor, that produces rotational dynamics from one side to the other of the sphere.

In the minimal implementation we show in Fig. 3.11, we decided for simplicity that the network only explores the plane κ_1–κ_2 during the estimation epoch and the delay. The first pulse elicits a slow transient response using the slow manifold on this plane (Fig. 3.11 A left). The second pulse positions the neural trajectories at a new state, so that, after a fast transient response to reach the slow manifold, trajectories reach different states around a stable fixed point on the manifold. The speed of the dynamics around this fixed point is very low, so that the state barely changes all along the duration of the random delay (Fig. 3.11 A center). The third pulse, which represents the beginning of production, moves

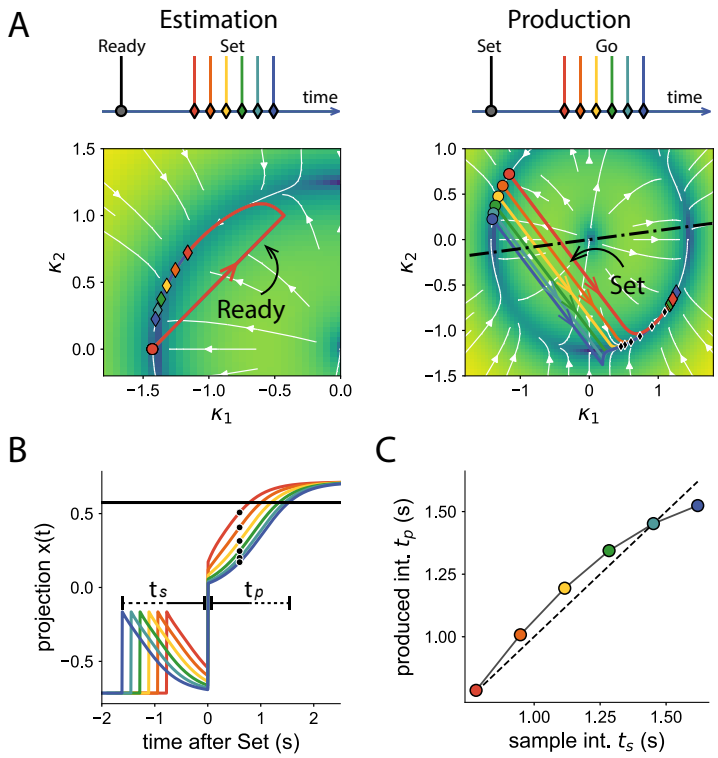

FIGURE 3.10: **Reduced model of rank-two network solving the Ready-Set-Go task**. **A** Top: Schematic of the task during the estimation (left) and production epoch (right). Bottom: Trajectories in the recurrent subspace during the two epochs. The start of the epoch is indicated by a circle and the end of the epoch by a diamond. Curves of different colors correspond to the different sample intervals. The colormap indicates the logarithmic speed Q, and the white lines are the streamlines. Left. Trajectories start at an initial fixed point, and generate a slow transient trajectory in response to 'Ready'. Importantly, the trajectories do not reach a fixed point at the end of the estimation epoch. Right. The Set pulse locates the trajectories at different distances with respect to the final state. Trajectories then evolve along the ring manifold towards this final state. The black dashed line represents the readout direction. Black dots 600ms after Set onset. **B** Projection of the neural trajectory along the readout. The black line indicates the threshold. The projected activity ramps up at different speeds towards a final state after the Set. **C** Mapping between sampled and produced time intervals. Parameters: $\tau = 60$ ms, $\sigma_{m_1 n_1} = 2.1$, $\sigma_{m_2,n_2} = 1.9$, Ready pulse: $\boldsymbol{\sigma_{mI}} = (1, 1)$, Set pulse: $\boldsymbol{\sigma_{mI}} = (1.6, -1.6)$. Readout vector $(1, 0)$. Threshold value 0.58.

the dynamics away from this plane (Fig. 3.11 A right). The trajectories then evolve along the sphere towards the opposite side following close parallel trajectories at the beginning of production, that end up converging to the same final state. The angle between the plane spanned by the trajectories during production and the plane of the line attractor determines the speed of the trajectories (see Appendix B for a detailed study of the production epoch on the spherical manifold). Therefore, when the trajectories are projected on the readout direction (Fig. 3.11 B), they evolve after the Set pulse towards the final state at different speeds. By setting a threshold, we quantify the relationship between sampled interval and produced interval (Fig. 3.11 C).

Neural trajectories are constrained to the surface of a sphere, instead of a cylinder as

in the Cue-Set-Go task. Focusing just on the trajectories during the production epoch (Fig. 3.11 A right, Appendix D for other 3D perspectives), we identify an input subspace, parallel to the storage attractor, which is given by the initial conditions at the beginning of production. Different levels on this input subspace, which is parallel to the κ_2 axis, correspond to different speeds during the rest of the production epoch. The recurrent subspace, largely orthogonal to the input subspace, is defined by the trajectories that rotate from one side to the other of the sphere during production. Importantly, the most relevant difference between production in the Cue-Set-Go task and in the Measure-Wait Go task is that the input subspace is generated in the latter task by the recurrent connectivity of the network instead of the result of an external input. Trajectories evolve on a sphere and not the surface of a cylinder is due to the fact autonomous networks cannot generate cylindrical manifolds.

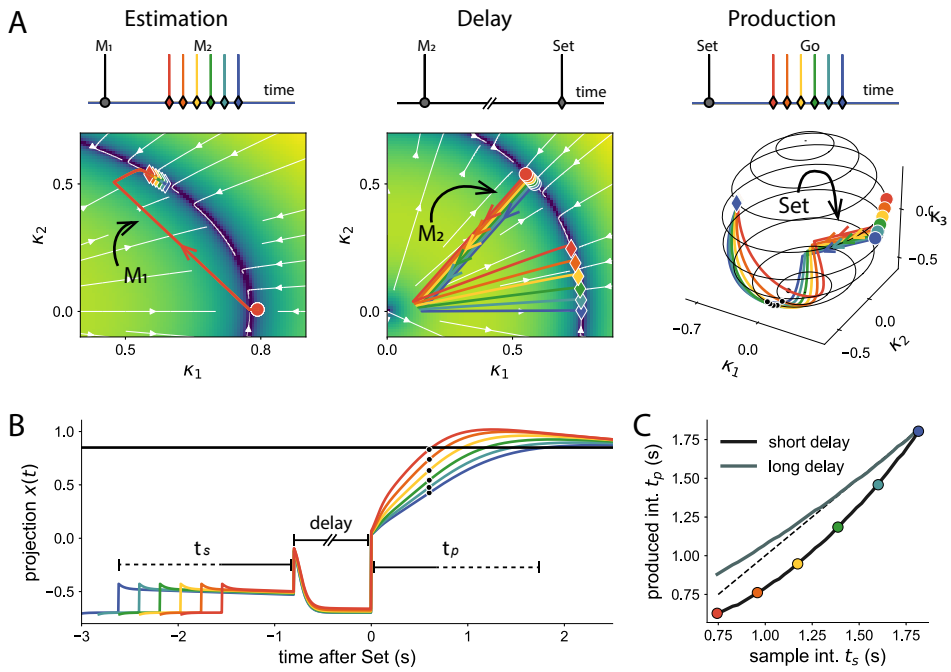

FIGURE 3.11: **Reduced model of rank-three network solving the Measure-Wait-Go task**. **A** Top: Scheme of the inputs during the three different task epochs: estimation (left), delay period (center), and production (right). Bottom: Trajectories in the recurrent subspace during the three epochs. The start of each epoch is indicated by circles and the end by diamonds. Different colors correspond to the different sample intervals r_s. The colormap indicates the log-speed Q and the white lines are the streamlines. Left. The network is initialized at a stable fixed point (red dot). The first pulse sends the network trajectories away from this stable fixed point, producing a slow transient response. At the end of the estimation epoch, the neural state is different for different sampled intervals t_s. Center. The second pulse maps the different sample intervals to different states next to the stable fixed point of the storage attractor. In this case, the second pulse does not move trajectories directly to the slow manifold, but they evolve quickly towards the manifold and stay there during the rest of the delay (in this trial, the delay lasts 600ms). Right: The Set pulse perturbs neural trajectories along the orthogonal κ_3 direction, so that they evolve in parallel towards the opposite stable fixed point along the spherical manifold. **B** Projection of the neural trajectory on the readout. The black lines indicates the threshold. The projected activity evolves at different speeds towards a final state. **C** Mapping between sampled and produced time intervals for two different delays, 600 ms (as shown in panels A and B), and 4500 ms. Different delay periods lead to differently biased produced intervals, due to the very slow dynamics that affect neural trajectories along the local line attractor. Results shown for a fixed delay of 600ms. Parameters: $\tau = 30$ ms. Correlation matrix $\boldsymbol{\sigma_m n}$ as in Eq. (3.23), where $\sigma_{mn} = 2$, $\Delta = 0.005$, $\epsilon = 0.1$. First input pulse $\boldsymbol{\sigma_{mI}} = (-0.3, 0.5, 0)$, second input pulse $\boldsymbol{\sigma_{mI}} = (-0.45, -0.5, 0)$, Set (third input pulse) $\boldsymbol{\sigma_{mI}} = (-0.62, -0.26, -0.19)$. Readout $(-0.9, 0, -0.9)$. Threshold value 0.85.

3.6 Discussion

In this Chapter, we explored the dynamical mechanisms that recurrent neural networks set up to solve flexible timing tasks. We first identified candidate dynamical components used by trained recurrent networks to solve three temporal tasks. The network connectivity generates slow attractive dynamics along a low-dimensional continuum of activity states, that we refer to as *slow manifolds*. Neural trajectories evolve along these slow manifolds to solve the temporal tasks. Secondly, we examined the candidate mechanisms by reproducing them in reduced network models. We assumed that the connectivity statistics of all neurons are generated from the same neural population, so that all neurons are statistically equivalent. Hence, the network dynamics depend only on a few parameters that define the statistics of the neural population. We were then able to describe how such simplified network models generate slow manifolds and control the dynamics along them. Finally, we tested these components by implementing the considered temporal timing tasks using the reduced models. We determined the general constraints on the inputs and connectivity patterns for each of the tasks, and provided one minimal implementation of the tasks using reduced models.

All the recurrent neural networks presented in this study were restricted to have low-rank connectivity matrices. The main advantage of low-rank networks is that they generate low-dimensional network activity, consistent with neural trajectories found in cortical areas. Namely, the network dynamics are determined by a low-dimensional dynamical system of collective variables. This low-dimensional mapping simplifies the analysis of the dynamics implemented by the network and their link to the connectivity structure.

Temporal scaling in recurrent networks. We assumed that cognitive computations are encoded in the dynamic changes of the collective state of recurrent neural networks. An application of this general framework to temporal computations, the *population clock* model, postulates that the often highly complex evolution of neural population activity can be read out as a code that represents time (Laje and Buonomano, 2013; Hardy and Buonomano, 2016; Cueva et al., 2020). At the same time, such time-varying changes of neural activity can simultaneously represent other sensory, cognitive or motor variables. In particular, motor actions can be executed at different speeds, so that neural activity generating those motor commands is temporally scaled accordingly.

Recently, Hardy et al. (2018) investigated how recurrent neural networks can account for temporal scaling. They trained recurrent networks to produce a complex output at different speeds, determined by the amplitude of a background input. Consistent with our analysis of the Ready-Set-Go task, neural trajectories in their setup evolved along neural manifolds with two orthogonal subspaces: a temporal scaling subspace, common to all speed commands, where trajectories overlap, and an orthogonal input subspace that determines the speed of trajectories. Similarly, Bi and Zhou (2020) trained recurrent networks to perform a temporal task resembling the Measure-Wait-Go task, with the difference that instead of generating a linear output ramps, the network's readout is forced to stay at baseline during production and produce a short burst of activity at the end of the production epoch. Neural trajectories in this case also showed strong temporal scaling along a given subspace of neural space, and activity along a non-scaling subspace that correlated with the speed command. This feature was also present in all temporal tasks studied in this Chapter. Such theoretical works suggest that existence of a temporal scaling and a time-invariant speed-controlling subspace is the basis of temporal flexibility, independently of the training algorithm, the dimensionality of neural trajectories or the target output pattern. Consistently, both trained networks and neural recordings in non-human primates performing flexible interval timing tasks found that cortical activity evolves along scaling subspaces along a given trial while the speed is controlled along orthogonal dimensions (Wang et al., 2018; Remington et al., 2018b; Sohn et al., 2019).

Reverse-engineering recurrent neural networks The aforementioned analyses of temporal scaling in recurrent networks limited themselves to a kinematic study of neural trajectories, and a study of the network dynamics close to stable fixed points explored by neural trajectories (Sussillo and Barak, 2013). In this work, we applied the novel theoretical framework of low-rank networks (Mastrogiuseppe and Ostojic, 2018; Schuessler et al., 2020a; Beiran et al., 2020) which allowed us to analyze the dynamics beyond the vicinity of fixed points, in order to extract and interpret the dynamical mechanisms learned by trained recurrent networks solving flecible timing tasks. A parallel study (Dubreuil et al., 2020) followed a similar approach to understand neural computations in decision making tasks that did not required explicit processing of temporal information.

