university of groningen

faculty of science
and engineering

Stochastic State Switching in Attractor Neural Networks

Master's Thesis

July 2025

Student: Ruben Hendriks

First supervisor (mathematics): dr. Réka Szabó

First supervisor (physics): prof. dr. Elisabetta Chicca

Daily supervisor: dr. Madison Cotteret

Stochastic State Switching in Attractor Neural Networks

Ruben Hendriks

Abstract

The CogniGron institute at the University of Groningen aims to mimic the computational abilities and energy efficiency of the brain in hardware. Converging evidence from neuroscience suggests that during computation and decision making, metastable stochastic switching between attractor states occurs in the brain. In this thesis, a minimal generalization of a stochastic two-pattern Hopfield network and a stochastic sparse block attractor network are shown to support controllable metastable stochastic attractor state switching. These networks are then used as building blocks to construct a network that can emulate an arbitrary embedded two-state Markov chain, and as a consequence, any N-state Markov chain, with its attractor state dynamics. The Hopfield implementation results in a malfunctioning embedded Markov chain due to the dense activity of Hopfield states. The sparse block network supports a successful embedding of a Markov chain, effectively demonstrating robust, controllable, multi-timescale state-dependent computing in sparse attractor networks. The generalized Hopfield network is rigorously analyzed with statistical mechanics theory. A transformation between Hopfield- and multi-group Curie-Weiss networks has been developed, which is used to analytically derive the phase diagram of the generalized Hopfield network and its key metastable properties.

Contents

1 Introduction

The brain is arguably the most complex object in the known Universe. With its 86 billion neurons and 100 trillion synapses, it is able to operate and make decisions in highly complex and changing environments. While a single neuron is able to perform simple operations, a very large and highly connected ensemble of them can perform complex tasks and computations. The massive size of brain circuits suggests that network size is fundamental to brain computation [1]. Long-term changes in synapses provide the basis for learning and memory, while short-term changes support a variety of computations [2]: a single individual part of brain 'hardware' can take part in multiple aspects of brain function.

In comparison, a conventional computer has the Von Neumann architecture. These computers consist of a central processing unit (CPU) which performs computations, and a separate memory in which data is stored. Computations are performed in series, and at fixed time intervals. A large bottleneck of this architecture is the fact that data needs to be transported from memory to CPU and back. As a consequence, doing large amounts of computations with such computers can become very costly in time and energy. In a world where energy becomes more scarce, and in which there is ever-increasing demand for computational power, it is key to find alternative forms of computation which are more energy efficient.

We look to biology to find inspiration for solutions. The brain is not only a computational powerhouse, it is also extremely energy efficient. It can perform learning tasks that require an exaflop of computations, while only consuming 20 watts of power (roughly the power used by a lightbulb). In comparison, the OLCF-5 supercomputer (second fastest computer in the world at time of writing) reached one exaflop of computation while consuming 21 megawatts (roughly the power used by 15.000 homes) [3]. But why is the brain so much more energy efficient? The energy efficiency of the brain is believed to lie in the way it organizes computation among its components [4]. So, to solve our energy problem, all we need to do is to understand the architecture of brain computation, which comes down to reverse engineering the most complex object in the Universe!

It is however limiting to motivate brain research from the perspective of information processing by its promises for energy efficient computing alone. Understanding the information processing abilities of the brain has more applications, for example brain diseases [5] [6], motor control [7] [8], and cognition [9].

In this thesis, we focus on a bottom-up approach to understanding brain function, and in particular the mechanisms of memory storage and retrieval. Not only do brains recall memories from input, they also stochastically explore the memory space, for example during decision making [10] [11]; this is an example of high-level brain function. In the spirit of physicist Richard Feynman: "What I can't create, I don't understand", we aim to mimic this phenomenon of high level brain function by making minimal generalizations to existing models of low-level brain function. Furthermore, we will construct a mathematical framework to analyze this generalized model rigorously, using theory from statistical mechanics. We first present basic concepts in neural computation, and the statistical mechanics models of large neuron ensembles used in this thesis. After that, we introduce Markov chains, which is the high level framework of computation we want to implement into the introduced models of large neuron ensembles. An overview of the thesis structure is given at the end of this Chapter.

1.1 Biological background

1.1.1 The neuron

The most elementary building blocks of the brain are *neurons* and *synapses.* There are many varieties of neurons, which can have specialized structure and function, but for our purposes it is enough to describe the general parts that make up a neuron in the human nervous system. By only considering the basic elements that most neurons share, we have a starting point for making a model of neural circuits.

A neuron consists of *dendrites*, which receives signals, a *cell body* (or *soma*), which accumulates the incoming signals, and an *axon*, which sends signals. The axon branches into terminals so that it can connect to multiple other neurons [12]. Synapses are structures that connect a *pre-synaptic neuron* (a neuron that sends a signal) to a *post-synaptic neuron* (a neuron that receives a signal). It is located between a terminal of the axon of a pre-synaptic neuron, and the dendrite or soma of a post-synaptic neuron.

There are two types of synapse: *excitatory* and *inhibitory.* An excitatory synapse will excite a post-synaptic neuron (make it more likely to fire) when it receives a signal, and an inhibitory synapse will suppress the post-synaptic neuron (make it less likely to fire) when it receives a signal. On average, a neuron receives input from ten thousand other neurons; they are highly connected.

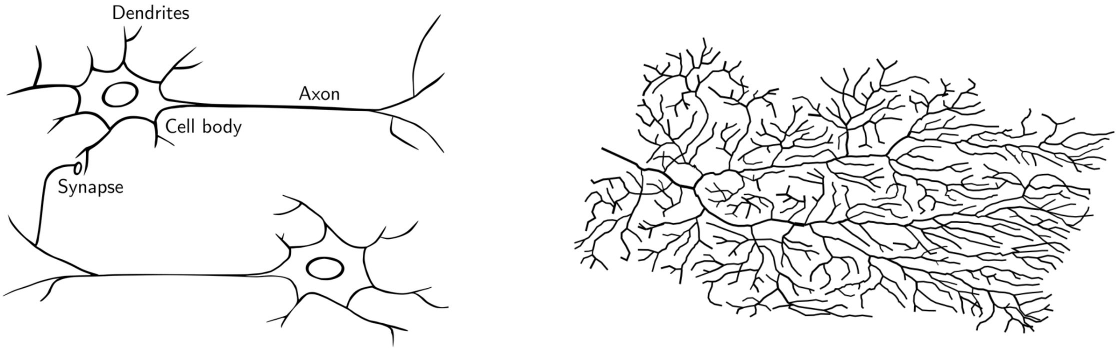


Figure 1: Left: Basic schematic of a neuron. Figure from [13]. Right: Dendritic tree of a Purkinje cell, a large neuron found in the cerebellum. Figure from [14].

The dynamics of neurons is as follows [15].

1. A neuron is either 'firing' or 'silent'. If a neuron is firing, a stable electric signal (called a *spike*, or sometimes *action potential*) propagates through its axon. The signal is duplicated at each branching point of the axon. If a neuron is not firing, there is no signal, and the axon is at its resting potential.
2. When the signal arrives at a synapse, neurotransmitters are passed across the synapse.
3. The neurotransmitters arrive at the membrane of the post-synaptic neuron, and attach to receptors. The receptors unlock and allow a signal to be passed. This signal is called the post-synaptic potential (PSP).
4. The PSP diffuses through the dendrites of the post-synaptic neuron into the soma. The PSP can be excitatory (depolarizing the membrane), making the post-synaptic neuron more likely to fire, or inhibitory (hyper-polarizing the membrane), making the post-synaptic neuron less likely to fire.
5. If the total sum of PSPs that enter the soma is large enough to surpass a certain threshold, the post-synaptic neuron will fire with high probability.

1.1.2 Attractor states

A large network of neurons, connected to each other with excitatory and inhibitory synapses, thus forms a dynamical system with the dynamics as described above. We make a simplification and say that at each moment in time, any neuron is either firing or silent. We further assume that the network does not get any external input or stimulus during its dynamics; it evolves on its own.

The *state* of this dynamical system at a given time is the collective activity (sometimes called *population activity*) of all neurons in the network at that time. The state is a point in the *state space* of the system. An *attractor* of a dynamical system is the minimal set of states in a state space, to which all nearby states flow in time [16]. In neural networks, these attractor states are stable patterns of neuron activity, to which the network converges over time. Neural networks that contain attractor states are called *attractor neural networks* (ANNs).

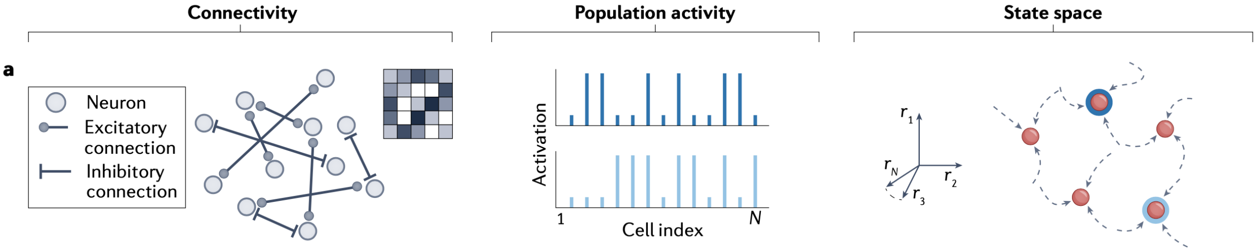


Figure 2: Left: Example of a network of neurons and their connections. Middle: Example of stable states of neuron activity. Right: Example of a state space of the dynamical system. Grey lines are trajectories, red dots are attracting states. Figure from [17].

There is good evidence that many cognitive functions, ranging from motor function to memory, depend on attractor states [10], and that they are essential for their functioning. Attractors are found in networks of head-direction cells of mammals [18], motor neurons that control eye position of monkeys [19] and associative memory of macaques [20]. For good reviews of attractor networks in neurobiology, see [17], [21].

In this thesis we focus on neural networks that function as *associative memory.* The idea of associative memory is that when an input (a stimulus, like an image or smell) is presented, the stored memory that is most like the input is recalled [22]. The stored memories are much like attractor states: if we start the neural network in a state that resembles a nearby attractor state, the network will 'recall' (converge to) the attractor state. Associative memory can thus very naturally be modeled by ANNs. We will present the Hopfield model [23], one of the first models of associative memory, in more detail in Section 1.3. Furthermore, in Section 1.4 we will present a more modern and biologically plausible [24] 'sparse' (fewer neurons firing at the same time) ANN. Both networks will play an important role in our analyses.

1.1.3 State-switching in the brain

A drawback of attractor networks is that once the network dynamics has converged to the attractor state, the network will stay in this attractor state indefinitely. There is increasing evidence that in biological neural networks, there are transitions between different relatively stable states of population activity of neurons[10]; the population activity will not stay in an attractor state indefinitely. Attractor states in which the dynamical system only stays temporarily are called *metastable* states. Metastability in neural networks has received increasing attention for its role in brain function and computation [25] [26]. Such transitions between attractors have also been observed experimentally: in cortical activity of monkeys during decision making [11], EEG microstates of humans [27] and cortical activity during taste processing [28].

Interestingly, transitions between attractor states occur even in the absence of stimuli [10][25]. The behaviour of such neuronal dynamics is distilled from neuron activity data using hidden Markov modeling (HMM) (Figure 3).

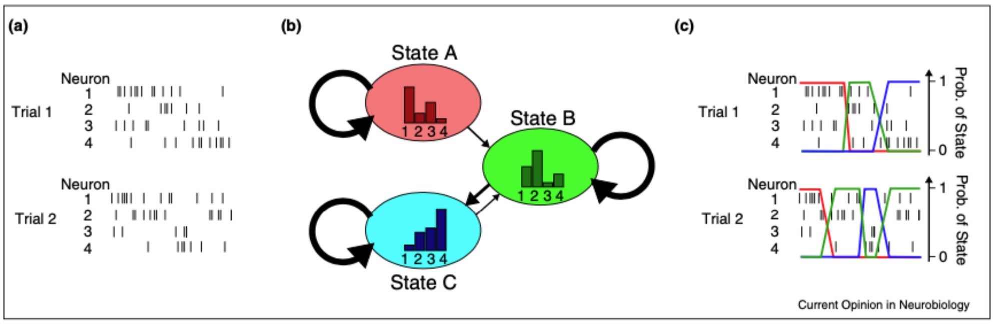


Figure 3: Hidden Markov modeling. (a) HMM is done for datasets of multiple simultaneously recorded spike trains. (b) A state (coloured circle) is defined as a probability distribution of spiking activity for each neuron (histogram in coloured circle). The probability of spiking in a small time bin is directly related to the firing rate. (1) Given a model and some data, the likelihood of the model producing the data is computed. (2) The firing probabilities are recalculated by fitting to the data. Repeating steps (1) and (2) optimizes the log-likelihood ratio. (c) The optimized model prescribes probabilities to each state at all timesteps, here plotted with coloured lines. Figure from [10].