Network vs animal behavior In this study, successfully trained recurrent networks were able to produce precisely timed outputs, although they did not show the variability features and biases observed in the behavior of human and non-human primates performing timing tasks, such as scaling variability or regression towards the average temporal interval. Unlike previous trained networks (Wang et al., 2018; Remington et al., 2018b; Sohn et al., 2019) that showed similar biases to those in observed behavior, networks were not trained in this study using noise in the sensory inputs to study the statistical inference process. The goal here was to identify and propose novel dynamical mechanisms for temporal computations. Our study could be further extended to use trained low-rank networks as behavioral models, by modifying the learning strategy, including adding suitable input noise to the sensory input. Likewise, it is possible to further constrain the reduced network models that implement the tasks, at the level of both input and connectivity patterns, to produce behavioral biases consistent with experimental results. It remains ti be determined whether a network where the connectivity statistics of all neurons belong to one single homogeneous is able to generate the suitable dynamics, or whether additional statistical populations should be considered, which increases the flexibility of the possible generated dynamical landscapes (Dubreuil et al., 2020; Beiran et al., 2020).

In particular, Sohn et al. (2019) showed that animals take into account the statistics of the sampled intervals to improve their performance, and they can flexibly adapt the behavior when the input statistics are modified. This statistical inference process results in produced intervals and neural trajectories that are systematically faster or slower than the sampled interval t_s, depending on the relation between t_s and the known prior statistics of the sensory stimuli. Tonic inputs from other brain areas implementing the required inference could provide constant input to the network, which would globally increase or decrease the speed of the dynamics along the manifold. This mechanism is similar to the effect of the Cue input in the Cue-Set-Go task, which modulates the speed along a ring manifold, but could be generally applied to more complex manifolds, and combined with other computations.

Complementary to the approach here presented, Egger et al. (2020) studied how neural circuit models solve flexible timing tasks, whose outputs match human behavior. The difference with our reduced models is that the neural network solving the tasks has a modular structure. Each module is formed by a group of two or three neurons, that are recurrently connected such that they produce bistable dynamics. By hierarchically combining such modules, it is possible to match the human behavioral outcomes in a wide range of sensorimotor timing tasks. We followed here an integrative approach, where we study the collective dynamics of a homogeneous recurrently connected network, to discover emergent mechanisms that can lead to the same behavior.

Functional role of manifolds To process time, an analog physical quantity, recurrent neural networks generate continuous slow manifolds that are employed to measure time intervals, store estimates and produce time-varying output signals. The continuous nature

of low-dimensional manifolds allows to generalize the temporal computations to unseen temporal intervals, at least within the range of learned intervals. However, discrete fixed points along these manifolds are also required to implement temporal computations. Networks explore different regions of neural space at different task epochs, that are separated by saddle points. During the delay period, the relation between the neighboring fixed points and the neural states corresponding to different stored intervals generate different behavioral biases. Stable fixed points appear also as initial network states or final states to which trajectories evolve to produce a given output pattern.

Slow manifolds might also serve as a useful dynamical structure to quickly learn novel tasks. Presumably, different learning processes are involved in modifying the input projections to local cortical networks than in altering the recurrent connections of a network. Slow manifolds can be reused in different timing tasks, by modulating the mapping of the input patterns, without necessarily changing the recurrent dynamical landscape. Further work is required to test whether networks can be retrained faster if they already generate slow manifolds. From a theoretical point of view, the same slow manifold can be used to solve different tasks. For instance, we used the same network structure (rank-two network generating a ring manifold) to implement the Cue-Set-Go task and the Ready-Set-Go task, with different input patterns. Alternatively, it would have also been possible to use the rank-three network with a spherical manifold, which can solve the Measure-Wait-Go task, also to solve the other flexible timing tasks.

Overall, we presented in this Chapter a set of novel dynamical mechanisms based on low-dimensional slow manifolds, that can be generated by networks with minimal structure in their synaptic connectivity that implement a wide variety of temporal computations.

3.7 Methods

3.7.1 Training of low-rank recurrent networks

Dynamics of trained networks We trained recurrent neural networks with N units and fixed rank R. The dynamics of the total input current received by the i-th unit reads

$$\tau \frac{dx_i}{dt} = -x_i + \frac{1}{N} \sum_{j=1}^{N} J_{ij} \phi\left(x_j\right) + \sum_{s=1}^{S} u_s\left(t\right) I_i^{(s)} + \eta_i\left(t\right). \tag{3.26}$$

The parameter τ is the single unit time constant. The matrix element J_{ij} represents the synaptic strength of the connection from unit j to unit i. The S external inputs, described in the previous section, are separated into their temporal profile $u_s\left(t\right)$ and the strength at which this input is fed to neuron i, $I_i^{(s)}$. The firing rate of neuron i is calculated as $\phi\left(x_i\right) = \tanh\left(x_i\right)$. Each neuron receives independent white-noise $\eta_i\left(t\right)$.

The rank of the connectivity is fixed, so that the connectivity matrix is described as a sum of R rank-one matrices:

$$J_{ij} = \sum_{r=1}^{R} m_i^{(r)} n_j^{(r)}. \tag{3.27}$$

We refer to vectors $\mathbf{m^{(r)}}$ and $\mathbf{n^{(r)}}$ as the left and right connectivity patterns, that constitute the r-th rank-one term of the connectivity. The elements of these connectivity patterns $m_i^{(r)}$ and $n_i^{(r)}$, corresponding to the contribution of the i-th neuron in the connectivity patterns, are called the pattern loadings.

The read-out of the network is defined as

$$z\left(t\right) = \sum_{i=1}^{N} w_i \phi\left(x_i\left(t\right)\right), \tag{3.28}$$

a linear combination of the firing rates of all network units, along the readout pattern $\mathbf{w}$.

The single unit time constant ranges in different trained networks between 30 and 200 ms. The network size N ranges between 300 and 1000 units. We simulated the network dynamics by applying Euler's method with a discrete time step $\Delta t = 10$ ms. The white noise process η_i is generated by drawing values from a zero-mean Gaussian distribution at each time step, with standard deviation 0.08.

Training procedure To train networks, we used backpropagation-through-time (Werbos, 1990). This algorithm minimizes the error between the readout of the network on trial q, $z_q\left(t\right)$, and the target output function for that trial $\hat{z}_q\left(t\right)$. The loss function can be written as

$$\mathcal{L} = \sum_{q} \sum_{t_1^{(q)} < t < t_2^{(q)}} \left(z_q\left(t\right) - \hat{z}_q\left(t\right)\right)^2 \tag{3.29}$$

where q runs over different trials, and $t_1^{(q)}$ and $t_2^{(q)}$ correspond to the minimal and maximal time point taken into account for computing the loss function. We set these boundary values to 50 ms before the beginning of the production epoch, and 50 ms after the end of the production epoch. The target output $\hat{z}_q\left(t\right)$ depends on the particular task, and is detailed in the Section 3.7.2.

The network parameters trained are the connectivity pattern loadings $m_i^{(r)}$ and $n_i^{(r)}$, for $i = 1, \ldots, N$ and $r = 1, \ldots, R$, and the initial network state at the beginning of each trial $x_i\left(t = 0\right)$.

We fixed the loadings of input and output patterns $\mathbf{I^{(s)}}$ and $\mathbf{w}$ by sampling from a random normal distribution at the beginning of training, and trained only the overall amplitudes of these patterns.

We initialized all parameters using random Gaussian variables of unit variance and zero-mean. The covariance between loadings of different connectivity patterns at the beginning of training is defined as

$$\sigma_{m_r n_s} = \sigma_0 \delta_{rs} \tag{3.30}$$

where σ_0 takes the value 0.7 and δ_{rs} is the Kronecker delta. These initial correlations between connectivity patterns generate initial activity with time constant slower than the membrane time constant, which is useful to propagate errors back in time during learning (Schuessler et al., 2020a).

We used a set of 500 trials for the training set, and a set of 100 trials for the test set. Following Dubreuil et al. (2020), we used the ADAM optimizer (Kingma and Ba, 2015) in pytorch (Paszke et al., 2017) with decay rates of the first and second moments of 0.9 and 0.999, and learning rates varying between 10^{-4} and 10^{-2}.

In order to successfully train networks, it was necessary to train first on the flexible timing task using shorter time intervals. We start training networks with (sampled and produced) time intervals at 30% of their duration. Once the network is trained, we increased the duration of time intervals gradually, so that after four steps, the time intervals range between 800 and 1550 ms.

Another hyperparameter important for learning is the number of sampled intervals used during training, and therefore, the number of different time intervals to be produced. For the Cue-Set-Go task, we used six different intervals, unless otherwise specified. In the Ready-Set-Go task and Measure-Wait-Go task, we trained first on two intervals, and then on four intervals. Training with a large number of different sampled intervals from the start often lead to networks that produce one single time interval, the average over all the sampled intervals.

The rank R of trained networks was preset at the beginning of training. For each task, we trained a set of networks with different random initializations, starting with rank-one connectivity matrices. When the readout vector of none of the networks converged to the target output function, we increased the rank by one unit, until finding networks of minimal rank R that solve the task.

The left and right connectivity patterns $\mathbf{m^{(r)}}$ and $\mathbf{n^{(r)}}$of a connectivity matrix $\mathbf{J}$ are found by applying singular value decomposition (SVD). Thus, the left (right) connectivity patterns are orthogonal to each other. However, we do not impose any constraints on the pattern loadings $m_i^{(r)}$ and $n_i^{(r)}$ that are trained. Instead, we recalculate the left and right connectivity patterns after training applying SVD to analyze the network dynamics. As a convention, we set the norm of the left connectivity patterns $\mathbf{m^{(r)}}$ to N without loss of generality.

3.7.2 Design of timing tasks

We studied three different flexible timing tasks, Cue-Set-Go (Wang et al., 2018), Ready-Set-Go (Jazayeri and Shadlen, 2010; Remington et al., 2018b; Sohn et al., 2019) and Measure-Wait-Go. The tasks are illustrated in Fig. 3.2 and the design of inputs and outputs is shown in Fig. 3.3.

Output function All tasks require to produce a time interval t_p after a brief pulse which we denote as 'Set'. The target output of the recurrent neural network is designed as a linear ramp, that starts at value -0.5 when the Set pulse is received, and grows until the threshold value $+0.5$. The produced interval t_p is defined as the time elapsed from the initial value until the threshold value. Thus, different produced intervals correspond to output ramps

with different slopes. The produced time intervals range between 800 ms and 1550 ms, that is, more than one order of magnitude longer than the membrane time constant of single units in the network. In a small fraction of trials, we omit the 'Set' pulse. In that case, the output of the network stays at the initial value -0.5.

Input functions The first transient input in all tasks is fed at a random time point between 200 ms and 800 ms after the beginning of the trial.

In the *Cue-Set-Go* task, two different inputs are received by the network: the 'Cue' and the 'Set' pulse. The Cue input is a tonic input, fed to the network during the whole trial duration along a given pattern $\mathbf{I_{cue}}$. The amplitude of this input u_s informs about the target time interval t_p and ranges between values 0.5 and 1.0. The lowest amplitude corresponds to the smallest produced interval (800 ms, in the network shown in Fig. 3.4), and the largest amplitude (1550 ms, in Fig. 3.4).

In the *Ready-Set-Go* task, the first pulse, 'Ready', indicates the beginning of the estimation epoch. The second delta pulse, 'Set', indicates the end of the estimation and beginning of the production epoch. These two different input pulses can be fed to the network along the same spatial pattern or along two different input patterns. In trained networks, we use the same spatial pattern for both inputs. In designed networks based on simplified low-rank networks, we choose two different spatial patterns, which allows for more flexibility.

In the *Measure-Wait-Go* the first and second pulses are identical to the inputs in the Ready-Set-Go task, and are fed to the network along the same input pattern in trained networks. They indicate the beginning and end of the measurement epoch. After the second pulse, there is a random delay, sampled from a uniform distribution bounded between 200 ms and 2500 ms. A third pulse is given to the network after the random delay, indicating the beginning of the production epoch. In designed networks (Section 3.5), each input pulse is received along different spatial patterns.

3.7.3 Theory of low rank networks: simplified network models

The dynamics of any rank-R recurrent neural network receiving S external inputs can be fully described by the dynamics of R collective variables, κ_r for $r = 1, \ldots, R$, and S collective variables κ_{I_s} related to the external inputs; as desceribed in Eqs. (3.4)-(3.6). The subspace spanned by the κ_r collective variables is the *recurrent subspace*, and the subspace spanned by the κ_{I_s} corresponds to the *input subspace* (Wang et al., 2018).

The collective variables of the recurrent space are defined as the projections of the neural activity $\mathbf{x}(t)$ onto the left connectivity patterns $\mathbf{m^{(r)}}$:

$$\kappa_r(t) = \frac{1}{N}\sum_{i=1}^{N} m_i^{(r)} x_i(t), \tag{3.31}$$

and the input variables are defined as the projections of the neural activity onto the input patterns

$$\kappa_{I_s}(t) = \frac{1}{N}\sum_{i=1}^{N} I_i^{(r)} x_i(t), \tag{3.32}$$

where we assume that the input patterns are orthogonal to the left connectivity patterns. In this section, we focus on recurrent networks receiving a constant external input of amplitude u along the normalized pattern $\mathbf{I}$. In that case, there is only one input collective variable which is also constant in time, $\kappa_I = u$. When the input patterns are not orthogonal to the left connectivity patterns, the input collective variables are defined as the projection along the component of the input pattern orthogonalized with respect to the left connectivity pattern.

We build simplified network models of low-rank networks by considering networks with minimal structure, i.e., random connectivity patterns. For that purpose, we assume that the loadings of connectivity patterns $m_i^{(r)}$ and $n_i^{(r)}$ together with the loadings of a possible tonic input pattern I_i are sampled from a multivariate Gaussian distribution, $P(m_1, \ldots, m_R, n_1, \ldots, n_R, I)$, which has mean zero and fixed covariance. This approach corresponds to the framework presented in Beiran et al. (2020), corresponding to one single neural population, $P = 1$, with zero-mean patterns. It is based on the more general case of random Gaussian connectivity matrices with low-rank perturbations previously studied by Mastrogiuseppe and Ostojic (2018) and Schuessler et al. (2020b).

We then study the dynamics of the collective variables in the limit of large networks ($N \to \infty$). In this limit, that we call the *mean-field limit*, the sum over multiple random samples of correlated Gaussian samples in Eq. (3.6) can be approximated by an integral over Gaussian variables, so that the recurrent dynamics read

$$\kappa_r^{rec} = \int dn_r \, dI \prod_{q=1}^{R} dm_q \, P(n_r, I, m_1, \ldots, m_R,) \, n_r \phi \left(\sum_{q=1}^{R} \kappa_q m_q + \kappa_I I \right). \tag{3.33}$$

Following the same steps as in Appendix 2.7.1, we apply Stein's lemma to Eq. 3.33, which simplifies to:

$$\kappa_r^{rec} = \int dn_r dI \prod_{q=1}^{R} dm_q P(n_r, I, m_1, \ldots, m_R,) \sum_{q=1}^{R} \left(\sigma_{m_q n_r} \kappa_q + \kappa_I \sigma_{n_r I} \right) \phi' \left(\sum_{q'=1}^{R} \kappa_{q'} m_{q'} + \kappa_I I \right), \tag{3.34}$$

where $\sigma_{m_q n_r}$ is a constant (the covariance between loadings m_q and n_r), and, analogously, $\sigma_{n_r I}$ is the covariance between the loadings of the r-th right connectivity pattern and the input pattern. Then, we use the fact that the sum of zero-mean independent Gaussian variables (the input of function ϕ in Eq. (3.34)) is itself a Gaussian variable, with zero-mean and variance equal to the sum of the Gaussian variables. The recurrent dynamics then read

$$\kappa_r^{rec} = \sum_{q=1}^{R} \left(\sigma_{m_q n_r} \kappa_q + \kappa_I \sigma_{n_r I} \right) \int dx \, P(x) \, \phi' \left(x \sqrt{\sum_{q'=1}^{R} \kappa_{q'}^2 + \kappa_I^2} \right) \tag{3.35}$$

where $P(x)$ is the probability density function of a normal Gaussian variable

$$P(x) = (2\pi)^{-\frac{1}{2}} \exp\left(-\frac{x^2}{2} \right). \tag{3.36}$$

Using the notation $\langle f(\mu, \Delta) \rangle = \int dx P(x) \, f\left(\mu + x\sqrt{\Delta} \right)$, the recurrent dynamics read

$$\kappa_r^{rec} = \sum_{q=1}^{R} \left(\sigma_{m_q n_r} \kappa_q + \kappa_I \sigma_{n_r I} \right) \langle \phi'(0, \Delta) \rangle \tag{3.37}$$

where $\Delta = \sqrt{\sum_{q=1}^{R} \kappa_q^2 + \kappa_I^2}$.

Based on Eq. (3.37) we define the covariance matrix $\boldsymbol{\sigma_{mn}}$ as the $R \times R$ matrix with pairwise covariance elements

$$[\boldsymbol{\sigma_{mn}}]_{rs} = \sigma_{n_r m_s} \tag{3.38}$$

for $r, s = 1, \ldots, R$. In this chapter, we refer to the matrix $\boldsymbol{\sigma_{mn}}$ indistinctly as the covariance matrix or the correlation matrix, since the Gaussian variables have zero mean. We also define the column vector of length R, $\boldsymbol{\sigma_{nI}}$ and the state vector $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_R)^T$.

We can then rewrite the dynamics in Eqs. (3.4) in the mean-field limit in vectorial form as

$$\tau \frac{d\boldsymbol{\kappa}}{dt} = -\boldsymbol{\kappa} + \left\langle \phi' \left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa} + \kappa_I^2\right) \right\rangle \left(\boldsymbol{\sigma_{mn}} \boldsymbol{\kappa} + \boldsymbol{\sigma_{nI}}\right). \tag{3.39}$$

Dynamics in the radial direction The equation of the dynamics (Eq. 3.39) maps every possible state $\boldsymbol{\kappa}$ in the recurrent subspace, to a velocity vector $\mathbf{F}(\boldsymbol{\kappa})$ that indicates the direction along which trajectories starting at state $\boldsymbol{\kappa}$ evolve. The norm of the velocity defines the speed at which trajectories evolve, that we denote with the scalar function Q (Eq. 3.7). We study now the dynamics of autonomous networks, so that $\boldsymbol{\sigma_{nI}} = 0$.

In order to analyze the dynamics of low-rank networks, we can project the velocity vector at every state of the recurrent subspace onto the radial direction $\mathbf{u_r}$ (the unitary vector pointing from the origin towards the state $\boldsymbol{\kappa}$). The radial component of the velocity is a scalar variable, indicated as $\frac{dr}{dt}$ or $\dot{r}$. We define the radial distance to the origin with the variable r, defined in Eq. (3.11). It is useful to express the dynamics of autonomous networks in such coordinates, because the non-linear factor of the dynamics $\left\langle \phi' \left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa}\right) \right\rangle$ depends only on the radial distance r.

The expression for the radial component of the dynamics can be calculated using the identity $r\dot{r} = \boldsymbol{\kappa}^T \dot{\boldsymbol{\kappa}}$, as shown in Eq. (3.12).

Apart from the radial component, the remaining velocity can also point in any other direction in the $R-1$-dimensional space orthogonal to the radial direction. We can assess the speed that flows in non-radial directions by computing the difference between the speed at every state, and the speed in the radial component.

The squared speed at every state in the recurrent subspace reads:

$$Q^2 = \dot{\boldsymbol{\kappa}}^T \dot{\boldsymbol{\kappa}} = r^2 - \langle \phi' \rangle \boldsymbol{\kappa}^T \left(\boldsymbol{\sigma_{mn}}^T + \boldsymbol{\sigma_{mn}}\right) \boldsymbol{\kappa} + \langle \phi' \rangle^2 \boldsymbol{\kappa}^T \boldsymbol{\sigma_{mn}^T} \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa}. \tag{3.40}$$

The square of the speed in the radial direction is

$$\dot{r}^2 = r^2 - 2 \langle \phi' \rangle \boldsymbol{\kappa}^T \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa} + \frac{1}{r} \langle \phi' \rangle^2 \left(\boldsymbol{\kappa}^T \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa}\right)^2. \tag{3.41}$$

Therefore, the square of the remaining speed in non-radial directions reads

$$Q^2 - \dot{r}^2 = \langle \phi' \rangle \boldsymbol{\kappa}^T \left(\left(\boldsymbol{\sigma_{mn}}^T - \boldsymbol{\sigma_{mn}}\right) + \langle \phi' \rangle \boldsymbol{\sigma_{mn}}^T \left(\mathbf{I} - \frac{\boldsymbol{\kappa}\boldsymbol{\kappa}^T}{\boldsymbol{\kappa}^T \boldsymbol{\kappa}} \right) \boldsymbol{\sigma_{mn}} \right) \boldsymbol{\kappa}. \tag{3.42}$$

Based on Eq. (3.42), we can show three important features of the dynamics:

- When the correlation matrix $\boldsymbol{\sigma_{mn}}$ is symmetric, the first term in Eq. (3.42) is zero, because $\boldsymbol{\sigma_{mn}}^T = \boldsymbol{\sigma_{mn}}$. The symmetric component of correlation matrices generates only radial dynamics.

- When $\boldsymbol{\kappa}^*$ is an eigenvector of the correlation matrix $\boldsymbol{\sigma_{mn}}$, so that $\boldsymbol{\sigma_{mn}} \boldsymbol{\kappa}^* = \lambda \boldsymbol{\kappa}^*$ and $\boldsymbol{\kappa}^{*T} \boldsymbol{\sigma_{mn}^T} = \boldsymbol{\kappa}^{*T} \lambda$, Eq. (3.42) simplifies to

$$\dot{\boldsymbol{\kappa}}^T \dot{\boldsymbol{\kappa}} - \dot{r}^2 = \langle \phi' \rangle \left(\boldsymbol{\kappa}^T (\lambda - \lambda) \boldsymbol{\kappa} + \langle \phi' \rangle \lambda^2 \left(\boldsymbol{\kappa}^T \boldsymbol{\kappa} - \frac{\left(\boldsymbol{\kappa}^T \boldsymbol{\kappa}\right)^2}{r^2} \right) \right) \boldsymbol{\kappa} = 0. \tag{3.43}$$

  This implies that in the directions given by the eigenvectors of $\boldsymbol{\sigma_{mn}}$ the flow can only point in the radial direction.

- If the correlation matrix $\boldsymbol{\sigma_{mn}}$ is isotropic, the first term is zero, because the matrix is symmetric, and the second term is also zero (all directions are eigenvectors of the correlation matrix), so that the dynamics of the recurrent subspace only evolve in the radial direction.

Spherical attractors When the connectivity patterns of the r-th rank-one component of the connectivity matrix, $\mathbf{m^{(r)}}$ and $\mathbf{n^{(r)}}$, are uncorrelated to the connectivity patterns of any other rank-one component, the correlation matrix $\boldsymbol{\sigma_{mn}}$ is diagonal. Furthermore, if the correlation between patterns $\mathbf{m^{(r)}}$ and $\mathbf{n^{(r)}}$ is the same for all r, as described in Eqs. (3.14) and (3.15), the correlation matrix is isotropic, and has one single (degenerate) eigenvalue σ_{mn}. As shown in the previous paragraph, in that case, the dynamics at every point in the recurrent space are constrained to the radial direction.

Furthermore, if the correlation is strong enough ($\sigma_{mn} > 1$), the network generates a continuum of fixed point at a distance r_0 away from the origin, where r_0 is defined by Eq. (3.17). In rank-two, this continuum of fixed point corresponds to a circle, while in rank-three, the continuum of fixed points are arranged on the surface of a sphere.

To assess the stability of this set of fixed points, we calculate the derivative of the radial speed (Eq. 3.16) with respect to the radial distance:

$$\tau \frac{\partial \dot{r}}{\partial r} = -1 + \left\langle \phi' \left(0, r^2\right) \right\rangle \sigma_{mn} + \left\langle \phi''' \left(0, r^2\right) \right\rangle \sigma_{mn} r^2. \tag{3.44}$$

The spatial derivative of the dynamics (the Jacobian) evaluated at the fixed point r_0 determines the stability. Combining Eqs. (3.17) and (3.44), the radial derivative at the fixed points read

$$\tau \left. \frac{\partial \dot{r}}{\partial r} \right|_{r=r_0} = \left\langle \phi''' \left(0, r_0^2\right) \right\rangle \sigma_{mn} r_0^2, \tag{3.45}$$

which is a negative value, because $\langle \phi''' (0, \Delta) \rangle$ is negative for any value of Δ (see Appendix 2.7.3 in Chapter 2). Therefore, the continuum of fixed points are stable spherical attractors. When the network is initialized at any state in the recurrent subspace away from the origin, the trajectory evolves in the direction between the origin and the initial state, until reaching the attractor at a radial distance $r = r_0$.

Slow manifolds Slow manifolds are defined in dynamical systems theory as a particular type of invariant invariant manifold (see Wiggins (2003) for an introduction to center manifold theory). An invariant manifold of a dynamical system is a smooth region of state-space where trajectories initiated within the manifold stay constrained to the manifold. For instance, a smooth limit cycle is an invariant manifold. The slow manifolds of a fixed point $\boldsymbol{\kappa_0}$ of a dynamical system, with Jacobian matrix A, are the invariant manifolds which correspond to the directions spanned by the eigenvectors of the Jacobian A with zero eigenvalue at the vicinity of the fixed point. In this work, we consider slow manifolds that are spanned by eigenvectors of the Jacobian with eigenvalue very close to zero, not necessarily zero. Hence, we do not take into account dynamics that evolve at much slower timescales than the duration of a single trial. Strictly speaking, those manifolds are classified as unstable or stable manifolds, depending on the sign of the eigenvalues.

For practical reasons, in the analysis of the dynamical landscape of trained networks, we defined as slow manifolds the continuous regions in neural space where the speed is very small (Eq. 3.8), similar to previous theoretical work in neuroscience defining slow points (Sussillo and Barak, 2013). This is a necessary but not sufficient condition to show the existence of a slow manifold in the classical sense of dynamical systems. Nevertheless, we found that such candidate slow manifolds correspond to invariant manifolds that span

trajectories in the vicinity of fixed points tangent to the zero or almost zero eigenvalues of their Jacobian.

In simplified low-rank network models, one example of a slow manifold is the continuum of fixed points generated in the mean-field description of rank-R networks with an isotropic correlation matrix $\boldsymbol{\sigma_{mn}}$. As shown above, they generate a continuum of fixed points arranged on an R-dimensional sphere, where the Jacobian at each fixed point has eigenvectors with zero eigenvalue tangent to the surface of the manifold. This is an attractor, a particular case of a slow manifold, since the dynamics are zero everywhere on the manifold.