HMM is used on datasets to find underlying distinct states and transitions between these states. In neuroscience, spike trains from individual neurons are recorded. A Markov chain is then fitted to the training data: different spiking activity from groups of neurons are bundled together into different states, and the transition probabilities between states are inferred from the data. After fitting with training data, the HMM can be used to discover transition sequences of states in new spike trains data.

The fact that HMM has been successful in analyzing time series of biological neural network activity suggests that there really might be Markovian dynamics 'embedded' in such networks. The fundamental components of a neural network are the neurons and synapses, but computation might actually be done at a higher level than that of the individual components. The success of HMM suggests that the fundamental units of information representations instead are the attractor states, and computation is performed in a higher conceptual framework, namely an embedded Markov chain, where the chain states correspond to attractor states in the network. The idea that biological neural networks operate as a state machine (which is a deterministic variant of a Markov chain) is supported by recent findings in neuroscience [29]. The idea of using neuron ensemble activity as fundamental representations for computation is explored in [1], [30].

1.2 Statistical mechanics

The function of biological neural networks comes most likely from their *emergent* properties [31]. To create mathematical models of emergent phenomena, we use the well-established field of statistical mechanics. It is a mathematical theory that was born more than a century ago, with the goal of explaining phenomena in thermodynamics starting from a microscopic setting. One century later, the theory has been used in almost all of science to explain all kinds of macroscopic phenomena, starting from a microscopic description. Considering that we have a microscopic description of neural networks (the mechanics of neurons), and that we're interested in macroscopic phenomena (emergent properties, like memory recall or computation), statistical mechanics is a very natural setting to work in. Furthermore, it is a *mathematical* theory: it is rigorous enough to develop definitions and to prove theorems about our models.

Statistical mechanics theory combines the available microscopic information and methods in probability theory to reveal macroscopic phenomena in the system of interest [32]. The first 'realistic' statistical mechanics model with nontrivial macroscopic behaviour is the famous Ising model. The model describes small magnets interacting with each other, and these magnets, when aligned, create a macroscopic magnetic field. The magnets sit on a lattice, and each magnet is influenced in its behaviour by its neighbors only; this complicates the analysis of the model. A *mean-field* version (removing geometry) of the Ising model is the Curie-Weiss model, in which every small magnet interacts with every other small magnet in the model. This mean-field model still has nontrivial macroscopic behaviour, and it will be the starting point for understanding the other models used in the thesis. We refer the reader interested in a rigorous treatment of the Curie-Weiss model to the excellent book [32]. Readers interested in statistical mechanics in the context of neuroscience are referred to [15],[33].

1.2.1 Curie-Weiss model

(Also called the *fully-connected Ising model.*) The model consists of N magnets, which we label with $x = 1, ..., N$. The magnets have two states: they can either point up $(+1)$ or down (-1). We call this orientation their *spin*, and denote the spin of magnet i by σ_i. Furthermore, we call the N-tuple of all spin values the *configuration*, which we denote by $\sigma = (\sigma_i)_{i=1,...,N}$. There are 2^N possible configurations. The set of all possible configurations is $\Omega = \{-1, +1\}^N$. Finally, we introduce the *magnetization* of a configuration, which is the average orientation of all magnets. It is the macroscopic variable of interest, and is given by

$$m(\sigma) = \frac{1}{N}\sum_{x=1}^{N} \sigma_x.$$

Each configuration gets assigned an energy. The energy comes from the pairwise interaction of the magnets: two magnets that are aligned *decrease* the energy, and two magnets that are anti-aligned *increase* the energy. The *Hamiltonian* (a fancy word for the energy) is given by

$$H_N(\sigma) = -\frac{1}{2N}\sum_{x,y=1}^{N} \sigma_x\sigma_y.$$

The 1/2 in front of the sum is to compensate for double-counting. Note that the Hamiltonian also includes self-interaction terms of the form $\sigma_x\sigma_x$, which are not part of our model. However, the total self-interaction is the same for any configuration (it is $-1/2$) and so it is just a constant shift of the energy; this should not influence the physics of our model, and so we forget about it.

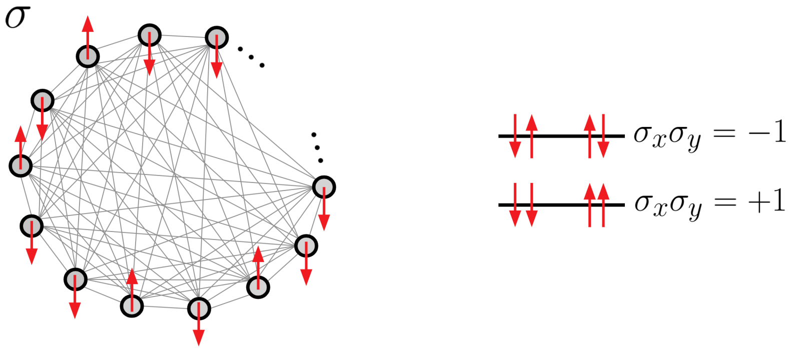


Figure 4: Left: example of a configuration. Right: energy levels. When two spins align, we either have $\sigma_x = \sigma_y = 1$ or $\sigma_x = \sigma_y = -1$. In both cases, their product $\sigma_x\sigma_y = 1$. Such pairs contribute $-1/N$ to the energy. When two spins anti-align, the product of their spins $\sigma_x\sigma_y = -1$. Such pairs contribute $1/N$ to the energy.

Let us assume that the magnets are in *equilibrium*: all macroscopic quantities (in our case, the mean magnetization) do not change over time. The fundamental result, due to Gibbs, is that all properties of equilibrium systems at a temperature T are completely determined by their energy. The probability to find the system in a certain microscopic configuration is given by the *Gibbs distribution* (also known as Boltzmann distribution):

$$\mu_{N,\beta}(\sigma) = \frac{e^{-\beta H_N(\sigma)}}{Z_{N,\beta}}, \tag{1}$$

where $\beta = 1/T$ is the *inverse temperature*, and $Z_{N,\beta} = \sum_{\sigma \in \Omega} \exp(-\beta H_N(\sigma))$ is the *partition function*; it is a normalizing constant. $\mu_{N,\beta}$ is a probability distribution. Configurations with low energy are very likely to occur, and configurations with high energy are very rare. The probability to find the system in a certain macroscopic state is given by a sum over all microscopic configurations that give rise to that specific macroscopic state:

$$\Pr(m(\sigma) = m) = \sum_{\sigma : m(\sigma) = m} \mu_{N,\beta}(\sigma).$$

The most important property of the Curie-Weiss model is that it contains a phase transition. Consider an enormous amount of tiny magnets, so that $N \to \infty$. If $\beta < 1$ (high temperatures), then $\Pr(m(\sigma) = 0) = 1$: with probability 1 the magnets cancel each other out, and no macroscopic magnetic field is left. While magnets still cooperate with each other to align, their effort is not good enough to overcome the thermal noise in the system. However, if $\beta > 1$ (low temperatures), $\Pr(m(\sigma) \neq 0) = 1$: the magnets cooperate to form a net nonzero magnetic field, and they conquer the noise. $\beta = 1$ is the critical value of this phase transition.

We now equip the model with dynamics, as we would like to study how equilibrium is reached. Of course we can arbitrarily make up some update rule, but not all will reach the Gibbs distribution in equilibrium. A natural dynamics to use is the *Glauber dynamics*, which has the following updating scheme: at each timestep,

1. Pick a spin σ_i uniform at random.

2. Compute the energy change ΔH in the system if the spin σ_i were to flip.

3. Flip σ_i with probability $\Pr(\Delta H) = 1/(1 + \exp(\beta \Delta H))$.

This dynamics has the Gibbs distribution as its *stationary distribution*: the probability that one finds the system in configuration σ after letting the dynamics run for a very long time is precisely given by $\mu_{N,\beta}(\sigma)$.

Now, if $\beta \to \infty$ (we have a very cold system), the Gibbs distribution dictates that with probability 1 the magnetization is either $+1$ or -1. This means that the Glauber dynamics converges also to this magnetization. Let m_t be the magnetization of the system at time t. Then, m_t converges to either $+1$ or -1 (depending on the initial conditions): the states $\{-1, +1\}$ are *attractor states* of the dynamics.

The Curie-Weiss model can thus be interpreted as an attractor network! It is however a very simple one, with two attractors. With some liberty, we can view it as an elementary model of memory: the states $-1, +1$ are stored memories. If we start the network an initial configuration with negative magnetization, the network will 'recall' the memory that is most like the input, namely -1. If we start with an initial configuration with positive magnetization, it will 'recall' the $+1$ memory. To convince ourselves of the link with neuroscience even more, we might call the magnets neurons instead; we're now getting close to the definition of Hopfield networks.

1.2.2 Multi-group Curie-Weiss model

(Also called the *General Block Spin Ising model.* For the mathematical definition of the model used in the thesis, see Chapter 2)

The multi-group Curie-Weiss model is a generalization of the Curie-Weiss model. The model is built by combining multiple groups of spins. Each single group is like a Curie-Weiss model, where the spins interact with themselves through a coupling. However, the spins of one group also interact with spins from other groups, and the coupling depends on which two groups the spins are in.

This model was first studied in [34], [35] and [36] for two groups, and models of more than two groups have received some attention in the last few years [37] [38]. The model has found applications in economics, sociology and ecology, but curiously not yet in neuroscience. In the thesis, we will develop a link between this model and Hopfield networks, and show that Hopfield networks are in some sense special cases of multi-group Curie-Weiss models.

Here we give a short introduction to the model. For convenience we introduce new nomenclature: we call a multi-group Curie-Weiss model with q groups a *q-group Curie-Weiss* (q-gCW) model.

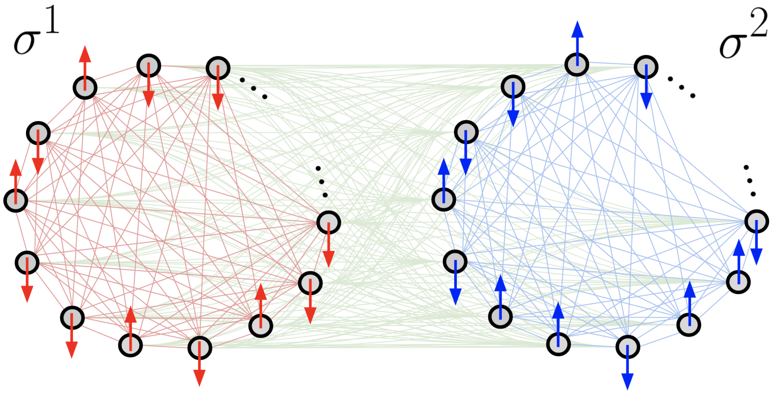


Figure 5: A 2-gCW model. All spins interact with each other, but the coupling is not always the same. Spins in the first group (red) interact with themselves (red connections) and with spins from the blue group (green connections). Similarly, spins in the second group (blue) interact with themselves (blue connections) and with spins from the red group (green connections). All connections with the same color represent the same coupling strength. The configuration on the first group is σ^1, on the second group σ^2; the full configuration of the model is the tuple (σ^1, σ^2).