In finite-size networks however, the noise from random sampling of the pattern loadings perturbs the isotropic correlation matrix, modifying the dynamical landscape. The perturbations in a finite-size networks introduce some non-zero dynamics that evolve along the surface of the sphere with radius r_0. However, the sphere remains a slow manifold (see Fig. 3.7 A for an example). The network behaves as predicted by the mean-field description far away from the manifold, showing fast dynamics in the radial direction towards the manifold. However, when trajectories converge to r_0, they evolve slowly along the surface of the manifold away from saddle points and towards a stable fixed point or limit cycle. In other words, in finite networks, rank-R networks can generate slow spherical manifolds corresponding to the spherical attractor of the mean-field description. We detail here how to control the dynamics on slow manifolds, by perturbing the correlation matrix $\boldsymbol{\sigma_{mn}}$ in Eqs. (3.14) and (3.15) so that they are robust to random sampling noise. We focus on rank-two networks.

Slow manifolds in rank-two networks We observe that autonomous rank-two networks generate one closed invariant manifold surrounding the origin, if all eigenvalues of the correlation matrix $\boldsymbol{\sigma_{mn}}$ have real part larger than unity. These closed trajectories are attractive, in the sense that when the network is initialized at any state different from the origin, trajectories evolve towards this curve, and then stay on it. This closed invariant curve can have a finite number of fixed points, in which case, it is defined as a *heteroclinic cycle*; it can have no fixed points at all, in which case it is a *limit cycle* (leading to oscillatory solutions), or, it can be a continuum of fixed points, in which case the invariant manifold is a *ring attractor*. If the correlation matrix is close to isotropic (Eqs. 3.14 and 3.15), as we considered in out network models, the dynamics along this attractive trajectory are very slow. For that reason, in the simplified network models we refer to this closed trajectories as *slow manifold*.

We can describe the slow manifold in a rank-two network as a curve $R(\theta)$, or R_θ in short notation, parameterized by one single intrinsic variable, for instance, the angle with respect to the κ_1 axis, θ. We define the slow manifold using the normal vector $\mathbf{u_n}(\theta)$ which is orthogonal to the curve at every point of the trajectory:

$$\mathbf{u_n}(\theta) = -R_\theta \mathbf{u_r} + R'_\theta \mathbf{u}_\theta \tag{3.46}$$

where vectors $\mathbf{u_r}$ and $\mathbf{u}_\theta$ correspond to unitary vectors in the radial and angular directions, respectively, and R'_θ is the derivative with respect to θ of the curve $R(\theta)$.

We can then formally define the slow manifold in rank-two networks as the curve R_θ whose normal vector is orthogonal to the velocity of the dynamics, $\dot{\boldsymbol{\kappa}} = \dot{r}\mathbf{u_r} + r\dot{\theta}\mathbf{u}_\theta$ at every point of the curve:

$$\mathbf{u_n}(\theta) \cdot \left(\dot{r}(R_\theta, \theta)\, \mathbf{u_r} + R'_\theta \dot{\theta}(R_\theta, \theta)\, \mathbf{u}_\theta\right) = 0 \tag{3.47}$$

for any angle $-\pi < \theta < \pi$, given that $Q(R_\theta, \theta) > 0$.

Combining Eqs. (3.46) and (3.47), we obtain the defining condition

$$\dot{r}(R_\theta, \theta) = R'_\theta\, \dot{\theta}(R_\theta, \theta)\,. \tag{3.48}$$

This condition provides no information at fixed points. However, we found that in autonomous rank-two networks all non-trivial fixed points belong to the slow manifold.

Finding a closed-form expression of the slow manifold generated by a correlation matrix $\boldsymbol{\sigma_{mn}}$ is in general a challenging problem. It is often more practical to approximate the slow manifold with an ellipse, using additional information we may have from the dynamical landscape.

In non-autonomous rank-two networks receiving a constant input, there might not be a closed invariant trajectory in the recurrent subspace. For instance, as we show in the next paragraph, networks receiving a very strong tonic input generate a dynamical landscape with one single fixed point, which is stable and located away from the origin. In that case, there is no slow manifold in the recurrent space.

The definition of slow manifolds in these simplified network models can be generalized to networks with rank higher than two, as the smooth $R-1$-dimensional closed surface, where the velocity $\dot{\boldsymbol{\kappa}}$ at each point of the surface is orthogonal to the corresponding normal vector. In rank-three for instance, such closed manifold exists when the real part of all eigenvalues is larger than one, and can be parameterized with two angular values (e.g., the altitude and the latitude).

Controlling the dynamics along ring manifolds We detail here a step-by-step explanation of the dynamical components #2, #3 and #4 described in Section 3.4, relative to rank-two networks. The general approach is to study the dynamics of the simplified network models, Eq. (3.10) in the case of autonomous networks, and Eq. (3.22) for networks receiving a constant input, given different correlation matrices $\boldsymbol{\sigma_{mn}}$. For a clear connection with the main text, we announce at the beginning of each derivation to which dynamical component in Section 3.4 we refer to.

We focus on analyzing fixed points. The fixed point equation is found by setting the velocity $\dot{\boldsymbol{\kappa}}$ to zero in the equation of the dynamics, Eq. (3.10). Rearranging terms, the fixed point equation reads:

$$\boldsymbol{\kappa_0} = \left\langle \phi' \left(0, \boldsymbol{\kappa_0}^T \boldsymbol{\kappa_0}\right) \right\rangle \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa_0}. \tag{3.49}$$

The fixed point equation is reminiscent of an eigenvalue problem: a vector is equal to a matrix times the vector itself. The only vectors $\boldsymbol{\kappa_0}$ that solve the equation are the eigenvectors of the correlation matrix $\boldsymbol{\sigma_{mn}}$. Therefore, we conclude that the only fixed points of the dynamical landscape are located in the directions of the real eigenvectors of the correlation matrix.

#2. Generating fixed points on ring manifold. The first correlation matrix $\boldsymbol{\sigma_{mn}}$ we consider to control the dynamics on the ring manifold is given by Eq. (3.18), and we focus on the dynamics of the autonomous network. This correlation matrix is diagonal, and has two real eigenvalues $\sigma_{mn} \pm \Delta$, with corresponding eigenvectors:

$$\mathbf{v}_+ = (1, 0)^T \tag{3.50}$$

$$\mathbf{v}_- = (0, 1)^T . \tag{3.51}$$

Consequently, the fixed points must lie along the κ_1 and κ_2 axes. Introducing the ansatz $\boldsymbol{\kappa_0^{\pm}} = \rho_{\pm} \mathbf{v}^{\pm}$ in Eq. (3.12), we obtain the following equation for the radial distance of fixed points along both axes:

$$(\sigma_{mn} \pm \Delta)^{-1} = \left\langle \phi' \left(0, \rho_{\pm}^2\right) \right\rangle . \tag{3.52}$$

We can study then the Jacobian around fixed points $\boldsymbol{\kappa_0^+}$ and $\boldsymbol{\kappa_0^-}$ to determine the stability of the fixed points. The Jacobian matrix of the dynamics is obtained by differentiating

Eq. (3.10) with respect to each collective variable, and reads (see Schuessler et al. (2020b), Appendix 2.7.3 in Chapter 2):

$$\nabla \mathbf{F}(\boldsymbol{\kappa}) = -\boldsymbol{I} + \left\langle \phi'\left(0, \boldsymbol{\kappa_0}^T \boldsymbol{\kappa_0}\right)\right\rangle \boldsymbol{\sigma_{mn}} + \left\langle \phi'''\left(0, \boldsymbol{\kappa_0}^T \boldsymbol{\kappa_0}\right)\right\rangle \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa_0} \boldsymbol{\kappa_0}^T. \tag{3.53}$$

Evaluating the Jacobian at the fixed point $\boldsymbol{\kappa_0}^+$, and using Eq. (3.52), we obtain

$$\nabla \mathbf{F}\left(\boldsymbol{\kappa_0^+}\right) = -\boldsymbol{I} + \frac{\boldsymbol{\sigma_{mn}}}{\sigma_{mn} + \Delta} + \left\langle \phi'''\left(0, \rho_+^2\right)\right\rangle \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa_0^+} \boldsymbol{\kappa_0^+}^T. \tag{3.54}$$

This Jacobian has one eigenvalue $-2\Delta/(\sigma_{mn} + \Delta)$, which is negative, with associated eigenvector $\mathbf{v}_-$, and another eigenvalue $\left\langle \phi'''\left(0, \rho_+^2\right)\right\rangle \rho_+^2 (\sigma_{mn} + \Delta)$ which is also negative. Therefore, since the eigenvalues of the Jacobian at this fixed point are negative, the fixed point is stable.

Evaluating now the Jacobian at the fixed point $\boldsymbol{\kappa_0}^-$ we obtain:

$$\nabla \mathbf{F}\left(\boldsymbol{\kappa_0^-}\right) = -\boldsymbol{I} + \frac{\boldsymbol{\sigma_{mn}}}{\sigma_{mn} - \Delta} + \left\langle \phi'''\left(0, \rho_-^2\right)\right\rangle \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa_0^-} \boldsymbol{\kappa_0^-}^T. \tag{3.55}$$

which has one eigenvalue $2\Delta/(\sigma_{mn} - \Delta)$, with eigenvector $\mathbf{v}_+$, that is positive. The other eigenvalue reads $\left\langle \phi'''\left(0, \rho_-^2\right)\right\rangle \rho_-^2 (\sigma_{mn} - \Delta)$, which is negative, and has eigenvector $\mathbf{v}_-$. The fixed points in the κ_2 direction are therefore saddle points: stable in the radial direction, and unstable in the κ_1 direction.

We can then infer the direction of the dynamics along the slow manifold by combining the fact that there are only a pair of stable fixed points symmetrically arranged on the κ_1 axis and a symmetric pair of saddle points on the κ_2 axis with the fact that there must be an attractive slow manifold containing them. Alternatively, as presented in the text, we can analyze the sign of the angular speed $\dot{\theta}$ for different angular values θ given by Eq. (3.20).

#3. Position of fixed points on ring manifold. If the correlation matrix $\boldsymbol{\sigma_{mn}}$ is diagonal, fixed points are generated along orthogonal directions, because the eigenvectors of a symmetric matrix are orthogonal to each other. We can modify the relative location between stable fixed points and saddle points by including off-diagonal terms in the correlation matrix $\boldsymbol{\sigma_{mn}}$. In particular, we use the same correlation matrix as in the previous dynamical component, but now including a small non-zero correlation ϵ between loading variables n_1 and m_2 (Eq. 3.23).

The eigenvalues remain $\sigma_{mn} \pm \Delta$, but the eigenvectors read

$$\mathbf{v}_+ = (1, 0)^T \tag{3.56}$$

$$\mathbf{v}_- = \frac{1}{\sqrt{\epsilon^2 + 4\Delta^2}} (-\epsilon, 2\Delta)^T. \tag{3.57}$$

The angle between eigenvectors is now given by $\arctan(-2\Delta/\epsilon)$, which is zero in the limit $\Delta \to 0$, and $\pi/2$ in the limit $\epsilon \to 0$, which is the case studied in the previous dynamical component.

The radial distance of the fixed points along the direction of the eigenvectors is given by Eq. (3.52), because the eigenvalues did not change. The Jacobian evaluated at the fixed point $\boldsymbol{\kappa_0^+}$ is also given by Eq. (3.54), because the eigenvector $\mathbf{v}^+$ is the same. Therefore, the fixed points in the κ_1 direction are stable in all directions.

The Jacobian around the fixed points in direction $\mathbf{v}_-$ reads

$$\nabla \mathbf{F}\left(\boldsymbol{\kappa_0^-}\right) = -\boldsymbol{I} + \frac{\boldsymbol{\sigma_{mn}}}{\sigma_{mn} - \Delta} + \left\langle \phi'''\left(0, \rho_-^2\right)\right\rangle \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa_0^-} \boldsymbol{\kappa_0^-}^T. \tag{3.58}$$

This matrix has one eigenvalue $\rho_-^2 \left\langle \phi''' \left(0, \rho_+^2\right)\right\rangle (\sigma_{mn} - \Delta)$, which is negative, along the radial direction $\mathbf{v}_-$. The second eigenvalue can be found by calculating the trace of the Jacobian and subtracting the first eigenvalue. We find that this eigenvalue is always positive, unless $\Delta = 0$, where this eigenvalue is zero. Therefore, the fixed points located on the direction $\mathbf{v}_-$ are saddle points.

We showed in Fig. 3.7 C the limit case where $\Delta = 0$. We detail here the derivations for that particular case. The correlation matrix $\boldsymbol{\sigma_{mn}}$ has one single eigenvalue σ_{mn} and one single eigenvector in the κ_1 direction. Therefore, only two fixed points are generated along the only eigenvector direction $\mathbf{u_1}$.

The Jacobian around these fixed points reads

$$\nabla \mathbf{F}\left(\boldsymbol{\kappa_0}\right) = -\boldsymbol{I} + \left\langle \phi' \left(0, \rho_0^2\right)\right\rangle \boldsymbol{\sigma_{mn}} + \left\langle \phi''' \left(0, \rho_0^2\right)\right\rangle \rho_0^2 \boldsymbol{\sigma_{mn}} \boldsymbol{u_1} \boldsymbol{u_1}^T. \tag{3.59}$$

The Jacobian has eigenvectors along the canonical directions $\mathbf{u_1}$ and $\mathbf{u_2}$ with corresponding eigenvalues $\left\langle \phi''' \left(0, \rho_0^2\right)\right\rangle \rho_0^2 \sigma_{mn}$ and 0, respectively. To determine the stability of a fixed point whose Jacobian has at least one zero eigenvalue, additional information must be taken into account.

In this case, we can express the dynamics in polar coordinates, by using the identities $r\dot{r} = \kappa_1 \dot{\kappa}_1 + \kappa_2 \dot{\kappa}_2$ and $r^2 \dot{\theta} = \kappa_1 \dot{\kappa}_2 - \kappa_2 \dot{\kappa}_1$:

$$\tau \frac{dr}{dt} = -r + \left\langle \phi' \left(0, r^2\right)\right\rangle \left(\sigma_{mn} r + \epsilon r \cos\theta \sin\theta\right) \tag{3.60}$$

$$\tau r \frac{d\theta}{dt} = -\left\langle \phi' \left(0, r^2\right)\right\rangle \epsilon r \sin^2\theta. \tag{3.61}$$

The dynamics along the angular component always point in the same direction ($\dot{\theta}$ has the same sign for any value of θ). Therefore, there is a slow manifold where the dynamics rotate in the same direction. The fixed points then are stable if perturbed against the direction of rotation, and unstable if perturbed in the sense of rotation. Such fixed points are named half-stable fixed points (Strogatz, 2000).

#4. Speed control of dynamics on manifold with a tonic input. We study now the effect that a external tonic input, with amplitude u_I and correlated with the connectivity patterns $\mathbf{n^{(1)}}$ and $\mathbf{n^{(2)}}$, produces on the dynamics of a rank-two network. The dynamics are given by Eq. (3.22). We focus on the case where the correlation matrix $\boldsymbol{\sigma_{mn}}$ is diagonal (Eq. 3.18). When the input's amplitude is zero, the recurrent subspace generates two saddle points and two stable fixed points, as shown in Fig. 3.7 B. In this analysis, we focus on the limit case where the external input dominates the dynamics, $u_I \to \infty$, to build some intuition about how the dynamics evolve when the input is increased.

From the dynamics in Eq. (3.22), we can determine the fixed point equation:

$$\boldsymbol{\kappa_0} = \left\langle \phi' \left(0, u_I^2 + \boldsymbol{\kappa_0}^T \boldsymbol{\kappa_0}\right)\right\rangle \left(\boldsymbol{\sigma_{mn}} \boldsymbol{\kappa_0} + u_I \boldsymbol{\sigma_{nI}}\right) \tag{3.62}$$

In the case $u_I \to \infty$, the contribution of the external input to the dynamics is much larger than the contribution of the recurrent connectivity $u_I |\boldsymbol{\sigma_{nI}}| \gg |\boldsymbol{\sigma_{mn}} \boldsymbol{\kappa_0}|$. Therefore, we study that scenario assuming that there are no recurrent dynamics $\boldsymbol{\sigma_{mn}} = \mathbf{0}$. The possible fixed points must appear then along the direction spanned by $\boldsymbol{\sigma_{nI}}$. Introducing the ansatz $\boldsymbol{\kappa_0} = \rho_0 \boldsymbol{\sigma_{nI}}$, where we assume that the correlation vector $\boldsymbol{\sigma_{nI}}$ has unit norm, we obtain

$$\rho_0 = \left\langle \phi' \left(0, u_I^2 + \rho_0^2\right)\right\rangle u_I. \tag{3.63}$$

This expression for the fixed point is not symmetric: if ρ_0 is a solution of the fixed point equation, it does not imply that $-\rho_0$ is also a solution. In autonomous networks, the dynamic landscape always shows such a symmetry, which is disrupted by the tonic input. The left hand side of Eq. (3.63) is a straight line with unit slope. The right hand side corresponds to a function with maximum at $\rho_0 = 0$, that decays asymptotically to zero as $|\rho_0|$ is increased. Both functions cross at one single point.

The Jacobian around this fixed point reads

$$\nabla \mathbf{F}\left(\boldsymbol{\kappa_0}\right) = -\boldsymbol{I} + \left\langle \phi'''\left(0, \rho_0^2 + u_I^2\right)\right\rangle u_I^2 \boldsymbol{\sigma_{nI}} \boldsymbol{\sigma_{nI}}^T \tag{3.64}$$

which has eigenvalues -1 in all directions except for direction $\boldsymbol{\sigma_{nI}}$, where the eigenvalue is even more negative: $-1 + \left\langle \phi'''\left(0, \rho_0^2 + u_I^2\right)\right\rangle u_I^2$. The fixed point is therefore stable in all directions.

On the other hand, when $u_I = 0$, the network generates two stable fixed points and two saddle points. Therefore, there must be one or more critical values u_I^* at which the dynamics undergo a bifurcation: from a heteroclinic cycle to a single fixed point in the direction $\boldsymbol{\sigma_{nI}}$. The input amplitudes studied in Fig. 3.8 A-D correspond to the subcritical regime, where the input does not remove the existence of a heteroclinic cycle. However, it perturbs the heteroclinic cycle even for weak inputs, by displacing the five autonomous fixed points along the $\boldsymbol{\sigma_{nI}}$ direction.

3.8 Supplementary information

3.8.1 Appendix A: Responses to transient pulses in simplified network models

The dynamics of a rank-R network with one single Gaussian population receiving a delta pulse along an input pattern $\mathbf{I}$ correlated only with the left connectivity patterns $\mathbf{m}^{(\mathbf{r})}$ read

$$\tau \frac{d\boldsymbol{\kappa}}{dt} = -\boldsymbol{\kappa} + \left\langle \phi' \left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa}\right) \right\rangle \boldsymbol{\sigma_{mn}} \boldsymbol{\kappa} + \boldsymbol{\sigma_{mI}} \delta \left(t - t_0\right). \tag{3.65}$$

The R-dimensional vector $\boldsymbol{\sigma_{mI}}$ measures the correlation between the input pattern and the R left connectivity patterns. The network response to this input with an immediate change of state in neural trajectory, from $\boldsymbol{\kappa}\left(t - t_0\right)$ to $\boldsymbol{\kappa}\left(t - t_0\right) + \boldsymbol{\sigma_{mI}}$.

The dynamics of the same network to a delta pulse along an input pattern correlated only with the right connectivity patterns $\mathbf{n}^{(\mathbf{r})}$ are determined by

$$\tau \frac{d\boldsymbol{\kappa}}{dt} = -\boldsymbol{\kappa} + \left\langle \phi' \left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa} + \kappa_I^2\right) \right\rangle \left(\boldsymbol{\sigma_{mn}} \boldsymbol{\kappa} + \boldsymbol{\sigma_{nI}} \kappa_I\right) \tag{3.66}$$

$$\tau \frac{d\kappa_I}{dt} = -\kappa_I + \delta \left(t - t_0\right), \tag{3.67}$$

where $\boldsymbol{\sigma_{nI}}$ is an R-dimensional vector that determines the correlations between input pattern and right connectivity patterns. The response to the pulse in the recurrent subspace is no longer immediate, because it is low-pass filtered by the membrane time constant τ and is modulated by the average gain of neurons in the network, $\left\langle \phi' \left(0, \boldsymbol{\kappa}^T \boldsymbol{\kappa} + \kappa_I^2\right) \right\rangle$.

However, both of the input pulses above can elicit qualitatively similar responses when the correlation vectors $\boldsymbol{\sigma_{mI}}$ are $\boldsymbol{\sigma_{nI}}$ parallel to each other (Fig. 3.12). When the input pulse is correlated when the left connectivity patterns, a discontinuity appears in the time-dependent trace of the collective variables (Fig. 3.12 A), while when the input pattern is correlated with the right connectivity patterns, there is no discontinuity in the recurrent collective variables (Fig. 3.12 B). Nevertheless, both input patterns are able to send the neural trajectories from one stable state towards a different stable fixed point, generating equivalent trajectories for implementing a given neural computation.

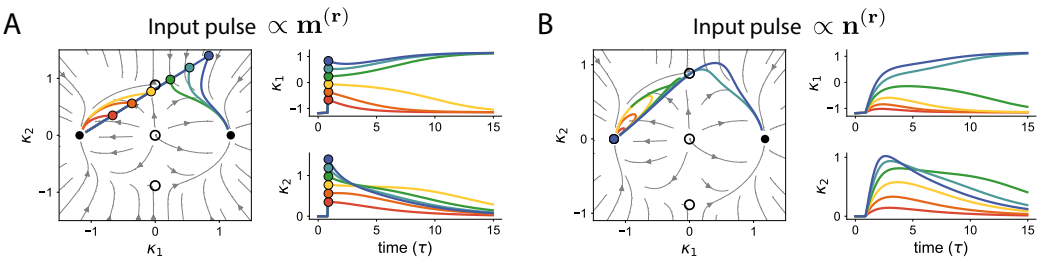

FIGURE 3.12: **Transient responses to input pulses in the recurrent subspace**. **A** Neural responses in a rank-two network to an input pulse proportional to the $\mathbf{m}^{(\mathbf{r})}$ connectivity patterns. Showing responses to six different pulses received along the same input pattern. Dots indicate network state right after the pulse is received. The pulse produces an immediate change in the network state, trajectories then evolve towards the slow manifold and evolve slowly towards a fixed point. Left: trajectories in the recurrent neural space. Right: activity of collective variables as a function of time. **B** Neural responses of a rank-two network to an input pulse received along an input pattern proportional to the $\mathbf{n}^{(\mathbf{r})}$ connectivity patterns. The input pulse shifts the neural trajectories to a third dimension because the input pattern is orthogonal to the recurrent subspace. The pulse does not generate an immediate change in state in the recurrent subspace. However, after a fast transient, trajectories are constrained to the recurrent subspace and generate responses which are qualitatively similar to those shown in **A**. Parameters: Rank-two network as in Fig. 3.7. In **A**, $\boldsymbol{\sigma_{mI}} = (1, 0.8)$, input strengths ranging from 0.5 to 2. In **B**, $\boldsymbol{\sigma_{nI}} = (1, 0.8)$, input strengths ranging from 0.5 to 3. Time in units of the membrane time constant τ.

3.8.2 Appendix B: Producing different time intervals on a spherical manifold

It is possible to implement temporal tasks using a spherical manifold, as shown in Fig. 3.10 for the Measure-Wait-Go task. A spherical manifold can be generated by a rank-three network with quasi-isotropic correlation between connectivity patterns. In order to find the input patterns that implement the required task, it is convenient to find first the neural states at the beginning of the production epoch that generate trajectories that evolve at different speeds, almost in parallel, towards a final state. Once such states are determined, together with their corresponding produced time intervals t_p, it is possible to find the right inputs that leave the neural trajectories at that state right at the beginning of the production epoch.

In the particular network we used to solve the Measure-Wait-Go task, there are two orthogonal ring manifolds on the surface of the sphere. One ring is the storage attractor, which is used as a line attractor to store estimated intervals. The orthogonal ring induces rotational dynamics along the sphere. In total, there are only two stable fixed point in the network. Therefore, during production, neural states must lie in the vicinity of one of the stable fixed points, and then move along the sphere towards the opposite stable state (Fig. 3.13 A). Trajectories that move from one side to the sphere to the other side closer to the plane of the storage attractor (blue curve, Fig. 3.13 B) evolve slower than trajectories that evolve further away from the storage attractor (red curve, Fig. 3.13 B). This is explained by the fact that the storage ring has very low speed, since it is designed to maintain a neural state over long periods of time. Neural states of the spherical manifold close to this plane are slower than neural states on the spherical manifold that are far from it (see color map in Fig. 3.13 B). This difference in speed along the sphere is the basis for the temporal stretching of neural responses.

Once a set of neural states has been found that evolve from one side of the sphere to the other at different speeds, it is possible to fix a readout direction and a threshold on the projected readout activity (Fig. 3.13 C) to determine the produced time interval t_p. That way, there is mapping between some neural state at the beginning of production and the timing response (Fig. 3.13 D). As a second step, in order to solve a given temporal task, we use those produced intervals as the sample intervals during the estimation epoch, and look for the right inputs that make the network reached those states at the beginning of production. This is the procedure we followed to implement the Measure-Wait-Go task. However, it would be possible to use the same trajectories during production to implement other timing tasks requiring estimating an interval or working memory, like the Reasy-Set-Go task or hte Cue-Set-Go task with a tonic input that does not remain present until the end of the trial. It would only be necessary to find the suitable input patterns that elicit the same neural trajectories during trajectories, without modifying the recurrent connectivity.

Finally, we can project the three-dimensional trajectories in the recurrent space to analyze their geometry (Fig. 3.13 E, left and right for two different projections). We find that along some projections (right), trajectories evolve in parallel along the surface of a sphere until reaching the final stable fixed point. The initial level on one given dimension of the recurrent subspace determines the speed at which trajectories evolve. This is the input subspace. In a projection orthogonal to this input subspace, the trajectories overlap almost completely along the same path, but evolve at different speeds, which defines the temporal scaling subspace.