A q-gCW model consists of q groups $\mathcal{V}_1, ..., \mathcal{V}_q$, and each group $i = 1, ..., q$ contains $n_i = |\mathcal{V}_i|$ spins. The total amount of spins is $n_1 + ... + n_q = N$. We store the fraction of sites in the partitions in a vector, called the *relative partition sizes vector* $\boldsymbol{X} = (n_1/N, ..., n_q/N)$. The full configuration is $\sigma = (\sigma_x)_{x=1,...,N} \in \Omega$, and the configuration on group i is given by $\sigma^i = (\sigma_x)_{x \in \mathcal{V}_i}$. On each partition we have an order parameter $m_i(\sigma^i) = \frac{1}{n_i} \sum_{x \in \mathcal{V}_i} \sigma_x$. The *state* of the full model is given by the q-component vector $\boldsymbol{m}(\sigma) = (m_1(\sigma^1), ..., m_q(\sigma^q))^\intercal$.

The coupling between spins from the same or different groups is given by the $q \times q$ *interaction matrix* $\mathcal{M}$, where the coupling between spins of partition i and partition j is given by the real number $\mathcal{M}_{ij}$ (note that the self-coupling of a partition $\mathcal{M}_{ii}$ can be different for different partitions). The interactions are given by the Hamiltonian

$$H_{N,X}(\sigma) = -\frac{1}{2N} \sum_{i,j=1}^{q} \sum_{x \in \mathcal{V}_i} \sum_{y \in \mathcal{V}_j} \mathcal{M}_{ij} \sigma_x \sigma_y. \tag{2}$$

If $q = 1$ and $\mathcal{M}_{11} = 1$, we get back the Hamiltonian of the Curie-Weiss model. Finally, we also equip this model with the same Glauber dynamics as for the Curie-Weiss model (except we now use the Hamiltonian (2)).

As this model contains multiple copies of a Curie-Weiss model, it will have multiple attractor states. Depending on $\mathcal{M}$, these attractor states can be quite complex. We can thus also interpret the multi-group Curie-Weiss model as an attractor network, and because of their richer state space they are a bit more versatile than the elementary Curie-Weiss model.

1.3 Hopfield networks

(For the mathematical definition of the model used in the thesis, see Chapter 2.)

The Hopfield network is a model of associative memory. It is a recurrent neural network: the network can be used to store memories and later recall them from input. The model originates from the Sherrington-Kirkpatrick (SK) model [39], which is a Curie-Weiss model with random Gaussian couplings between each spin. This model was studied for its *spin-glass* properties. Different models were studied independently by Little [40] and Amari [41] in 1972, who all proposed to use a modified Ising model as a model of associative memory. In his 1982 paper [23], Hopfield combined the recently developed theory for the SK model with the ideas of Little to develop the famous Hopfield network (see [42] for a comparison of the Little and Hopfield networks). The essential difference is that the Little network has synchronous dynamics (all neurons update at the same time), while the Hopfield network has asynchronous dynamics, which is biologically more plausible.

1.3.1 Model neurons

Let us construct the network from scratch, starting with the mechanics of a single neuron.

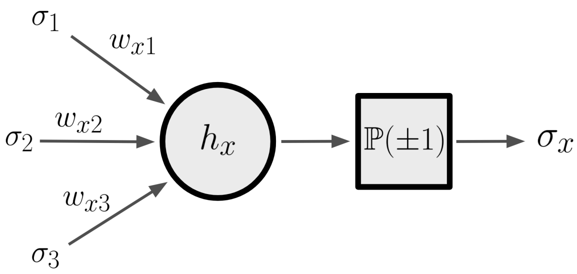


Figure 6: Schematic of a single model neuron. A neuron receives inputs $\sigma_y = \pm1$ from other neurons, and weighs these inputs via the synaptic weights w_{xy}. The total weighed input into neuron x is h_x. Depending on the sign and size of this input, neuron x wil either fire ($\sigma_x = +1$) or stay silent ($\sigma_x = -1$).

We will pretend that a neuron only has two states: firing (+1) or silent (−1). The network consists of N neurons, and each neuron $x = 1, ..., N$ has a state σ_x, which can be $+1$ or -1. The configuration of the network (the state of all neurons) is denoted by $\sigma = (\sigma_x)_{x=1,...,N}$. Any two distinct neurons x, y are connected via synapses, which have a synaptic weight $w_{xy} \in [-1, 1]$. Synapses with positive synaptic weight are excitatory, while synapses with negative synaptic weight are inhibitory. Any neuron x has a post-synaptic potential (PSP), given by

$$h_x = \sum_{y, y \neq x}^{N} w_{xy}\sigma_y.$$

It is the sum of all input from other neurons, weighed by the synaptic weights. If a synapse is excitatory, a firing pre-synaptic neuron increases the PSP of the post-synaptic neuron. If a synapse is inhibitory, a firing pre-synaptic neuron decreases the PSP of the post-synaptic neuron. Depending on the sign and size of the PSP, there is a probability that a neuron starts to fire, or becomes silent. While realistic neurons have a threshold that the PSP needs to overcome, in this model we set the threshold to 0. This does not influence the collective behaviour of the network much [15].

1.3.2 Dynamics

The dynamics of the network are as follows: at each timestep,

1. Pick a neuron x uniform at random.
2. Compute the PSP h_x.
3. Let neuron x fire ($\sigma_x = +1$) with probability $\Pr(\sigma_x = +1) = 1/(1 + \exp(-2\beta h_x))$, and let it be silent ($\sigma_x = -1$) with probability $\Pr(\sigma_x = -1) = 1 - \Pr(\sigma_x = +1)$.

The parameter β is a real nonnegative number, and determines the amount of noise in the network. If $\beta = 0$, all neurons just fire at random without caring about the state of other neurons. If $\beta \to \infty$, there is no randomness: a neuron will only fire if its PSP is positive, and will stay silent if its PSP is negative. We call the network with $\beta \to \infty$ the *deterministic* network.

Hopfield made a crucial observation about the deterministic network (the reason the model bears his name). If we have symmetric synaptic weights ($w_{xy} = w_{yx}$ for all x, y), then we can construct a function $H(\sigma)$ whose value will never increase during dynamics. That is, if the configuration at time t is $\sigma(t)$, and at the next timestep it is $\sigma(t + 1)$, then we always have $H(\sigma(t + 1)) \leq H(\sigma(t))$. It follows immediately that if the network is in a configuration that is at a local minimum of the function H, then the network will stay in this configuration forever; otherwise the function H would have to increase during dynamics. Thus, the attractor states of the network are precisely the minima of this function $H(\sigma)$. Hopfield found that

$$H(\sigma) = \frac{1}{2N} \sum_{x,y=1}^{N} w_{xy}\sigma_x\sigma_y. \tag{3}$$

The dynamics of memory recall kind of resembles the physics of a ball on a hilly landscape: The ball will start at an initial height on some hill, and will roll downward until it gets stuck in a local minimum. In this analogy, the height of the ball is the Hamiltonian, and rolling downward corresponds to recalling a memory (figure 7).

The existence of the function $H(\sigma)$ also gives us immediate results for the stochastic case. The stochastic dynamics (finite β) is precisely the Glauber dynamics, with as energy function exactly this function $H(\sigma)$. $H(\sigma)$ is thus the Hamiltonian of the network. Immediately we realize that, just as for the Curie-Weiss model, in the limit $t \to \infty$, the probability to find the network in configuration σ is given by the Gibbs distribution $\mu_{N,\beta,w}(\sigma)$.

Here we add the subscript w to indicate that the Hamiltonian (3), and thus the Gibbs distribution, depends on the weights.

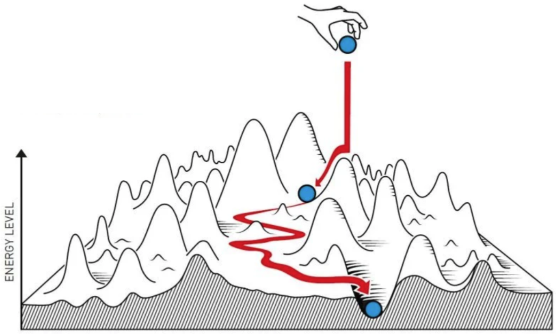


Figure 7: Analogy of the attractor dynamics. We can imagine starting the network in some initial configuration, and the dynamics will converge to a minimum of the energy landscape (the Hamiltonian). It resembles the physics of dropping a ball on a hilly landscape: the ball will only roll downwards on this landscape, until it reaches a local minimum, and, in the absence of noise, it will stay there forever. Figure adapted from [43].

1.3.3 Patterns

Memories are associated with stable activity of the neurons. For each memory, there is a certain set of neurons that are firing, and others are silent. This specification of which neurons are firing and which aren't is a configuration, and a configuration associated with a memory is called a *pattern*. So, when we want to store p memories in the network to recall, we need to construct p patterns (which are configurations), and design the network in such a way that these patterns are attractor states. When the network dynamics converges from an initial configuration to a pattern, we say that the network has 'recalled' the memory associated with that pattern.

Denote with ξ_x^i the state of neuron x in pattern i, where $i = 1, ..., p$. If we set $\xi_x^i = +1$, then neuron x is firing when we're recalling the i-th memory; if we set $\xi_x^i = -1$, the neuron is silent during recall of the i-th memory. We denote the full i-th pattern by $\xi^i = (\xi_x^i)_{x=1,...,N}$. ξ^i is a configuration. We construct p patterns randomly: for each neuron $x = 1, .., N$, and each pattern $i = 1, ..., p$ let

$$\xi_x^i = \begin{cases} +1 & \text{probability } 1/2, \\ -1 & \text{probability } 1/2. \end{cases}$$

This way, the p patterns are uncorrelated, which will decrease errors in memory recall.

We will construct the synaptic weights w_{xy} of our network in such a way that these patterns become attractor states. We will do this with a *Hebbian learning rule*. Roughly, the rule is as follows (for symmetric synapses)[15]:

- w_{xy} is enhanced when neurons x and y are active.
- w_{xy} is enhanced when neurons x and y are silent.
- w_{xy} is reduced when one neuron is active, but the other is silent.

The rule is often summarized as "neurons that fire together, wire together".

If the network configuration is equal to one of the patterns ($\sigma = \xi^i$) for a long time, then based on Hebb's rule, we expect the synapses between neurons x and y to have positive weight when $\xi_x^i = \xi_y^i$ (both neurons fire together, or are silent together). We expect the synapses to have negative weight when $\xi_x^i = -\xi_y^i$ (one neuron fires, but the other is silent). Notice that the product $\xi_x^i \xi_y^i$ is positive when $\xi_x^i = \xi_y^i$, and negative when $\xi_x^i = -\xi_y^i$. This leads to the *storage prescription* [15], which tells us that when a new pattern ξ^i is learned, the synaptic weights are linearly modified by adding the term

$$\Delta w_{xy} = \xi_x^i \xi_y^i.$$

A network that has learned p patterns is thus connected by the synaptic weights

$$w_{xy} = \sum_{i=1}^{p} \xi_x^i \xi_y^i. \tag{4}$$

Most satisfactory, when the weights (4) are implemented in the Hamiltonian (3), one can quickly show that the patterns $\xi^1, ..., \xi^p$ are energy minima. This immediately implies that the learned patterns are attractor states of the network.

1.3.4 Overlaps and phase diagram

Lastly, we introduce order parameters to describe the state of the network. We define the *overlap* of the network configuration with pattern i to be

$$m_i(\sigma) = \frac{1}{N}\sum_{x=1}^{N} \xi^i_x \sigma_x.$$

The overlap m_i essentially tells us how much the current configuration the network 'looks like' pattern i. If $m_i = 1$, the network configuration is exactly ξ^i; if $m_i = -1$, every neuron in the network configuration does exactly the opposite of ξ^i; when $m_i = 0$, the network has no overlap with pattern ξ^i. We combine all overlaps into a vector $\boldsymbol{m} = (m_1, ..., m_p)$, and call the value of this vector the state of the network.

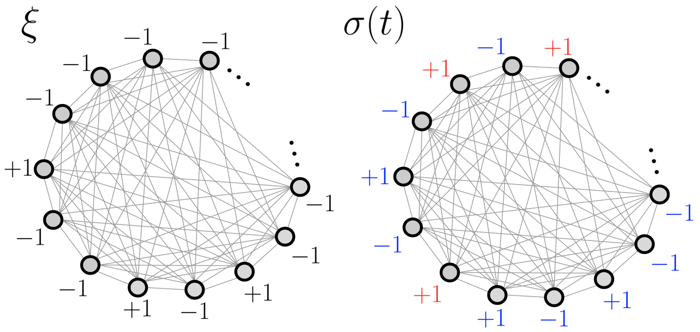


Figure 8: Left: example of a pattern. Right: example of the configuration of the network at time t. The state of each neuron that has the same state as in the pattern is coloured blue, while neurons that do the opposite are coloured red. The overlap with the pattern is the amount of neurons that agree with the pattern, minus the amount of neurons that disagree, all divided by N.