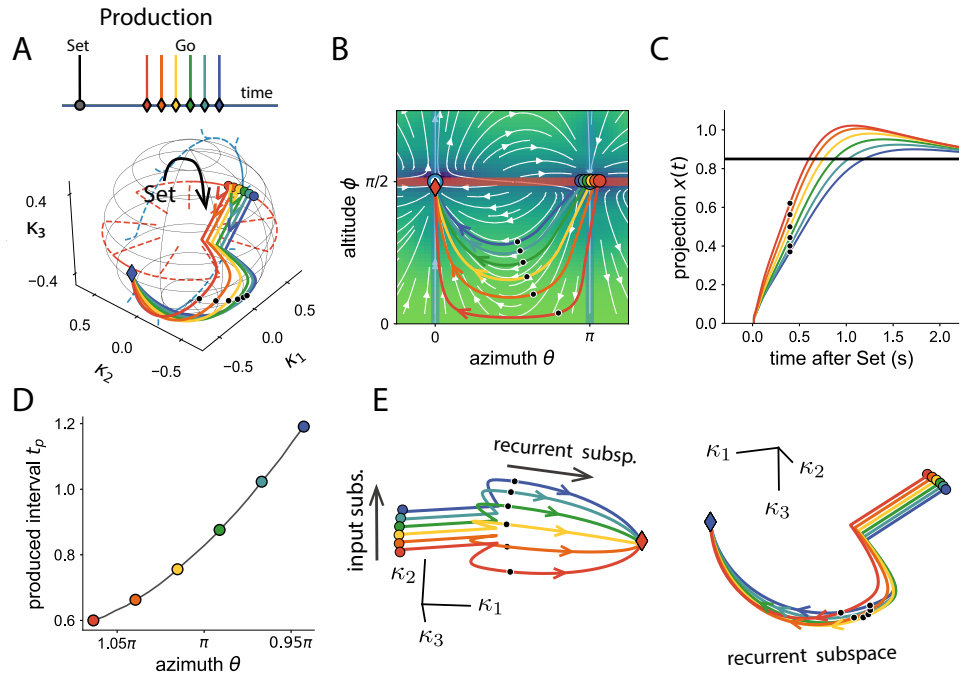

FIGURE 3.13: **Simplified model for production on spherical attractor**. **A** Sketch of the spherical manifold, with two embedded ring manifolds, the storage ring (red) and the production ring (blue). The trajectories during production are shown in thick color lines. The initial states are indicated with dots and the final states with diamonds. Black dots 500 ms after the Set input. Trajectories evolve along the bottom hemisphere, from one side of the storage attractor to the other. **B** Projected dynamics on the surface of the sphere, where each point on the sphere is parametrized by two angles, the altitude ϕ and azimuth θ. The colormap represents the logarithmic speed of the dynamics. Trajectories are also shown in this projection. **C** One-dimensional projection of the neural trajectories during the production epoch. The black line represents the threshold. The time point at which the projected activity crosses threshold corresponds to the produced interval t_p **D** Relation between initial state on the storage attractor, given by the azimuth θ and the corresponding produced interval. **E** Two different projections of the neural trajectories during the production epoch. Parameters: $\tau = 50$ ms, $\sigma_{m_1 n_1} = 1.8$, $\sigma_{m_2,n_2} = 1.4$, Set pulse: $\boldsymbol{\sigma_{mI}} = (1, 1)$. Input cue $\sigma_{n_2 I} = u_I$, $\sigma_{I^2} = u_I^2$, $\sigma_{n_1 I} = 0$, orientation of projection vector $\theta = 0.65\pi$, threshold value 0.12.

Bibliography

Abbott, L. F. (1994). Decoding neuronal firing and modelling neural networks. *Quarterly Reviews of Biophysics*, 27(3):291–331.

Aertsen, A. M. and Johannesma, P. I. (1981). The Spectro-Temporal Receptive Field - A functional characteristic of auditory neurons. *Biological Cybernetics*, 42(2):133–143.

Ahissar, E., Haidarliu, S., and Zacksenhouse, M. (1997). Decoding temporally encoded sensory input by cortical oscillations and thalamic phase comparators. *Proceedings of the National Academy of Sciences of the United States of America*, 94(21):11633–11638.

Ahmadian, Y., Rubin, D. B., and Miller, K. D. (2013). Analysis of the stabilized supralinear network. *Neural computation*, 25(8):1994–2037.

Aljadeff, J., Renfrew, D., and Stern, M. (2015a). Eigenvalues of block structured asymmetric random matrices. *Journal of Mathematical Physics*, 56(10):103502.

Aljadeff, J., Stern, M., and Sharpee, T. (2015b). Transition to chaos in random networks with cell-type-specific connectivity. *Physical Review Letters*, 114(8).

Amit, D. and Brunel, N. (1997). Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cerebral Cortex*, 7(3):237–252.

Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1987). Statistical mechanics of neural networks near saturation. *Annals of Physics*, 173(1):30–67.

Augustin, M., Ladenbauer, J., Baumann, F., and Obermayer, K. (2017). Low-dimensional spike rate models derived from networks of adaptive integrate-and-fire neurons: Comparison and implementation. *PLoS Computational Biology*, 13(6):1–46.

Augustin, M., Ladenbauer, J., and Obermayer, K. (2013). How adaptation shapes spike rate oscillations in recurrent neuronal networks. *Frontiers in Computational Neuroscience*, 7:9.

Bair, W. and Movshon, J. A. (2004). Adaptive Temporal Integration of Motion in Direction-Selective Neurons in Macaque Visual Cortex. *Journal of Neuroscience*, 24(33):7305–7323.

Barak, O. (2017). Recurrent neural networks as versatile tools of neuroscience research. *Current Opinion in Neurobiology*, 46:1–6.

Batchelor, A. M., Madge, D. J., and Garthwaite, J. (1994). Synaptic activation of metabotropic glutamate receptors in the parallel Fibre-Purkinje cell pathway in rat cerebellar slices. *Neuroscience*, 63(4):911–915.

Beiran, M., Dubreuil, A., Valente, A., Mastrogiuseppe, F., and Ostojic, S. (2020). Shaping dynamics with multiple populations in low-rank recurrent networks. *arXiv*.

Bellec, G., Salaj, D., Subramoney, A., Legenstein, R., and Maass, W. (2018). Long short-term memory and learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing Systems*, volume 2018-Decem, pages 787–797.

Benda, J. and Herz, A. V. M. (2003). A Universal Model for Spike-Frequency Adaptation. *Neural Computation*, 15(11):2523–2564.

Bengio, Y., Goodfellow, I., and Courville, A. (2017). *Deep learning*, volume 1. MIT press Massachusetts, USA:.

Bernacchia, A., Seo, H., Lee, D., and Wang, X. J. (2011). A reservoir of time constants for memory traces in cortical neurons. *Nature Neuroscience*, 14(3):366–372.

Berridge, M. J., Bootman, M. D., and Roderick, H. L. (2003). Calcium signalling: Dynamics, homeostasis and remodelling. *Nature Reviews Molecular Cell Biology*, 4(7):517–529.

Bi, Z. and Zhou, C. (2020). Understanding the computation of time using neural network models. *Proceedings of the National Academy of Sciences of the United States of America*, 117(19):10530–10540.

Bimbard, C., Ledoux, E., and Ostojic, S. (2016). Instability to a heterogeneous oscillatory state in randomly connected recurrent networks with delayed interactions. *Physical Review E*, 94(6):3–8.

Brown, D. A. (2000). M-Current: From Discovery to Single Channel Currents. In *Slow Synaptic Responses and Modulation*, pages 15–26. Springer Japan, Tokyo.

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

Brunel, N., Hakim, V., and Richardson, M. J. E. (2003). Firing-rate resonance in a generalized integrate-and-fire neuron with subthreshold resonance. *Physical Review E*, 67(5):051916.

Buonomano, D. V. and Maass, W. (2009). State-dependent computations: Spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, 10(2):113–125.

Carpenter, R. H. and Williams, M. (1995). Neural computation of log likelihood in control of saccadic eye movements. *Nature*, 377(6544):59–62.

Chaisangmongkon, W., Swaminathan, S. K., Freedman, D. J., and Wang, X. J. (2017). Computing by Robust Transience: How the Fronto-Parietal Network Performs Sequential, Category-Based Decisions. *Neuron*, 93(6):1504—-1517.e4.

Chen, T. W., Wardill, T. J., Sun, Y., Pulver, S. R., Renninger, S. L., Baohan, A., Schreiter, E. R., Kerr, R. A., Orger, M. B., Jayaraman, V., Looger, L. L., Svoboda, K., and Kim, D. S. (2013). Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458):295–300.

Churchland, M. M. and Shenoy, K. V. (2007). Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. *Journal of Neurophysiology*, 97(6):4235–4257.

Churchland, M. M., Yu, B. M., Ryu, S. I., Santhanam, G., and Shenoy, K. V. (2006). Neural variability in premotor cortex provides a signature of motor preparation. *Journal of Neuroscience*, 26(14):3697–3712.

Cohen, M. R. and Maunsell, J. H. (2009). Attention improves performance primarily by reducing interneuronal correlations. *Nature Neuroscience*, 12(12):1594–1600.

Cueva, C. J., Saez, A., Marcos, E., Genovesio, A., Jazayeri, M., Romo, R., Salzman, C. D., Shadlen, M. N., Fusi, S., Cueva, C. J., Fusi, S., Cueva, C. J., Saez, A., Marcos, E., Genovesio, A., Jazayeri, M., Romo, R., Salzman, C. D., Shadlen, M. N., and Fusi, S. (2020). Low dimensional dynamics for working memory and time encoding. *Proceedings of the National Academy of Sciences of the United States of America*, 117(37):1–20.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314.

Dombeck, D. A., Khabbaz, A. N., Collman, F., Adelman, T. L., and Tank, D. W. (2007). Imaging Large-Scale Neural Activity with Cellular Resolution in Awake, Mobile Mice. *Neuron*, 56(1):43–57.

Douglas Creelman, C. (1962). Human Discrimination of Auditory Duration. *Journal of the Acoustical Society of America*, 34(5):582–593.

Doya, K. (1993). Universality of Fully-Connected Recurrent Neural Networks. *Dept. of Biology, UCSD, Tech. Rep*, 1:1–6.

Drugowitsch, J., Moreno-Bote, R. N., Churchland, A. K., Shadlen, M. N., and Pouget, A. (2012). The cost of accumulating evidence in perceptual decision making. *Journal of Neuroscience*, 32(11):3612–3628.

Dubreuil, A., Valente, A., Beiran, M., Mastrogiuseppe, F., and Ostojic, S. (2020). Complementary roles of dimensionality and population structure in neural computations. *bioRxiv*, page 2020.07.03.185942.

Dummer, B., Wieland, S., and Lindner, B. (2014). Self-consistent determination of the spike-train power spectrum in a neural network with sparse connectivity. *Frontiers in Computational Neuroscience*, 8(September):1–12.

Dunlap, J. C. (1999). Molecular bases for circadian clocks. *Cell*, 96(2):271–290.

Egger, S. W., Le, N. M., and Jazayeri, M. (2020). A neural circuit model for human sensorimotor timing. *Nature Communications*, 11(1):3933.

Egger, S. W., Remington, E. D., Chang, C. J., and Jazayeri, M. (2019). Internal models of sensorimotor integration regulate cortical dynamics. *Nature Neuroscience*, 22(11):1871–1882.

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

Elsayed, G. F., Lara, A. H., Kaufman, M. T., Churchland, M. M., and Cunningham, J. P. (2016). Reorganization between preparatory and movement population responses in motor cortex. *Nature Communications*, 7(1):1–15.

Ermentrout, B., Pascal, M., and Gutkin, B. (2001). The effects of spike frequency adaptation and negative feedback on the synchronization of neural oscillators. *Neural Computation*, 13(6):1285–1310.

Fairhall, A. L., Lewen, G. D., Bialek, W., and De Ruyter van Steveninck, R. R. (2001). Efficiency and ambiguity in an adaptive neural code. *Nature*, 412(6849):787–792.

Fan, F., Xiong, J., and Wang, G. (2020). Universal approximation with quadratic deep networks. *Neural Networks*, 124:383–392.

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183–192.

Fusi, S., Miller, E. K., and Rigotti, M. (2016). Why neurons mix: High dimensionality for higher cognition. *Current Opinion in Neurobiology*, 37:66–74.

Gal, A., Eytan, D., Wallach, A., Sandler, M., Schiller, J., and Marom, S. (2010). Dynamics of Excitability over Extended Timescales in Cultured Cortical Neurons. *Journal of Neuroscience*, 30(48):16332–16342.

Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A., and Miller, L. E. (2020). Long-term stability of cortical population dynamics underlying consistent behavior. *Nature Neuroscience*, 23(2):260–270.

Gallego, J. A., Perich, M. G., Miller, L. E., and Solla, S. A. (2017). Neural Manifolds for the Control of Movement. *Neuron*, 94(5):978–984.

Gallego, J. A., Perich, M. G., Naufel, S. N., Ethier, C., Solla, S. A., and Miller, L. E. (2018). Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nature Communications*, 9(1):1–13.

Gao, P., Ganguli, S., Battaglia, F. P., and Schnitzer, M. J. (2015). On simplicity and complexity in the brave new world of large-scale neuroscience This review comes from a themed issue on Large-scale recording technology. *Current Opinion in Neurobiology*, 32:148–155.

Gardner, E. P. (1988). Somatosensory cortical mechanisms of feature detection in tactile and kinesthetic discrimination. *Canadian Journal of Physiology and Pharmacology*, 66(4):439–454.

Garthwaite, J. (1991). Glutamate, nitric oxide and cell-cell signalling in the nervous system. *Trends in Neurosciences*, 14(2):60–67.

Gerstner, W. and Kistler, W. M. (2002). Mathematical formulations of Hebbian learning. *Biological Cybernetics*, 87(5-6):404–415.

Goldman, M. S., Levine, J. H., Major, G., Tank, D. W., and Seung, H. S. (2003). Robust Persistent Neural Activity in a Model Integrator with Multiple Hysteretic Dendrites per Neuron. *Cerebral Cortex*, 13(11):1185–1195.

Gray, C. M., Maldonado, P. E., Wilson, M., and McNaughton, B. (1995). Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. *Journal of Neuroscience Methods*, 63(1-2):43–54.

Gross, C. G. (1992). Representation of visual stimuli in inferior temporal cortex. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 335(1273):3–10.