Just as the Curie-Weiss model, the Hopfield model has a phase transition for large N at $\beta = 1$. When $\beta < 1$, all overlaps $m_1, ..., m_p$ will be approximately zero. When $\beta > 1$, the network is able to retrieve patterns, and the network dynamics converges to the pattern with highest initial overlap.

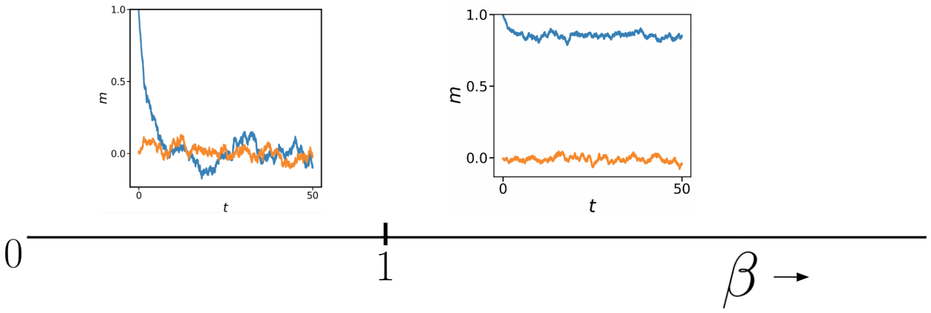


Figure 9: The phase diagram of the Hopfield network. Two simulations of the time series $m_1(t)$ (blue) and $m_2(t)$ (orange) are plotted for 50 timesteps. Both simulations start with full overlap with the first pattern: $m_1(t = 0) = 1$. If $\beta < 1$ (*disordered phase*), there is almost no overlap with any of the patterns (except for some random fluctuations). If $\beta > 0$ (*ordered phase*), the first pattern is an attractor state, and the network state stays close to the attractor state. $m_1(t)$ is not exactly equal to 1, as finite temperatures still give some random fluctuations.

1.3.5 Storage capacity

When the amount of patterns p that we want to store in the network grows too fast with respect to the amount of neurons N in the network, then the Hopfield model enters a spin-glass phase. This happens when $\alpha = p/N$ is larger than a critical value α_c. Simply put, in the spin-glass phase, the stable states of the network are no longer correlated with any of the stored patterns ξ^i [33], and the network is useless as a memory storage. The well-known result is that for Hopfield networks (with uncorrelated patterns) $\alpha_c \approx 0.137$; a similar result also exists for correlated patterns [44]. The *storage capacity* of the Hopfield network depends linearly on N, which is a big bottleneck when we want to store a lot of patterns with limited resources.

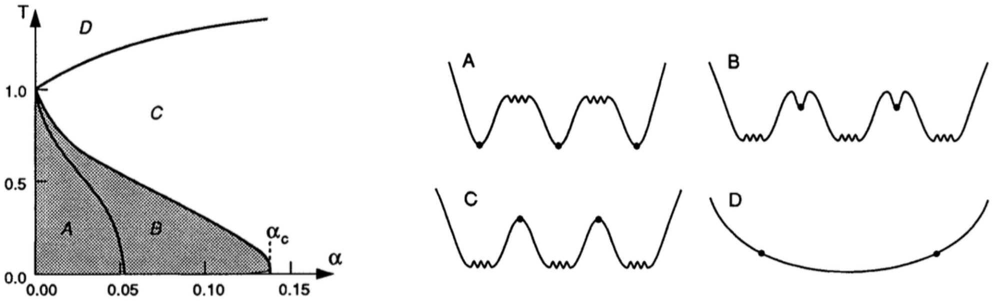


Figure 10: Left: phase diagram of the Hopfield network. Temperature $T = 1/\beta$ is on the y-axis, and $\alpha = p/N$ on the x-axis. Right: schematic of the energy landscape in each phase. A) There are a few global minima corresponding to the patterns, and between them 'spurious' local minima. B) There are many spurious global minima, but there are still local minima at the patterns. C) Spin-glass phase: there are many spurious global and local minima, but none correspond to any of the patterns. D) There is only one global minimum, corresponding to $\boldsymbol{m} = 0$. Figure from [33].

Modern Hopfield networks [45] [46] change the Hamiltonian H in order to increase storage capacity. The key idea is to make the basins of attraction around the minima more sharp, using nonlinear functions, in order to reduce memory retrieval errors.

Throughout the thesis, we will always work with *low memory loading*: $\alpha \ll 1$.

1.3.6 Asymmetric synaptic weights

It is assumed in the Hopfield network that the synaptic weights w are symmetric. With this assumption we constructed a network which can store *static* memories: once we recall a memory, we will not recall a next memory. However, as all humans are probably aware from daily life, memory recall can be a dynamical process: the recall of a first memory can be a trigger for recalling a second memory, and so on. Often we recall a temporal sequence of memories instead of just a single one.

This behaviour can be implemented by adding asymmetric weights to the synaptic weights matrix [47]. Consider two patterns ξ^1, ξ^2 and the weights $w_{xy} = \xi^2_x \xi^1_y$. Let the current configuration of the network be equal to the first pattern: $\sigma_x(t) = \xi^1_x$ for all $x = 1, ..., N$. Let's see how a uniformly at random selected neuron x updates during the next timestep.

1. At time t, $\sigma_x(t) = \xi^1_x$.
2. The PSP of the neuron is

$$h_x(t) = \sum_{y, y \neq x}^{N} w_{xy} \sigma_y(t) = \sum_{y, y \neq x}^{N} \xi^2_x \xi^1_y \xi^1_y = (N-1)\xi^2_x.$$

3. If $\xi^2_x = +1$, $h_x(t)$ is large and positive, so with high probability $\sigma_x(t+1) = +1$. If $\xi^2_x = -1$, $h_x(t)$ is large and negative, so with high probability $\sigma_x(t+1) = +1$. In any case, with high probability $\sigma_x(t+1) = \xi^2_x$.

After enough neurons have updated, we see that the network changes its configuration from ξ^1 to ξ^2. So, once the network has recalled the first pattern, it will immediately start to transition to the second pattern, and we get a temporal sequence of memories.

1.4 Sparse networks

The limited storage capacity of Hopfield networks gives them limited utility in applications. Another drawback of the Hopfield network is in terms of its biological realism: at any timestep, approximately half of all neurons fire; in realistic biological neural networks, neuron activity is much lower. Both problems can simultaneously be solved by introducing *sparse* neural networks, in which during any timestep only a fixed fraction of neurons fire.

In this work, we use a sparse network with the synaptic weights from [48]. It has been shown that such network have enhanced storage capabilities [49]. To ensure that the network activity stays sparse during dynamics, we introduce dynamics with *local inhibition*, as done by Shim et al. in [50]. In that work, Shim et al. show that a stochastic Willshaw model [51], (which is a Hopfield network with binary weight values), combined with local inhibition, functions as an attractor network. We expect that, with some minor modifications, the results of Shim et al. still carry over to 'non-clipped' weights which are not necessarily binary valued, and that our network functions as an attractor network.

1.4.1 Sparse block network

We define the sparse network that is used in this work, which we call the *sparse block network*. The network consists of N neurons. Each neuron $x = 1, ..., N$ has a state σ_x: firing ($\sigma_x = 1$) or silent ($\sigma_x = 0$). The configuration of the network is denoted by $\sigma = (\sigma_x)_{x=1,...,N}$. The space of all possible configurations is again denotedy by Ω, which is a subset of $\{0, 1\}^N$.

We partition the set of all neurons into *blocks* of length L, and L divides N. We denote these blocks as $b^{(1)} = (1, ..., L), b^{(2)} = (L + 1, ..., 2L), ..., b^{(L/N)} = (N - L + 1, ..., N)$. A neuron has a network index x (its location in the full network), and a block-index l (its location with in its block, which ranges from 1 to L). We introduce shorthand notation: if a neuron x sits at the l-th location within block k, we write its network index as $b_l^{(k)} = (k - 1)L + l$.

The synaptic weight between neurons x and y is denoted by w_{xy}. Again, any neuron x has a PSP, given by

$$h_x = \sum_{y,y\neq x}^{N} w_{xy}\sigma_y.$$

The dynamics of this network are different from the dynamics of a Hopfield network. Here we employ *winner-take-all* dynamics: within a block, only one neuron can be active, and all others must be silent. The neurons compete to be active through the value of their PSP. This dynamics is also called local inhibition: we can imagine neurons within the same block to be connected with strong inhibitory synapses, as when one neuron fires, it completely suppresses all other neurons in the block.

The dynamics of the network is as follows: at each timestep,

1. Pick a block k uniform at random from all possible L/N blocks.
2. Compute the PSPs of all neurons in the block: $h_{b_1^{(k)}}, h_{b_2^{(k)}}, ..., h_{b_L^{(k)}}$.
3. On the set of neurons in the block $\{1, ..., L\}$ construct the probability distribution

$$g_\beta(l) = \frac{\exp\left[\beta h_{b_l^{(k)}}\right]}{\sum_{m=1}^{L} \exp\left[\beta h_{b_m^{(k)}}\right]}$$

4. Sample a single neuron from $g_\beta(l)$; let this neuron fire, and set all other neurons in the block to be silent.

Under this dynamics, the network always has exactly a fraction N/L neurons firing.

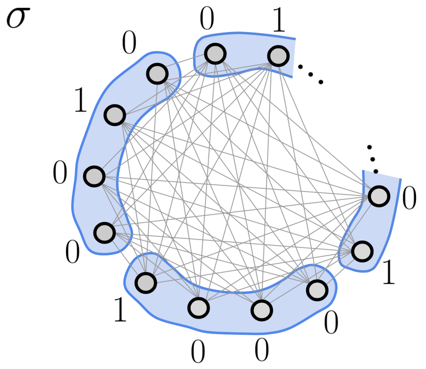


Figure 11: Example of a configuration of the sparse block model. Blocks are indicated in blue (in this example, $L = 4$). Any neuron is connected with all other neurons from different blocks. Neurons in the same block are not connected. Neurons take values in $\{0, 1\}$.

We construct p patterns to store in the network. Again, we denote patterns as ξ^i, with $i = 1, ..., p$, and the state of neuron x in pattern i is given by ξ_x^i (and note that ξ_x^i is 0 or 1). Patterns are generated randomly as follows: for a pattern i, in each of its blocks, pick one neuron uniform at random and let it fire; let all other neurons be silent.

The synaptic weights between neurons in different blocks are as in [48]. If x, y are in different blocks:

$$w_{xy} = \sum_{i=1}^{p} \left(\xi_x^i - \frac{1}{L}\right)\left(\xi_y^i - \frac{1}{L}\right) = \sum_{i=1}^{p} \zeta_x^i \zeta_y^i,$$

and if x, y are in the same block, $w_{xy} = 0$. Here $\zeta_x^i = \xi_x^i - \frac{1}{L}$.

The overlap of the network configuration with a pattern i is given by

$$m_i(\sigma) = \frac{L}{N} \sum_{x=1}^{N} \xi_x^i \sigma_x,$$

and ranges between 0 (no overlap) and 1 (full overlap). Note that on average, a randomly sampled configuration will have $1/L$ overlap with any pattern.

1.5 Markov chains

A Markov chain is a stochastic process that can model random transitions from one state to another within a finite or countable set of states. The defining property of this process is that the probability to transition to a given next state only depends on the current state. Markov chains are widely used as statistical models of real-world processes.

We denote by S the set of possible states for the Markov chain, and by X_t the state of the Markov chain at time t (note that X_t is a random variable). At time $t = 0$, we pick an *initial state*: $X_0 = x_0$, where x_0 is a state in S.

Transitions are defined as follows: if the previous states of the Markov chain are $X_0, ..., X_t$, the state at the next timestep $t + 1$ is drawn from the probability distribution

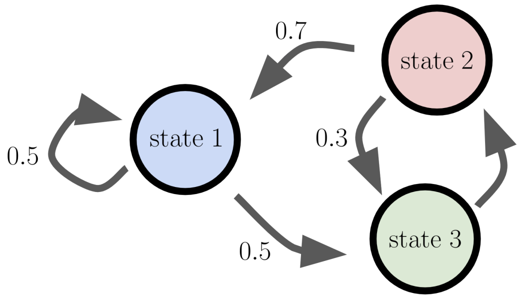
$$\Pr(X_{t+1} = x \mid X_0 = x_0, ..., X_t = x_t) = \Pr(X_{t+1} = x \mid X_t = x_t). \tag{5}$$

The probability distribution of X_{t+1} only depends on the previous state X_t. A sequence of random variables $X_0, X_1, X_2, ...$ for which (5) holds is said to satisfy the *Markov property* (sometimes $X_0, X_1, X_2, ...$ is said to be *memoryless*).

In order to build a Markov chain, we need to specify the probabilities $\Pr(X_{t+1} = y \mid X_t = x)$ for all possible values of x and y. Let $|S|$ be the amount of possible states. Then we can write the transition probabilities as an $|S| \times |S|$ matrix P, where

$$P_{xy} = \Pr(X_{t+1} = y \mid X_t = x).$$

P is called the *transition matrix*. It can be neatly visualized with a graph that describes the possible transitions and their probabilities (see figure 12 for an example).



$$P = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0.7 & 0 & 0.3 \\ 0 & 0 & 1 \end{pmatrix}$$

Figure 12: Example of a Markov chain with three states, and its transition matrix.

Defining v_x as the probability that we start the chain in state x, the $|S|$-component vector v is the probability distribution over states at time $t = 0$. The probability distribution at $t = 1$ is vP (vector-matrix multiplication). At $t = 2$ it is $(vP)P = vP^2$, and so on; the probability distribution of the chain state at time t is given by the vector vP^t.

The long-term behaviour is captured by the *stationary distribution* π, usually written as an $|S|$-component vector, and it is defined as

$$\pi = \pi P.$$

In words, it is the distribution that is unchanged as the process evolves over time. If the Markov chain is *irreducible* (any state can be reached in finite steps from any other state) and *aperiodic* (it doesn't get stuck in cycles of fixed length), the stationary distribution is the probability distribution of states that we will see in the limit $t \to \infty$: it captures the long-term behaviour of the process.

1.6 Thesis structure

First, we introduce the mathematical definitions of the Hopfield network and the multi-group Curie-Weiss network in Chapter 2. In Chapter 3, we use these definitions to prove the *Hopfield/gCW correspondence*. The idea is first introduced heuristically, and then rigorously. The correspondence is used in is used in Sections 5.2, 5.3.3, 5.5.1, and Chapter 6. In Chapter 4, we present a method to *embed* an arbitrary two-state Markov chain into an attractor neural network: the network can simulate a two-state Markov chain with its attractors. This method is applied to Hopfield networks in Sections 5.5 and to sparse block networks in Chapter 7. In Chapter 5, we construct and investigate the 2-pattern switching network, which is a modified Hopfield network. This network allows for controlled metastable state switching; this controlled state switching is one of the building blocks for the embedded Markov chain network. Its phase diagram and controlled state switching behaviour is investigated in Sections 5.3 and 5.4 respectively. In Section 5.5, we show that the 2-pattern switching network cannot be used to construct an embedded two-state Markov chain. Chapter 6 is on the metastability properties of the 2-pattern switching network. The applied mathematical theory, the *pathwise approach to metastability*, is introduced and applied to a 2-group Curie-Weiss network; its metastable properties are stated as a Theorem and proven in Section 6.4. The 2-pattern switching network has exactly the same metastable behaviour as this Curie-Weiss network, due to the Hopfield/gCW correspondence. Concluding remarks on the metastable behaviour of the 2-pattern switching network are given in Section 6.5. In Chapter 7, a sparse 2-pattern switching network is defined and used to construct a sparse network that can embed an arbitrary 2-state Markov chain. The motivation for using a sparse network is given in Section 7.2. In Section 7.3, the sparse 2-pattern switching network is analyzed with simulations, and in Section 7.4, the embedded Markov chain network is introduced and tested. Finally, Chapters 8 and 9 contain a discussion and conclusion, in which all results are summarized and compared with current literature. Furthermore, an outlook on possible improvements and applications of the results is given.

2 Definitions of networks

2.1 Equilibrium networks

We set the notation and nomenclature. The models live on a *network*, which is the complete graph $K_N = (V_N, E_N)$, where $V_N = \{1, ..., N\}$ is the set of sites of the network, and $E_N = \{\{x, y\} : x, y \in V_N\}$ the set of edges.

Each site $x \in V_N$ gets assigned a *spin* σ_x, which takes values in $\{-1, +1\}$. We call the N-tuple that contains the spins of all the sites in the network a *configuration*, which we denote as $\sigma = (\sigma_x)_{x \in V_N}$. The space of all configurations is $\Omega = \{-1, +1\}^N$. Each edge $\{x, y\} \in E_N$ gets assigned a *weight* $w_{xy} \in \mathbb{R}$.

All models will be defined through their *Hamiltonian*, which is a function $H : \Omega \to \mathbb{R}$. The value of the Hamiltonian for a particular configuration will be called the *energy* of that configuration.

The probability of finding the network in a certain configuration is dependent on its energy, and is given by the *Gibbs measure*

$$\mu(\sigma) = \frac{1}{Z} e^{-\beta H(\sigma)}, \tag{6}$$

where $\beta \geq 0$ is the *inverse temperature*, and

$$Z = \sum_{\sigma \in \Omega} e^{-\beta H(\sigma)}$$

is the *partition function.*

Finally, we equip models with a fixed amount $p \in \mathbb{N}$ of order parameters, which describe the global behaviour of the model. We denote these by $m_i \in [-1, 1] \subset \mathbb{R}$, $i \in \{1, ..., p\}$, and these order parameters are random variables on Ω. We combine these order parameters into a vector $\boldsymbol{m} = (m_i)_{i \in \{1,...,p\}}$, and we call the value of this vector the *state* of the network.

If the Hamiltonian, partition function or Gibbs measure depends on some external parameters, we will add these parameters to the notation as subscripts; for probability distribution functions we add them as superscripts.

2.1.1 Hopfield network

Definition 2.1 (p-pattern Hopfield network)**.** *Let $\xi^1, ..., \xi^p$ be random configurations in Ω, called patterns, where for all $x \in V_N$ and $i \in \{1, ..., p\}$ the pattern spins ξ^i_x are* i.i.d. Ber(1/2) *random variables on $\{-1, +1\}$. Let $\boldsymbol{\xi} = (\xi^1, ..., \xi^p) \in \Omega^p$ be a p-tuple containing all patterns. Let Q be a $p \times p$ real symmetric positive semi-definite (PSD) matrix, called the interaction matrix.*
The p-pattern Hopfield Hamiltonian is given by

$$H_{N,\boldsymbol{\xi}}(\sigma) := - \sum_{\{x,y\} \in E_N} w_{xy} \sigma_x \sigma_y,$$

where

$$w_{xy} := \begin{cases} \frac{1}{N} \sum_{i,j=1}^{p} Q_{ij} \xi^i_x \xi^j_y & \text{if } x \neq y, \\ 0 & \text{if } x = y. \end{cases}$$

The Gibbs measure of the Hopfield network is

$$\mu_{N,\beta,\boldsymbol{\xi}}(\sigma) = \frac{1}{Z_{N,\beta,\boldsymbol{\xi}}} e^{-\beta H_{N,\boldsymbol{\xi}}(\sigma)}.$$

The order parameters that describe the state of the network are

$$m_i(\sigma) = \frac{1}{N} \sum_{x \in V_N} \xi^i_x \sigma_x.$$

with $i \in \{1, ..., p\}$, and the state of the Hopfield network is $\boldsymbol{m}(\sigma) = (m_1(\sigma), ..., m_p(\sigma))^\intercal$.

Remark 2.1.1. *The Hopfield Hamiltonian can be written in terms of the network state as follows:*

$$H_{N,\boldsymbol{\xi}}(\sigma) = -\frac{N}{2} \sum_{i,j=1}^{p} Q_{ij} \left(\frac{1}{N} \sum_{x \in V_N} \xi^i_x \sigma_i \right) \left(\frac{1}{N} \sum_{y \in V_N} \xi^j_y \sigma_y \right) + c = -\frac{N}{2} \boldsymbol{m}^\intercal(\sigma) Q \boldsymbol{m}(\sigma) + c,$$

where $c > 0$ is a constant that removes the contribution of the self-interaction terms w_{xx}; the value of this constant depends on the particular realization of the random patterns. However, as adding a constant to the Hamiltonian does not influence the Gibbs measure, we will leave it out from here on. We will rely heavily on this expression for the Hamiltonian in our analysis of the Hopfield network.

Definition 2.2. *We denote with $f_{\boldsymbol{\xi}}^{N,\beta}(\boldsymbol{\xi}') := \mathbb{P}_{N,\beta}(\boldsymbol{\xi} = \boldsymbol{\xi}')$ be the probability mass function under which the random variable $\boldsymbol{\xi} \in \Omega^p$ is defined; by our definition of the random patterns (Def. 2.1) $f_{\boldsymbol{\xi}}^{N,\beta}$ is the uniform distribution on Ω^p. For all $\boldsymbol{m}' \in \operatorname{Im}(\boldsymbol{m}), \boldsymbol{\xi}' \in \Omega^p$, the conditional probability density function of the network state is given by*

$$\begin{aligned} f_{\boldsymbol{m}|\boldsymbol{\xi}}^{N,\beta}(\boldsymbol{m}', \boldsymbol{\xi}') &:= \mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) = \boldsymbol{m}' | \boldsymbol{\xi} = \boldsymbol{\xi}') \\ &= \mu_{N,\beta,\boldsymbol{\xi}'}(\{\sigma \in \Omega : \boldsymbol{m}(\sigma) = \boldsymbol{m}'\}) \\ &= |\{\sigma \in \Omega \ : \ \boldsymbol{m}(\sigma) = \boldsymbol{m}'\}| \frac{1}{Z_{N,\beta,\boldsymbol{\xi}'}} e^{-\frac{N}{2}\boldsymbol{m}'^{\intercal} Q \boldsymbol{m}'}. \end{aligned}$$

The joint probability density/mass function is given by

$$\begin{aligned} f_{\boldsymbol{m},\boldsymbol{\xi}}^{N,\beta}(\boldsymbol{m}', \boldsymbol{\xi}') &:= \mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) = \boldsymbol{m}', \boldsymbol{\xi} = \boldsymbol{\xi}') \\ &= \mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) = \boldsymbol{m}' | \boldsymbol{\xi} = \boldsymbol{\xi}') \mathbb{P}_N(\boldsymbol{\xi} = \boldsymbol{\xi}'), \end{aligned}$$

and the marginal probability density function of the network state is

$$f_{\boldsymbol{m}}^{N,\beta}(\boldsymbol{m}') := \mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) = \boldsymbol{m}') = \sum_{\boldsymbol{\xi}' \in \Omega^p} f_{\boldsymbol{m},\boldsymbol{\xi}}^{N,\beta}(\boldsymbol{m}', \boldsymbol{\xi}').$$

2.1.2 Multi-group Curie-Weiss network

We introduce additional notation for this model. Again, we start with a network K_N, but this time we split the sites of the network into q *partitions*. First we pick the sizes of the partitions $n_i \in \mathbb{N}$ for $i \in \{1, ..., q\}$, such that $\sum_{i=1}^q n_i = N$. Setting $n_0 = 0$, we then define the partitions as

$$\mathcal{V}_i = \left\{ \left(\sum_{j=1}^{i-1} n_j \right) + 1, ..., \sum_{j=1}^{i} n_j \right\}, \qquad V_N = \bigcup_{i=1}^q \mathcal{V}_i,$$

i.e. the first n_1 sites belong to $\mathcal{V}_1$, the next n_2 sites to $\mathcal{V}_2$ and so on; and every site is in some partition.