Gross, C. G., Rocha-Miranda, C. E., and Bender, D. B. (1972). Visual properties of neurons in inferotemporal cortex of the Macaque. *Journal of neurophysiology*, 35(1):96–111.

Grossberg, S. and Schmajuk, N. A. (1989). Neural dynamics of adaptive timing and temporal discrimination during associative learning. *Neural Networks*, 2(2):79–102.

Grothe, B., Pecka, M., and McAlpine, D. (2010). Mechanisms of Sound Localization in Mammals. *Physiological Reviews*, 90(3):983–1012.

Hanes, D. P. and Schall, J. D. (1996). Neural control of voluntary movement initiation. *Science*, 274(5286):427–430.

Hanks, T. D., Kiani, R., and Shadlen, M. N. (2014). A neural mechanism of speed-accuracy tradeoff in macaque area LIP. *eLife*, 2014(3).

Hardy, N. F. and Buonomano, D. V. (2016). Neurocomputational models of interval and pattern timing. *Current Opinion in Behavioral Sciences*, 8:250–257.

Hardy, N. F., Goudar, V., Romero-Sosa, J. L., and Buonomano, D. V. (2018). A model of temporal scaling correctly predicts that motor timing improves with speed. *Nature Communications*, 9(1):1–14.

Harish, O. and Hansel, D. (2015). Asynchronous Rate Chaos in Spiking Neuronal Circuits. *PLoS Computational Biology*, 11(7):e1004266.

Hebb, D. O. (1949). *The organization of behavior: a neuropsychological theory.* J. Wiley; Chapman & Hall.

Heiney, S. A., Wohl, M. P., Chettih, S. N., Ruffolo, L. I., and Medina, J. F. (2014). Cerebellar-dependent expression of motor learning during eyeblink conditioning in head-fixed mice. *Journal of Neuroscience*, 34(45):14845–14853.

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

Hennig, M. H. (2013). Theoretical models of synaptic short term plasticity. *Frontiers in Computational Neuroscience*, 7.

Henry, G. H., Dreher, B., and Bishop, P. O. (1974). Orientation specificity of cells in cat striate cortex. *Journal of Neurophysiology*, 37(6):1394–1409.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558.

Horio, Y. and Aihara, K. (2008). Analog computation through high-dimensional physical chaotic neuro-dynamics. *Physica D: Nonlinear Phenomena*, 237(9):1215–1225.

Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

Huang, C. and Doiron, B. (2017). Once upon a (slow) time in the land of recurrent neuronal networks…. *Current Opinion in Neurobiology*, 46:31–38.

Hubel, D. H. (1957). Tungsten microelectrode for recording from single units. *Science*, 125(3247):549–550.

Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3):574–591.

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

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13.

Jazayeri, M. and Shadlen, M. N. (2010). Temporal context calibrates interval timing. *Nature Neuroscience*, 13(8):1020–1026.

Jazayeri, M. and Shadlen, M. N. (2015). A Neural Mechanism for Sensing and Reproducing a Time Interval. *Current Biology*, 25(20):2599–2609.

Johnston and Wu (1995). *Principles of cellular neurophysiology*. MIT Press.

Kacelnik, A. and Brunner, D. (2002). Timing and foraging: Gibbon's scalar expectancy theory and optimal patch exploitation. *Learning and Motivation*, 33(1):177–195.

Kadmon, J. and Sompolinsky, H. (2015). Transition to chaos in random neuronal networks. *Physical Review X*, 5(4).

Karmarkar, U. R. and Buonomano, D. V. (2007). Timing in the Absence of Clocks: Encoding Time in Neural Network States. *Neuron*, 53(3):427–438.

Kaufman, M. T., Churchland, M. M., Ryu, S. I., and Shenoy, K. V. (2014). Cortical activity in the null space: Permitting preparation without movement. *Nature Neuroscience*, 17(3):440–448.

Kilavik, B., Confais, J., and Riehle, A. (2014). Signs of Timing in Motor Cortex During Movement Preparation and Cue Anticipation. *Advances in Experimental Medicine and Biology*, 829:121–142.

Killeen, P. R. and Fetterman, J. G. (1988). A Behavioral Theory of Timing. *Psychological Review*, 95(2):274–295.

Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.

Ko, H., Hofer, S. B., Pichler, B., Buchanan, K. A., Sjöström, P. J., and Mrsic-Flogel, T. D. (2011). Functional specificity of local synaptic connections in neocortical networks. *Nature*, 473(7345):87–91.

La Camera, G., Rauch, A., Thurbon, D., Lüscher, H.-R., Senn, W., and Fusi, S. (2006). Multiple Time Scales of Temporal Response in Pyramidal and Fast Spiking Cortical Neurons. *Journal of Neurophysiology*, 96(6):3448–3464.

Ladenbauer, J., Augustin, M., and Obermayer, K. (2013). How adaptation currents change threshold, gain and variability of neuronal spiking. *Journal of Neurophysiology*, 111(5):939–953.

Ladenbauer, J., Augustin, M., Shiau, L. J., and Obermayer, K. (2012). Impact of adaptation currents on synchronization of coupled exponential integrate-and-fire neurons. *PLoS Computational Biology*, 8(4):e1002478.

Laing, C. R. and Chow, C. C. (2002). A Spiking Neuron Model for Binocular Rivalry. *Journal of Computational Neuroscience*, 12(1):39–53.

Laje, R. and Buonomano, D. V. (2013). Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nature Neuroscience*, 16(7):925–933.

Lerchner, A., Sterner, G., Hertz, J., and Ahmadi, M. (2006). Mean field theory for a balanced hypercolumn model of orientation selectivity in primary visual cortex. *Network: Computation in Neural Systems*, 17(2):131–150.

Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867.

Lester, R. A., Clements, J. D., Westbrook, G. L., and Jahr, C. E. (1990). Channel kinetics determine the time course of NMDA receptor-mediated synaptic currents. *Nature*, 346(6284):565–567.

Litwin-Kumar, A. and Doiron, B. (2012). Slow dynamics and high variability in balanced cortical networks with clustered connections. *Nature Neuroscience*, 15(11):1498–1505.

Logiaco, L., Abbott, L., and Escola, S. (2019). A model of flexible motor sequencing through thalamic control of cortical dynamics. *bioRxiv*, page 2019.12.17.880153.

Ma, W. J., Husain, M., and Bays, P. M. (2014). Changing concepts of working memory. *Nature Neuroscience*, 17(3):347–356.

Maass, W., Joshi, P., and Sontag, E. D. (2007). Computational aspects of feedback in neural circuits. *PLoS Comput Biol*, 3(1):165.

Maass, W., Natschläger, T., and Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560.

Machens, C. K., Romo, R., and Brody, C. D. (2010). Functional, but not anatomical, separation of "what" and "when" in prefrontal cortex. *Journal of Neuroscience*, 30(1):350–360.

Maimon, G. and Assad, J. A. (2006). A cognitive signal for the proactive timing of action in macaque lip. *Nature neuroscience*, 9(7):948–955.

Mainen, Z. F. and Seinowski, T. J. (1995). Reliability of spike timing in neocortical neurons. *Science*, 268(5216):1503–1506.

Malapani, C. and Fairhurst, S. (2002). Scalar timing in animals and humans. *Learning and Motivation*, 33(1):156–176.

Manohar, S. G., Chong, T. T., Apps, M. A., Batla, A., Stamelou, M., Jarman, P. R., Bhatia, K. P., and Husain, M. (2015). Reward Pays the Cost of Noise Reduction in Motor and Cognitive Control. *Current Biology*, 25(13):1707–1716.

Mante, V., Sussillo, D., Shenoy, K. V., and Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84.

Markram, H., Wang, Y., and Tsodyks, M. (1998). Differential signaling via the same axon of neocortical pyramidal neurons. *Proceedings of the National Academy of Sciences*.

Martí, D., Brunel, N., and Ostojic, S. (2018). Correlations between synapses in pairs of neurons slow down dynamics in randomly connected neural networks. *Physical Review E*, 97(6):062314.

Mastrogiuseppe, F. and Ostojic, S. (2017). Intrinsically-generated fluctuating activity in excitatory-inhibitory networks. *PLoS Computational Biology*, 13(4):1–40.

Mastrogiuseppe, F. and Ostojic, S. (2018). Linking Connectivity, Dynamics, and Computations in Low-Rank Recurrent Neural Networks. *Neuron*, 99(3):609–623.e29.

Mastrogiuseppe, F. and Ostojic, S. (2019). A geometrical analysis of global stability in trained feedback networks. *Neural Computation*, 31(6):1139–1182.

Mauk, M. D. and Buonomano, D. V. (2004). The neural basis of temporal processing. *Annual Review of Neuroscience*, 27(1):307–340.

Mauk, M. D. and Donegan, N. H. (1997). A model of pavlovian eyelid conditioning based on the synaptic organization of the cerebellum. *Learning and Memory*, 4(1):130–158.

Mello, G. B., Soares, S., and Paton, J. J. (2015). A scalable population code for time in the striatum. *Current Biology*, 25(9):1113–1122.

Miall, C. (1989). The Storage of Time Intervals Using Oscillating Neurons. *Neural Computation*, 1(3):359–371.

Miyashita, Y. and Chang, H. S. (1988). Neuronal correlate of pictorial short-term memory in the primate temporal cortex. *Nature*, 331(6151):68–70.

Moore, J. W., Desmond, J. E., and Berthier, N. E. (1989). Adaptively timed conditioned responses and the cerebellum: A neural network approach. *Biological Cybernetics*, 62(1):17–28.

Murphy, B. K. and Miller, K. D. (2009). Balanced amplification: a new mechanism of selective amplification of neural activity patterns. *Neuron*, 61(4):635–48.

Murray, J. D., Bernacchia, A., Freedman, D. J., Romo, R., Wallis, J. D., Cai, X., Padoa-Schioppa, C., Pasternak, T., Seo, H., Lee, D., and Wang, X. J. (2014). A hierarchy of intrinsic timescales across primate cortex. *Nature Neuroscience*, 17(12):1661–1663.

Muscinelli, S. P., Gerstner, W., and Schwalger, T. (2019). How single neuron properties shape chaotic dynamics and signal transmission in random neural networks. *PLoS Computational Biology*, 15(6).

Nakatsukasa, Y. (2019). The low-rank eigenvalue problem. *arXiv*, page 1905.11490.

Naud, R. and Gerstner, W. (2012). Coding and Decoding with Adapting Neurons: A Population Approach to the Peri-Stimulus Time Histogram. *PLoS Computational Biology*, 8(10):e1002711.

Naud, R., Marcille, N., Clopath, C., and Gerstner, W. (2008). Firing patterns in the adaptive exponential integrate-and-fire model. *Biological Cybernetics*, 99(4-5):335–347.

Newberry, N. R. and Nicoll, R. A. (1984). Direct hyperpolarizing action of baclofen on hippocampal pyramidal cells. *Nature*, 308(5958):450–452.

Ni, A. M., Ruff, D. A., Alberts, J. J., Symmonds, J., and Cohen, M. R. (2018). Learning and attention reveal a general relationship between population activity and behavior. *Science*, 359(6374):463–465.

Nicola, W. and Clopath, C. (2017). Supervised learning in spiking neural networks with FORCE training. *Nature Communications*, 8(1):2208.

Nobre, A. C. and Van Ede, F. (2018). Anticipated moments: Temporal structure in attention. *Nature Reviews Neuroscience*, 19(1):34–48.

Ostojic, S. (2014). Two types of asynchronous activity in networks of excitatory and inhibitory spiking neurons. *Nature Neuroscience*, 17(4):594–600.

Ostojic, S. and Brunel, N. (2011). From Spiking Neuron Models to Linear-Nonlinear Models. *PLoS Computational Biology*, 7(1):e1001056.

Panichello, M. F., DePasquale, B., Pillow, J. W., and Buschman, T. J. (2019). Error-correcting dynamics in visual working memory. *Nature Communications*, 10(1).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., Facebook, Z. D., Research, A. I., Lin, Z., Desmaison, A., Antiga, L., Srl, O., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems*, pages 8024–8035.

Paton, J. J. and Buonomano, D. V. (2018). The neural basis of timing: Distributed mechanisms for diverse functions. *Neuron*, 98(4):687–705.

Paulin, M. G. (2004). Neural Engineering: Computation, Representation and Dynamics in Neurobiological Systems. *Neural Networks*, 17(3):461–463.

Pinto, L., Rajan, K., DePasquale, B., Thiberge, S. Y., Tank, D. W., and Brody, C. D. (2019). Task-Dependent Changes in the Large-Scale Dynamics and Necessity of Cortical Regions. *Neuron*, 104(4):810–824.e9.

Pollock, E. and Jazayeri, M. (2020). Engineering recurrent neural networks from task-relevant manifolds and dynamics. *PLOS Computational Biology*, 16(8):e1008128.

Pozzorini, C., Naud, R., Mensi, S., and Gerstner, W. (2013). Temporal whitening by power-law adaptation in neocortical neurons. *Nature Neuroscience*, 16(7):942–948.

Rabinovich, M., Huerta, R., and Laurent, G. (2008). Neuroscience: Transient dynamics for neural processing. *Science*, 321(5885):48–50.

Rabinovich, M. I., Simmons, A. N., and Varona, P. (2015). Dynamical bridge between brain and mind. *Trends in cognitive sciences*, 19(8):453–461.

Rabinovich, M. I., Varona, P., Selverston, A. I., and Abarbanel, H. D. (2006). Dynamical principles in neuroscience. *Reviews of Modern Physics*, 78(4):1213.