Furthermore, we denote the *relative partition sizes* as $X_i = n_i/N$ for $i \in \{1, ..., q\}$, and we define a vector $\boldsymbol{X} = (X_1, ..., X_k)$ and a $q \times q$ diagonal matrix $\mathcal{X} = \operatorname{diag}(X_1, ..., X_q)$. The relative partition sizes take values in $S_N = \{\frac{l}{N} : l = 0, ..., N\}$, and the vector of relative partition sizes takes values in $P_{N,q} = \{x \in (S_N)^q : \|x\|_1 = 1\}$.

Each site in a partition $x \in \mathcal{V}_i$ gets assigned a spin $\tilde{\sigma}_x \in \{-1, +1\}$, and each partition has its own configuration, denoted as $\tilde{\sigma}^i = (\tilde{\sigma}_x)_{x \in \mathcal{V}_i}$; we write the full configuration as $\tilde{\sigma} = (\tilde{\sigma}_x)_{x \in V_N}$. The space of all possible configurations on a partition is $\Omega_i = \{-1, +1\}^{|\mathcal{V}_i|}$, and the space of all possible full configurations is again Ω.

We also split the set of edges E_N of the network into partitions, where we sort them by the site partitions that they connect. Let $\mathcal{E}_{ij} = \{\{x, y\} \in E_N : x \in \mathcal{V}_i, y \in \mathcal{V}_j\}$. Note that $\mathcal{E}_{ij} = \mathcal{E}_{ji}$; we will take the convention to always write the smaller index first. Also note that $\mathcal{E}_{ii}$ is the set of edges that connects all sites within the partition $\mathcal{V}_i$ with each other.

Definition 2.3 (q-group Curie Weiss network)**.** *Let $\boldsymbol{X}$ be a q-dimensional vector of relative partition sizes, and let $\mathcal{V}_1, ..., \mathcal{V}_q$ be the corresponding partitions of V_N. Let $\mathcal{M}$ be a $q \times q$ real symmetric PSD matrix, called the interaction matrix.*
The q-group Curie Weiss (q-gCW) Hamiltonian is given by

$$\tilde{H}_{N,\boldsymbol{X}}(\tilde{\sigma}) := -\sum_{\{x,y\}\in E_N} \tilde{w}_{xy}\tilde{\sigma}_x\tilde{\sigma}_y,$$

where

$$\tilde{w}_{xy} := \begin{cases} \frac{1}{N}\sum_{i,j=1}^{q} \mathcal{M}_{ij}\mathbb{1}_{\{x,y\}\in\mathcal{E}_{ij}}, & \textit{if } x \neq y, \\ 0, & \textit{if } x = y, \end{cases}$$

i.e. the weight between a site $x \in \mathcal{V}_i$ and a site $y \in \mathcal{V}_j$ is $\mathcal{M}_{ij}$.
The Gibbs measure of the q-gCW network is

$$\nu_{N,\beta,\boldsymbol{X}}(\tilde{\sigma}) = \frac{1}{\tilde{Z}_{N,\beta,\boldsymbol{X}}} e^{-\beta \tilde{H}_{N,\boldsymbol{X}}(\tilde{\sigma})}.$$

The order parameters that describe the state of the network are

$$\tilde{m}_i(\tilde{\sigma}) = \frac{1}{n_i}\sum_{x\in\mathcal{V}^i} \tilde{\sigma}_x,$$

with $i \in \{1, ..., q\}$, and the state of the q-gCW network is $\tilde{\boldsymbol{m}}(\tilde{\sigma}) = (\tilde{m}_1(\tilde{\sigma}), ..., \tilde{m}_q(\tilde{\sigma}))$.

Remark 2.3.1. *The q-gCW Hamiltonian can be written in terms of the network state as follows:*

$$\tilde{H}_{N,\boldsymbol{X}}(\tilde{\sigma}) = -\frac{N}{2}\sum_{i,j=1}^{q} \mathcal{M}_{ij}\left(X_i \frac{1}{n_i}\sum_{x\in\mathcal{V}_i}\tilde{\sigma}_x\right)\left(X_j \frac{1}{n_j}\sum_{y\in\mathcal{V}_j}\tilde{\sigma}_y\right) + c = -\frac{N}{2}\tilde{\boldsymbol{m}}^\intercal(\tilde{\sigma})\mathcal{X}\mathcal{M}\mathcal{X}\tilde{\boldsymbol{m}}(\tilde{\sigma}) + c$$

again with c some constant that depends on the partitions. Like with the Hopfield Hamiltonian, we will leave the constant out as it doesn't influence the Gibbs measure. We will also rely heavily on this expression for the Hamiltonian in our analyses.

Definition 2.4. *For all $\tilde{\boldsymbol{m}}' \in \mathrm{Im}(\tilde{\boldsymbol{m}}), \boldsymbol{X} \in P_{N,q}$, the conditional probability density function of the network state is given by*

$$\begin{aligned} \tilde{f}^{N,\beta}_{\tilde{\boldsymbol{m}}|\boldsymbol{X}}(\tilde{\boldsymbol{m}}', \boldsymbol{X}') &:= \mathbb{P}_{N,\beta}(\tilde{\boldsymbol{m}}(\tilde{\sigma}) = \tilde{\boldsymbol{m}}'|\boldsymbol{X} = \boldsymbol{X}') \\ &= \nu_{N,\beta,\boldsymbol{X}'}(\{\tilde{\sigma} \in \Omega : \tilde{\boldsymbol{m}}(\tilde{\sigma}) = \tilde{\boldsymbol{m}}'\}) \\ &= |\{\tilde{\sigma} \in \Omega \ : \ \tilde{\boldsymbol{m}}(\tilde{\sigma}) = \tilde{\boldsymbol{m}}'\}|\frac{1}{Z_{N,\beta,\boldsymbol{X}'}} e^{-\frac{N}{2}\tilde{\boldsymbol{m}}'^\intercal \mathcal{X}\mathcal{M}\mathcal{X}\tilde{\boldsymbol{m}}'}. \end{aligned}$$

Definition 2.5. *A random q-gCW network is a q-gCW network where the sizes of the partitions $n_1, ..., n_q$ are random. We define a probability density function of $\boldsymbol{X}$ on $P_{N,q}$ as $f^N_{\boldsymbol{X}}(\boldsymbol{X}') := \mathbb{P}_N(\boldsymbol{X} = \boldsymbol{X}')$. The random partitions can immediately be reconstructed from knowledge of the random relative partition sizes vector.*

Definition 2.6. *The joint probability density/mass function of the random q-gCW network is given by*

$$\begin{aligned} \tilde{f}^{N,\beta}_{\tilde{\boldsymbol{m}},\boldsymbol{X}}(\tilde{\boldsymbol{m}}', \boldsymbol{X}') &:= \mathbb{P}_{N,\beta}(\tilde{\boldsymbol{m}}(\tilde{\sigma}) = \tilde{\boldsymbol{m}}', \boldsymbol{X} = \boldsymbol{X}') \\ &= \mathbb{P}_{N,\beta}(\tilde{\boldsymbol{m}}(\tilde{\sigma}) = \tilde{\boldsymbol{m}}'|\boldsymbol{X} = \boldsymbol{X}')\mathbb{P}_N(\boldsymbol{X} = \boldsymbol{X}'), \end{aligned}$$

and the marginal density function of the network state is

$$\tilde{f}^{N,\beta}_{\tilde{\boldsymbol{m}}}(\tilde{\boldsymbol{m}}') := \mathbb{P}_{N,\beta}(\tilde{\boldsymbol{m}}(\tilde{\sigma}) = \tilde{\boldsymbol{m}}') = \sum_{\boldsymbol{X}'\in P_{N,q}} \tilde{f}^{N,\beta}_{\tilde{\boldsymbol{m}},\boldsymbol{X}}(\tilde{\boldsymbol{m}}', \boldsymbol{X}').$$

2.2 Dynamics of networks

We define dynamics on our networks. They will be modeled as discrete time Markov chains, where the states of the chain are configurations. We start in an initial configuration, and at each step of the chain, we update one or more spins of the network with some given probability to go to a new configuration.

First we define the single spin probability mass function. From this function, we will construct the transition matrices of our Markov chains.

Definition 2.7. *Let $s \in \{-1, +1\}$, and let $\sigma \in \Omega$ be a configuration. We denote by $\sigma^{x \to s}$ the configuration σ where the spin at site x is set to the value s. That is, for all $y \in V_N$,*

$$\sigma_y^{x \to s} := \begin{cases} \sigma_y, & \text{if } x \neq y, \\ s, & \text{if } x = y. \end{cases}$$

(we will only use this notation when there is no confusion with the previously defined notation $\sigma^i = (\sigma_x)_{x \in \mathcal{V}_i}$). The single spin probability distribution function is given by

$$g_\sigma^{N,\beta}(x, s) := \frac{\exp[-\beta H(\sigma^{x \to s})]}{\exp[-\beta H(\sigma^{x \to s})] + \exp[-\beta H(\sigma^{x \to -s})]}.$$

2.2.1 Asynchronous dynamics

The asynchronous dynamics (also called Glauber dynamics) is the discrete time Markov chain $\sigma(t) \in \Omega$, where at each timestep $t \in \mathbb{N}$ we pick a site $x' \in V_N$ uniform at random, and set $\sigma_{x'}(t+1) = s$ with probability $f_{\sigma(t)}^{N,\beta}(x', s)$, while all other sites remain the same: $\sigma_{y \neq x'}(t+1) = \sigma_{y \neq x'}(t)$.

Definition 2.8. *For any $\sigma, \eta \in \Omega$ such that $\sigma \neq \eta$, the transition matrix of the asynchronous dynamics is given by*

$$\pi_{N,\beta}(\sigma, \eta) := \begin{cases} \frac{1}{N} g_\sigma^{N,\beta}(x', \eta_{x'}) & \text{if } \exists x' \in V_N : \eta_{x'} \neq \sigma_{x'} \text{ and } \eta_x = \sigma_x \ \forall x \neq x' \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\pi_{N,\beta}(\sigma, \sigma) = 1 - \sum_{\eta \in \Omega \setminus \{\sigma\}} \pi_{N,\beta}(\sigma, \eta).$$

The asynchronous dynamics is reversible with respect to the Gibbs measure (6). To see this, note that for any two configurations σ, η that differ only at a single fixed site x', we have

$$H(\sigma^{x' \to \eta_{x'}}) = H(\eta), \qquad H(\eta^{x' \to \sigma_{x'}}) = H(\sigma),$$

and so, for such configurations

$$\pi_{N,\beta}(\sigma, \eta) = \frac{1}{N} \frac{e^{-\beta H(\eta)}}{e^{-\beta H(\sigma)} + e^{-\beta H(\eta)}}, \qquad \pi_{N,\beta}(\eta, \sigma) = \frac{1}{N} \frac{e^{-\beta H(\sigma)}}{e^{-\beta H(\sigma)} + e^{-\beta H(\eta)}}.$$

Then,

$$\frac{\pi_{N,\beta}(\sigma, \eta)}{\pi_{N,\beta}(\eta, \sigma)} = \frac{e^{-\beta H(\eta)}}{e^{-\beta H(\sigma)}} = \frac{\frac{1}{Z} e^{-\beta H(\eta)}}{\frac{1}{Z} e^{-\beta H(\sigma)}}.$$

The *detailed balance condition*

$$\mu_{N,\beta}(\sigma) \pi_{N,\beta}(\sigma, \eta) = \mu_{N,\beta}(\eta) \pi_{N,\beta}(\eta, \sigma)$$

thus holds for any two configurations $\sigma, \eta \in \Omega$ that differ only at one site; for all other configurations it holds trivially.

3 p-pattern Hopfield/2^{p-1}-group Curie-Weiss correspondence

The weight of an edge between two sites of the Hopfield network is determined by the values of the patterns at those sites. We will simplify the analysis of the Hopfield network by grouping together sites with the same values of patterns. The partitions that we create in this way allow us to rewrite the weight of an edge in terms of what partitions the edge connects with each other; much like the definition of weights for a multi-group Curie-Weiss network. From this, one might suspect that it is possible to develop a relation between the two different types of networks, where one could compute quantities for one network by computing a related quantity in the other network. In this chapter, we will develop the link between the two different types of networks.

3.1 Heuristic example

Consider a 'standard' two-pattern Hopfield network. Any edge $\{x, y\} \in E_N$ has weight

$$w_{xy} = \xi_x^1 \xi_y^1 + \xi_x^2 \xi_y^2. \tag{7}$$

Now, we split the sites of the network into two partitions, $\mathcal{V}_1$ and $\mathcal{V}_2$, defined as follows:

$$\mathcal{V}_1 := \{x \in V_N : \xi_x^1 = \xi_x^2\}, \mathcal{V}_2 := \{x \in V_N : \xi_x^1 = -\xi_x^2\}.$$

Notice that if both $x, y \in \mathcal{V}_1$, we can rewrite the weight (7) as

$$w_{xy} = \xi_x^1 \xi_y^1 + \xi_x^1 \xi_y^1 = 2\xi_x^1 \xi_y^1, \qquad x, y \in \mathcal{V}_1.$$

If both $x, y \in \mathcal{V}_2$, we have

$$w_{xy} = \xi_x^1 \xi_y^1 + (-\xi_x^1)(-\xi_y^1) = 2\xi_x^1 \xi_y^1, \qquad x, y \in \mathcal{V}_2,$$

and if $x \in \mathcal{V}_1, y \in \mathcal{V}_2$,

$$w_{xy} = \xi_x^1 \xi_y^1 + \xi_x^1(-\xi_y^1) = 0$$

(the case $x \in \mathcal{V}_2, y \in \mathcal{V}_1$) is the same, as the weights are symmetric).

We can thus write the weights as follows:

$$w_{xy} = \left(\mathcal{M}_{11} \mathbb{1}_{\{x,y\} \in \mathcal{E}_{11}} + \mathcal{M}_{22} \mathbb{1}_{\{x,y\} \in \mathcal{E}_{22}}\right) \xi_x^1 \xi_y^1.$$

where $\mathcal{M}_{11} = \mathcal{M}_{22} = 2$ (and $\mathcal{M}_{12} = \mathcal{M}_{21} = 0$).

The weights of a standard two-pattern Hopfield network can thus be written as weights of a 2-group Curie-Weiss network, with only the first pattern left in the expression. We can get rid of this pattern with a gauge transformation, defining the spins $\tilde{\sigma}_x := \xi_x^1 \sigma_x$ (note that this transformation is reversible). Then, the Hopfield Hamiltonian can be written as the Hamiltonian of a 2-group Curie-Weiss network (Definition 2.3). As the Hamiltonian completely determines the properties of the network, we see that a two-pattern Hopfield network is in this sense equal to a 2-group Curie-Weiss network. This procedure can be generalized: we prove a similar connection for Hopfield networks with p patterns (where p is finite).

3.2 Results

We develop the connection between Hopfield networks and multi-group Curie-Weiss networks. The connection is most easily made for Hopfield networks with orthogonal patterns.

Definition 3.1 (Overlap and orthogonal patterns)**.** *Define a set of paired indices $I_p := \{(i, j) \in \{1, ..., p\}^2 : i < j\}$; $|I_p| = p(p-1)/2$.*
*Let $\boldsymbol{\xi} = (\xi^1, ..., \xi^p) \in \Omega^p$ be a p-tuple of patterns. For $(i, j) \in I_p$, the **overlap** of two patterns $\xi^i, \xi^j \in \boldsymbol{\xi}$ is*

$$O_{(i,j)}(\boldsymbol{\xi}) := \left| \frac{1}{N} \sum_{x \in V_N} \xi_x^i \xi_x^j \right| \in [0, 1].$$

*Two patterns $\xi^i, \xi^j \in \boldsymbol{\xi}$ are said to be **orthogonal** to each other when $O_{(i,j)}(\boldsymbol{\xi}) = 0$. The p-tuple of patterns $\boldsymbol{\xi}$ is said to be **orthogonal** if all patterns $\xi^1, ..., \xi^p \in \boldsymbol{\xi}$ are orthogonal to each other. We denote the set of all orthogonal p-tuples of patterns by $\Omega_\perp^p := \{\boldsymbol{\xi} \in \Omega^p : O_{(i,j)}(\boldsymbol{\xi}) = 0 \; \forall (i, j) \in I_p\}$. Furthermore, for any $\boldsymbol{\xi} \in \Omega^p$ we define*

$$d(\boldsymbol{\xi}, \Omega_\perp^p) := \max_{(i,j) \in I_p} O_{(i,j)}(\boldsymbol{\xi}).$$

Remark 3.1.1. *$\Omega_\perp^p$ is empty whenever N is finite and odd, and $\Omega_\perp^p$ can be empty when N is small and p is large (for example, $N = 2$ and $p = 3$). We will only consider networks with **low loading**, that is, $N \geq 2^p$. Whenever we consider a network with orthogonal patterns, we assume N to be even.*

Corollary 3.1.1. *$d(\boldsymbol{\xi}, \Omega^p_\perp) = 0$ iff $\boldsymbol{\xi} \in \Omega^p_\perp$.*

Theorem 3.2 (Hopfield/gCW correspondence)**.** *For any p-pattern Hopfield network with N sites, interaction matrix Q and with any orthogonal p-tuple of patterns $\boldsymbol{\xi}'$, there exists a $p \times 2^{p-1}$ matrix A, and a 2^{p-1}-gCW network with N sites, interaction matrix $\mathcal{M} = A^\intercal Q A$ and relative partition sizes $\boldsymbol{X}' = (1/2^{p-1}, ..., 1/2^{p-1})$, such that*

$$\boldsymbol{m}(\sigma) = \frac{1}{2^{p-1}} A \tilde{\boldsymbol{m}}(\tilde{\sigma}),$$

where $\tilde{\sigma} = (\xi^1_x \sigma_x)_{x \in V_N}$ is the ***gauge transform*** *of a configuration $\sigma \in \Omega$. Furthermore,*

1. *$H_{N,\boldsymbol{\xi}'}(\sigma) = \tilde{H}_{N,\boldsymbol{X}'}(\tilde{\sigma})$,*
2. *$Z_{N,\beta,\boldsymbol{\xi}'} = \tilde{Z}_{N,\beta,\boldsymbol{X}'}$,*
3. *$\mu_{N,\beta,\boldsymbol{\xi}'}(\sigma) = \nu_{N,\beta,\boldsymbol{X}'}(\tilde{\sigma})$.*

Corollary 3.2.1 (Allowed states)**.** *For any $i \in \{1, ..., 2^{p-1}\}$, the set of all possible values of the order parameter $\tilde{m}_i$ is*

$$\mathrm{Im}(\tilde{m}_i) = \left\{ -1 + \frac{2k}{n_i} : k = 0, ..., n_i \right\}.$$

The space of all possible states of the 2^{p-1}-gCW network is

$$\mathrm{Im}(\tilde{\boldsymbol{m}}) = \bigotimes_{i=1}^{2^{p-1}} \mathrm{Im}(\tilde{m}_i)$$

The space of all possible states of a p-pattern Hopfield network with orthogonal patterns is

$$\mathrm{Im}(\tilde{\boldsymbol{m}}) = \left\{ \frac{1}{2^{p-1}} A \tilde{\boldsymbol{m}}' : \tilde{\boldsymbol{m}} \in \mathrm{Im}(\tilde{\boldsymbol{m}}) \right\}.$$

We will sometimes call states in these spaces ***allowed states****.*

Corollary 3.2.2. *Let $S \subseteq \mathrm{Im}(\boldsymbol{m})$ be any set of Hopfield network states, and $\mathcal{A}(S) = \{\tilde{\boldsymbol{m}}' \in \mathrm{Im}(\tilde{\boldsymbol{m}}) : \frac{1}{2^{p-1}} A \tilde{\boldsymbol{m}}' \in S\}$ be the set of all states of the 2^{p-1}-gCW network that map to the same set of Hopfield network states S. We have that*

$$\mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) \in \cdot \, | \boldsymbol{\xi} \in \Omega^p_\perp) = \mathbb{P}_{N,\beta}(\tilde{\boldsymbol{m}}(\tilde{\sigma}) \in \mathcal{A}(\cdot) | \boldsymbol{X} = \boldsymbol{X}'),$$

i.e. the probability of finding the Hopfield network in a certain set of states is equal to the probability of finding the 2^{p-1}-gCW network in the set of states that map to these Hopfield network states.

Remark 3.2.1. *The gauge transformation in Theorem 3.2 is sometimes also called a* ***Mattis transformation*** *[52].*

It is also possible to prove a more general statement that works for any Hopfield network (not just for networks with orthogonal patterns), without changing much of the proof of Theorem 3.2. The 2^{p-1}-gCW network that corresponds to a p-pattern Hopfield network with non-orthogonal patterns will have a different relative partition sizes vector. We will work this out for the 2-pattern Hopfield model in the following

Theorem 3.3 (2-pattern Hopfield/gCW correspondence)**.** *Consider any 2-pattern Hopfield network with N sites, interaction matrix Q and any two patterns $\boldsymbol{\xi}' = (\xi^1, \xi^2) \in \Omega^2$ (not necessarily orthogonal). Let $X = |\{x \in V_N : \xi^1_x = \xi^2_x\}|$ and*

$$(A)_{ij} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

There exists a 2-gCW network with N sites, interaction matrix $\mathcal{M} = A^\intercal Q A$ and relative partition sizes $\boldsymbol{X}' = (X, 1 - X)$, such that $\boldsymbol{m}(\sigma) = A \mathcal{X}' \tilde{\boldsymbol{m}}(\tilde{\sigma})$, where we defined $\mathcal{X}' = \mathrm{diag}(X, 1 - X)$. Furthermore,

1. *$H_{N,\boldsymbol{\xi}'}(\sigma) = \tilde{H}_{N,\boldsymbol{X}'}(\tilde{\sigma})$,*
2. *$Z_{N,\beta,\boldsymbol{\xi}'} = \tilde{Z}_{N,\beta,\boldsymbol{X}'}$,*
3. *$\mu_{N,\beta,\boldsymbol{\xi}'}(\sigma) = \nu_{N,\beta,\boldsymbol{X}'}(\tilde{\sigma})$.*

Corollary 3.3.1. *Consider the same 2-pattern Hopfield- and 2-group Curie-Weiss networks as in Theorem 3.3. Let* $\mathcal{A}(S) = \{\frac{1}{2}\mathcal{X}^{-1}A\boldsymbol{m} : \boldsymbol{m} \in S\}$.

$$\mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) \in \cdot\ |\boldsymbol{\xi} = (\xi^1, \xi^2)) = \mathbb{P}_{N,\beta}(\tilde{\boldsymbol{m}}(\tilde{\sigma}) \in \mathcal{A}(\cdot)|\boldsymbol{X} = (X, 1-X)).$$

Remark 3.3.1. *From Corollary 3.3.1 we see that we don't exactly need to know the pattern values at each site to compute state probabilities; we only need to know how often the patterns agree or disagree (which is captured in the relative partition sizes vector* $\boldsymbol{X}$*). This is a very satisfactory result: no site is more special than any other, and no spin direction* $(+1, -1)$ *is preferred. The only thing that matters for the state probabilities is whether globally the patterns compete or cooperate. The 2-pattern Hopfield network is invariant under a pattern transformation that preserves the relative partition sizes vector.*

To prove that the Hamiltonian $\tilde{H}_{N,\boldsymbol{X}'}(\tilde{\sigma})$ in Theorem 3.2 is indeed the Hamiltonian of a multi-group Curie-Weiss network, we need the following

Lemma 3.1. *Let* $p, q \in \mathbb{N}$ *and* $p \leq q$*. Let* Q *be a* $p \times p$ *real symmetric positive semi-definite (PSD) matrix, and let* A *be a* $p \times q$ *real matrix. Then,* $A^\intercal Q A$ *is a* $q \times q$ *real symmetric PSD matrix.*

The next step is to use Theorem 3.2 for large N. The intuition is that as the amount of sites increases, any tuple of patterns will become more and more orthogonal. This is because by the law of large numbers, the ratio between the amount of sites where any two patterns agree and where they disagree approaches the expected amount, which is an equal division of sites.