Rajan, K. and Abbott, L. F. (2006). Eigenvalue spectra of random matrices for neural networks. *Physical Review Letters*, 97(18):2–5.

Rajan, K., Abbott, L. F., and Sompolinsky, H. (2010). Stimulus-dependent suppression of chaos in recurrent neural networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 82(1):1–5.

Rajan, K., Harvey, C. D. D., and Tank, D. W. W. (2016). Recurrent Network Models of Sequence Generation and Memory. *Neuron*, 90(1):128–142.

Ramón y Cajal, S. (1909). *Histologie du système nerveux de l'homme & des vertébrés.* Maloine, Paris.

Remington, E. D., Egger, S. W., Narain, D., Wang, J., and Jazayeri, M. (2018a). A Dynamical Systems Perspective on Flexible Motor Timing. *Trends in Cognitive Sciences*, 22(10):938–952.

Remington, E. D., Narain, D., Hosseini, E. A., and Jazayeri, M. (2018b). Flexible Sensorimotor Computations through Rapid Reconfiguration of Cortical Dynamics. *Neuron*, 98(5):1005—-1019.e5.

Richardson, M. J. E., Brunel, N., and Hakim, V. (2003). From Subthreshold to Firing-Rate Resonance. *Journal of Neurophysiology*, 89(5):2538–2554.

Rigotti, M., Barak, O., Warden, M. R., Wang, X. J., Daw, N. D., Miller, E. K., and Fusi, S. (2013). The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451):585–590.

Rivkind, A. and Barak, O. (2017). Local Dynamics in Trained Recurrent Neural Networks. *Physical Review Letters*, 118(25):258101.

Roitman, J. D. and Shadlen, M. N. (2002). Response of Neurons in the Lateral Intraparietal Area during a Combined Visual Discrimination Reaction Time Task. *The Journal of Neuroscience*, 22(21):9475–9489.

Romo, R., Brody, C. D., Hernández, A., and Lemus, L. (1999). Neuronal correlates of parametric working memory in the prefrontal cortex. *Nature*, 399(6735):470–473.

Romo, R. and Schultz, W. (1987). Neuronal activity preceding self-initiated or externally timed arm movements in area 6 of monkey cortex. *Experimental Brain Research*, 67(3):656–662.

Rosenblatt, F. (1962). *Principles of neurodynamics;perceptrons and the theory of brain mechanisms*. Washington,.

Russo, A. A., Bittner, S. R., Perkins, S. M., Seely, J. S., London, B. M., Lara, A. H., Miri, A., Marshall, N. J., Kohn, A., Jessell, T. M., Abbott, L. F., Cunningham, J. P., and Churchland, M. M. (2018). Motor Cortex Embeds Muscle-like Commands in an Untangled Population Response. *Neuron*, 97(4):953–966.e8.

Safaie, M., Jurado-Parras, M. T., Sarno, S., Louis, J., Karoutchi, C., Petit, L. F., Pasquet, M. O., Eloy, C., and Robbe, D. (2020). Turning the body into a clock: Accurate timing is facilitated by simple stereotyped interactions with the environment. *Proceedings of the National Academy of Sciences of the United States of America*, 117(23):13084–13093.

Saxena, S. and Cunningham, J. P. (2019). Towards the neural population doctrine. *Current Opinion in Neurobiology*, 55:103–111.

Schuecker, J., Goedeke, S., and Helias, M. (2018). Optimal Sequence Memory in Driven Random Networks. *Physical Review X*, 8(4):041029.

Schuessler, F., Dubreuil, A., Mastrogiuseppe, F., Ostojic, S., and Barak, O. (2020a). Dynamics of random recurrent networks with correlated low-rank structure. *Physical Review Research*, 2(1):13111.

Schuessler, F., Mastrogiuseppe, F., Dubreuil, A., Ostojic, S., and Barak, O. (2020b). The interplay between randomness and structure during learning in RNNs. *arXiv*, page 2006.11036.

Schwalger, T., Fisch, K., Benda, J., and Lindner, B. (2010). How noisy adaptation of neurons shapes interspike interval histograms and correlations. *PLoS Computational Biology*, 6(12):e1001026.

Schwalger, T. and Lindner, B. (2013). Patterns of interval correlations in neural oscillators with adaptation. *Frontiers in Computational Neuroscience*, 7:164.

Semedo, J. D., Zandvakili, A., Machens, C. K., Yu, B. M., and Kohn, A. (2019). Cortical Areas Interact through a Communication Subspace. *Neuron*, 102(1):249–259.e4.

Sjöström, P. J., Turrigiano, G. G., and Nelson, S. B. (2001). Rate, Timing, and Cooperativity Jointly Determine Cortical Synaptic Plasticity. *Neuron*, 32(6):1149–1164.

Smith, P. L. and Ratcliff, R. (2004). Psychology and neurobiology of simple decisions. *Trends in Neurosciences*, 27(3):161–168.

Sohn, H., Narain, D., Meirhaeghe, N., and Jazayeri, M. (2019). Bayesian Computation through Cortical Latent Dynamics. *Neuron*, 103(5):934—-947.e5.

Sompolinsky, H., Crisanti, A., and Sommers, H. J. (1988). Chaos in random neural networks. *Physical Review Letters*, 61(3):259–262.

Stanley, D. A., Bardakjian, B. L., Spano, M. L., and Ditto, W. L. (2011). Stochastic amplification of calcium-activated potassium currents in Ca2+ microdomains. *Journal of Computational Neuroscience*, 31(3):647–666.

Steinmetz, N. A., Koch, C., Harris, K. D., and Carandini, M. (2018). Challenges and opportunities for large-scale electrophysiology with Neuropixels probes. *Current Opinion in Neurobiology*, 50:92–100.

Stern, M., Sompolinsky, H., and Abbott, L. F. (2014). Dynamics of random neural networks with bistable units. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 90(6):1–7.

Strogatz, S. (2000). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering*. Westview Press.

Sussillo, D. (2014). Neural circuits as computational dynamical systems. *Current Opinion in Neurobiology*, 25:156–163.

Sussillo, D. and Abbott, L. (2009). Generating Coherent Patterns of Activity from Chaotic Neural Networks. *Neuron*, 63(4):544–557.

Sussillo, D. and Barak, O. (2013). Opening the Black Box: Low-Dimensional Dynamics in High-Dimensional Recurrent Neural Networks. *Neural Computation*, 25(3):626–649.

Sussillo, D., Churchland, M. M., Kaufman, M. T., and Shenoy, K. V. (2015). A neural network that finds a naturalistic solution for the production of muscle activity. *Nature Neuroscience*, 18(7):1025–1033.

Tchumatchenko, T., Malyshev, A., Wolf, F., and Volgushev, M. (2011). Ultrafast Population Encoding by Cortical Neurons. *Journal of Neuroscience*, 31(34):12171–12179.

Tee, J. and Taylor, D. P. (2018). Is Information in the Brain Represented in Continuous or Discrete Form? *arXiv*.

Thorpe, S., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, 381(6582):520–522.

Thura, D. and Cisek, P. (2016). Modulation of premotor and primary motor cortical activity during volitional adjustments of speed-accuracy trade-offs. *Journal of Neuroscience*, 36(3):938–956.

Todd, N. P. A., Lee, C. S., and O'Boyle, D. J. (2002). A sensorimotor theory of temporal tracking and beat induction. *Psychological Research*, 66(1):26–39.

Troyer, T. W. and Miller, K. D. (1997). Physiological Gain Leads to High ISI Variability in a Simple Model of a Cortical Regular Spiking Cell. *Neural Computation*, 9(5):971–983.

Vyas, S., Golub, M. D., Sussillo, D., and Shenoy, K. V. (2020). Computation through neural population dynamics. *Annual Review of Neuroscience*, 43:249–275.

Wang, J., Narain, D., Hosseini, E. A., and Jazayeri, M. (2018). Flexible timing by temporal scaling of cortical responses. *Nature Neuroscience*, 21(1):102–112.

Wang, X. J. (2001). Synaptic reverberation underlying mnemonic persistent activity. *Trends in Neurosciences*, 24(8):455–463.

Wang, X. J. (2008). Decision Making in Recurrent Neuronal Circuits. *Neuron*, 60(2):215–234.

Werbos, P. J. (1990). Backpropagation Through Time: What It Does and How to Do It. *Proceedings of the IEEE*, 78(10):1550–1560.

Wieland, S., Bernardi, D., Schwalger, T., and Lindner, B. (2015). Slow fluctuations in recurrent networks of spiking neurons. *Phys. Rev. E*, 92:040901(R).

Wiggins, S. (2003). *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. Springer-Verlag.

Wilson, H. R. and Cowan, J. D. (1972). Excitatory and Inhibitory Interactions in Localized Populations of Model Neurons. *Biophysical Journal*, 12(1):1–24.

Wilson, H. R. and Cowan, J. D. (1973). A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik*, 13(2):55–80.

Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T., and Wang, X.-J. (2019). Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306.

Yin, H. H. (2014). Action, time and the basal ganglia. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1637).

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., and Sahani, M. (2009). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, 102(1):614–635.

Yuste, R. (2015). From the neuron doctrine to neural networks. *Nature Reviews Neuroscience*, 16(8):487–497.

Zucker, R. S. and Regehr, W. G. (2002). Short-Term Synaptic Plasticity. *Annual Review of Physiology*, 64(1):355–405.

RÉSUMÉ

L'activité neuronale chez l'animal présente une vaste gamme d'échelles temporelles qui donne lieu à des comportements pouvant s'adapter à un environnement en constante évolution. Comment ces motifs temporels complexes sont-ils générés, sachant que les neurones individuels fonctionnent avec une constante de temps de membrane de l'ordre de quelques dizaines de millisecondes ? Comment les opérations neuronales s'appuient-elles sur ces motifs d'activité pour produire des comportements temporels flexibles ?
Une des hypothèses possibles pour l'émergence de dynamiques lentes au niveau de l'activité neuronale est que celles-ci soient héritées de processus biophysiques sous-jacents au niveau des neurones individuels, tels que les courants ioniques d'adaptation et la transmission synaptique. Dans la première partie de cette thèse, nous analysons des réseaux de neurones connectés de façon aléatoire qui prennent en compte ces processus cellulaires lents et nous caractérisons les statistiques temporelles de l'activité neuronale émergente. Notre conclusion principale est que les échelles temporelles des différents processus biophysiques n'entraînent pas nécessairement une grande variété d'échelles temporelles dans l'activité collective des réseaux de neurones.
D'autre part, des motifs d'activité complexes peuvent être générés par des structures spécifiques de connectivité synaptique. Dans le deuxième chapitre de cette thèse, nous considérons une nouvelle classe de modèles, des réseaux récurrents de bas rang à mixture de gaussiennes. La structure de la connectivité y est caractérisée par deux propriétés indépendantes : le rang de la matrice de connectivité, et le nombre de populations définies par les statistiques de la connectivité. Nous montrons que ces réseaux agissent comme des approximateurs universels des systèmes dynamiques et peuvent en conséquence générer des activités temporelles complexes.
Dans le dernier chapitre, nous étudions les mécanismes dynamiques au niveau de réseaux de neurones à la base de tâches sensorielles et motrices qui exigent des calculs temporels flexibles. On montre d'abord que des réseaux de bas rang entraînés sur ces tâches donnent lieu à des variétés invariantes de basse dimensionnalité, où la dynamique évolue lentement et peut être modulée de manière flexible. Nous identifions ensuite les composantes dynamiques clés et les validons dans des modèles de réseaux simplifiés qui effectuent les mêmes tâches temporelles. Globalement, nous avons découvert de nouveaux mécanismes dynamiques générant des comportements temporels flexibles, qui sont fondés sur une structure de connectivité minimale et peuvent implémenter une ample gamme de tâches.

MOTS CLÉS

réseaux de neurones récurrents, dynamique de réseaux, variétés, codage temporel, neuroscience théorique

ABSTRACT

Neural activity in awake animals exhibits a vast range of timescales giving rise to behavior that can adapt to a constantly evolving environment. How are such complex temporal patterns generated in the brain, given that individual neurons function with membrane time constants in the range of tens of milliseconds? How can neural computations rely on such activity patterns to produce flexible temporal behavior?
One hypothesis posits that long timescales at the level of neural network dynamics can be inherited from long timescales of underlying biophysical processes at the single neuron level, such as adaptive ionic currents and synaptic transmission. We analyzed large networks of randomly connected neurons taking into account these slow cellular process, and characterized the temporal statistics of the emerging neural activity. Our overarching result is that the timescales of different biophysical processes do not necessarily induce a wide range of timescales in the collective activity of large recurrent networks.
Conversely, complex temporal patterns can be generated by structure in synaptic connectivity. In the second chapter of the dissertation, we considered a novel class of models, Gaussian-mixture low-rank recurrent networks, in which connectivity structure is characterized by two independent properties, the rank of the connectivity matrix and the number of statistically-defined populations. We show that such networks act as universal approximators of arbitrary low-dimensional dynamical systems, and therefore can generate temporally complex activity.
In the last chapter, we investigated how dynamical mechanisms at the network level implement flexible sensorimotor timing tasks. We first show that low-rank networks trained on such tasks generate low-dimensional invariant manifolds, where dynamics evolve slowly and can be flexibly modulated. We then identified the core dynamical components and tested them in simplified network models that carry out the same flexible timing tasks. Overall, we uncovered novel dynamical mechanisms for temporal flexibility that rely on minimal connectivity structure and can implement a vast range of computations.

KEYWORDS

recurrent neural network, network dynamics, manifolds, timing, theoretical neuroscience