Lemma 3.2. *Let* $f_{\boldsymbol{\xi}}^{N,\beta}$ *be the uniform distribution on* Ω^p*, and let* $\boldsymbol{\xi} \sim f_{\boldsymbol{\xi}}^{N,\beta}$ *be a random* p*-tuple of patterns. For any fixed* $\delta > 0, \varepsilon > 0$ *there exists* $N \in \mathbb{N}$ *large enough such that*

$$\mathbb{P}_{N,\beta}(d(\boldsymbol{\xi}, \Omega_\perp^p) \geq \delta) \leq \varepsilon.$$

We now want to derive the marginal probability density function of states of the Hopfield network, for networks with a large amount of sites. This marginal probability density function will tell us what the phase diagram of the Hopfield network looks like. While it is difficult to get an exact expression, we *can* obtain a large deviations result. This in turn will allow us to get a good estimate of the marginal probability density function, and will allow us to extract information about the phase diagram.

We first prove that for large N, the logarithm of the marginal probability density function can be written as a logarithm of the conditional probability density function, where we condition on the patterns of the network being (almost) orthogonal.

Lemma 3.3. *For any* $\delta > 0$*, let* $\Delta_\delta := \{\boldsymbol{\xi} \in \Omega^p : d(\boldsymbol{\xi}, \Omega_\perp^p) < \delta\}$ *be the set of* p*-tuples of patterns that are (almost) orthogonal. Let* $E \subseteq \operatorname{Im}(m)$ *be an event.*
Then, for any fixed $\delta > 0$ *and* $\varepsilon > 0$*, there exists* N *large enough such that*

$$|\log\left[\mathbb{P}_{N,\beta}(m \in E)\right] - \log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\right]| < \varepsilon.$$

We will use Lemma 3.3 to prove

Proposition 3.1. *Let* $E \subseteq \operatorname{Im}(m)$ *be an event.*

$$\lim_{N\to\infty} \frac{1}{N} \log\left[\mathbb{P}_{N,\beta}(m \in E)\right] = \lim_{N\to\infty} \frac{1}{N} \log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Omega_\perp^p)\right].$$

The last step is to use a result from Knöpfel et al., who derived a Large Deviations Principle (LDP) for the multi-group Curie-Weiss network (also called 'General Block Spin Ising Model') [37]. We reformulate their result using our previous notation and nomenclature. Let us explicitly denote that $\tilde{\boldsymbol{m}}$ and $\boldsymbol{X}$ depend on N, by writing $\tilde{\boldsymbol{m}}^{(N)}$ and $\boldsymbol{X}^{(N)}$. Furthermore, let $\boldsymbol{Y} = \lim_{N\to\infty} \boldsymbol{X}^{(N)}$.

Theorem 3.4 (Knöpfel, Löwe, Schubert, Sinulis)**.** *Let $q \in \mathbb{N}$ and $\mathcal{M}$ be an interaction matrix of a q-gCW network. Let $\boldsymbol{Y} = (1/q, ..., 1/q)$, i.e. the relative partition sizes become equally large asymptotically.*
The sequence $(\tilde{\boldsymbol{m}}^{(N)})_{N\in\mathbb{N}}$ satisfies an LDP under $(\nu_{N,\beta,\boldsymbol{X}^{(N)}})_{N\in\mathbb{N}}$ with speed N and rate function

$$R_\beta(\tilde{\boldsymbol{m}}) := \beta \left(F_\beta(\tilde{\boldsymbol{m}}) - \inf_{\tilde{\boldsymbol{m}}' \in [-1,1]^q} F_\beta(\tilde{\boldsymbol{m}}') \right),$$

where

$$F_\beta(\tilde{\boldsymbol{m}}) := \frac{1}{2q^2} \tilde{\boldsymbol{m}}^\intercal \mathcal{M} \tilde{\boldsymbol{m}} - \frac{1}{\beta q} \sum_{i=1}^{q} S(\tilde{m}_i),$$

and

$$S\left(\tilde{m}_i\right) := -\frac{1+\tilde{m}_i}{2} \log \frac{1+\tilde{m}_i}{2} - \frac{1-\tilde{m}_i}{2} \log \frac{1-\tilde{m}_i}{2}.$$

Corollary 3.4.1. *If we simply take the elements of the sequence $(\boldsymbol{X}^{(N)})_{N\in\mathbb{N}}$ to be $\boldsymbol{X}^{(N)} = (1/q, ..., 1/q)$ for all N, then for a closed set $E \subset [-1,1]^p$,*

$$\lim_{N\to\infty} \frac{1}{N} \log \mathbb{P}_{N,\beta}(\tilde{\boldsymbol{m}}(\tilde{\sigma}) \in E \,|\boldsymbol{X} = (1/q, ..., 1/q))) = - \inf_{\tilde{\boldsymbol{m}}' \in E} R_\beta(\tilde{\boldsymbol{m}}').$$

We are now in a position to put all the pieces together, and obtain a similar 'large deviations limit' for the Hopfield model.

Theorem 3.5 (Hopfield LDP)**.** *Let $E \subset [-1,1]^p$ be a closed set.*

$$\lim_{N\to\infty} \frac{1}{N} \log \mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) \in E) = - \inf_{\tilde{\boldsymbol{m}}' \in \mathcal{A}(E)} R_\beta(\tilde{\boldsymbol{m}}').$$

3.3 Proofs

Proof of Lemma 3.1. As $p \leq q$, we have for all $v \in \mathbb{R}^q$ that $Av \in \mathbb{R}^p$. Therefore, as Q is PSD, $v^\intercal(A^\intercal QA)v = (Av)^\intercal Q(Av) \geq 0$ for all $v \in \mathbb{R}^q$, so by definition $A^\intercal QA$ is also PSD. Both Q and A are real matrices, so $A^\intercal QA$ is also a real matrix. As Q is symmetric, $(A^\intercal QA)^\intercal = A^\intercal Q^\intercal A = A^\intercal QA$, so $A^\intercal QA$ is also symmetric. □

Proof of Theorem 3.2. From a given p-pattern Hopfield network with orthogonal patterns, we will explicitly construct the 2^{p-1} partitions of the 2^{p-1}-gCW network. On each partition, we define the corresponding order parameters and we show that the Hopfield order parameters are linear combinations of the gCW order parameters. Explicit examples of the construction that follows are given in the 'Examples'-section.

We start by partitioning the set of sites of our network, which is the set V_N. For any site $x \in V_N$, we compare the spins of the patterns on that site with the spin of the first pattern. Each other pattern can have equal, or opposite spin compared to the first pattern spin; there are thus 2^{p-1} ways that the other pattern spins can relate to the first pattern spin. We create 2^{p-1} disjoint partitions

$$\begin{gathered}\mathcal{V}_1 := \{x \in V_N \;:\; \xi_x^1 = \xi_x^2 = ... = \xi_x^p\}, \\ \mathcal{V}_2 := \{x \in V_N \;:\; \xi_x^1 = \xi_x^2 = ... = -\xi_x^p\}, \\ \vdots \\ \mathcal{V}_{2^{p-1}} := \{x \in V_N \;:\; \xi_x^1 = -\xi_x^2 = ... = -\xi_x^p\}.\end{gathered}$$

We can also write these partitions with the help of a $p \times 2^{p-1}$ matrix A. Let the set of all possible vectors of length $p-1$ with entries $-1, +1$ be $\{-1,+1\}^{p-1}$, and let

$$(A)_{ij} := \begin{pmatrix} 1 & 1 & ... & 1 \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & ... & \boldsymbol{v}_{2^{p-1}} \end{pmatrix}, \qquad \boldsymbol{v}_l \in \{-1,+1\}^{p-1},$$

where for $l \in \{1, ..., 2^{p-1}\}$, $\boldsymbol{v}_l$ are column vectors. We can now write the partitions as

$$\begin{gathered}\mathcal{V}_1 := \{x \in V_N \;:\; A_{11}\xi_x^1 = A_{21}\xi_x^2 = ... = A_{p1}\xi_x^p\}, \\ \mathcal{V}_2 := \{x \in V_N \;:\; A_{12}\xi_x^1 = A_{22}\xi_x^2 = ... = A_{p2}\xi_x^p\}, \\ \vdots \\ \mathcal{V}_{2^{p-1}} := \{x \in V_N \;:\; A_{1(2^{p-1})}\xi_x^1 = A_{2(2^{p-1})}\xi_x^2 = ... = A_{p(2^{p-1})}\xi_x^p\}.\end{gathered}$$

Note that $A_{ij} = 1/A_{ij}$ for all $i, j \in \{1, ..., p\}$, and $A_{1j} = 1$ for all $j \in \{1, ..., p\}$. So, when a site $x \in \mathcal{V}_j$, then at that site we have that $A_{ij}\xi_x^i = A_{1j}\xi_x^1$ for all $i \in \{1, ..., p\}$, which implies that at that site, $\xi_x^i = A_{ij}\xi_x^1$ for all $i \in \{1, ..., p\}$.

The p-tuple of patterns $\boldsymbol{\xi}'$ of the Hopfield network is orthogonal, which means that for any two patterns $\xi^i, \xi^j \in \boldsymbol{\xi}'$ we have $|\frac{1}{N}\sum_{x \in V_N} \xi_x^i \xi_x^j| = 0$. Therefore, there must be exactly as many sites where these patterns agree ($\xi_x^i \xi_x^j = 1$) as where they disagree ($\xi_x^i \xi_x^j = -1$). This implies that each of our partitions must contain exactly the same amount of sites. Furthermore, by our definition of the partitions, all sites will end up in some partition. As there were 2^{p-1} partitions and N sites, each partition $\mathcal{V}_i, i \in \{1, ..., 2^{p-1}\}$ contains $n_i = N/2^{p-1}$ sites. The relative partition sizes vector is then $\boldsymbol{X}' = (1/2^{p-1}, ..., 1/2^{p-1})$.

Now we define the order parameters. Let $\tilde{\sigma} = (\xi_x^1 \sigma_x)_{x \in V_N} \in \Omega$ denote the gauge transformation of any configuration $\sigma \in \Omega$; this transformation is a bijection from Ω to itself. Define for all $j \in \{1, ..., 2^{p-1}\}$

$$\tilde{m}_j(\tilde{\sigma}) := \frac{1}{n_j}\sum_{x \in \mathcal{V}_j} \tilde{\sigma}_x = \frac{1}{n_j}\sum_{x \in \mathcal{V}_j} \xi_x^1 \sigma_x.$$

These are the order parameters of a 2^{p-1}-gCW network with relative partition sizes vector $\boldsymbol{X}'$.

We rewrite the Hopfield order parameters. Using the remarks above, we get for any $\sigma \in \Omega$ and $i \in \{1, ..., p\}$

$$\begin{aligned} m_i(\sigma) &= \frac{1}{N}\sum_{x \in V_N} \xi_x^i \sigma_x \\ &= \sum_{j=1}^{2^{p-1}} \frac{1}{N}\sum_{x \in \mathcal{V}_j} \xi_x^i \sigma_x \\ &= \sum_{j=1}^{2^{p-1}} \frac{n_j}{N}\frac{1}{n_j}\sum_{x \in \mathcal{V}_j} \xi_x^i \sigma_x \\ &= \frac{1}{2^{p-1}}\sum_{j=1}^{2^{p-1}} \frac{1}{n_j}\sum_{x \in \mathcal{V}_j} A_{ij}\xi_x^1 \sigma_x \\ &= \frac{1}{2^{p-1}}\sum_{j=1}^{2^{p-1}} A_{ij}\tilde{m}_j(\tilde{\sigma}), \end{aligned}$$

and so

$$\boldsymbol{m}(\sigma) = \frac{1}{2^{p-1}}A\tilde{\boldsymbol{m}}(\tilde{\sigma}). \tag{8}$$

We conclude that if the p patterns of the Hopfield network are orthogonal, we can express any of its order parameters as a linear combination of order paramters of a 2^{p-1}-gCW network with relative partition sizes vector $\boldsymbol{X}' = (1/2^{p-1}, ..., 1/2^{p-1})$.

We now prove the other claims of the theorem. As the p patterns of the Hopfield network are orthogonal, we can apply (8). Let us define the $2^{p-1} \times 2^{p-1}$ matrix $\mathcal{M} := A^\intercal Q A$. By Lemma 3.1, $\mathcal{M}$ is real symmetric PSD, and is thus an interaction matrix for a 2^{p-1}-gCW network. We further define the $2^{p-1} \times 2^{p-1}$ diagonal matrix $\mathcal{X} =: \text{diag}(1/2^{p-1}, ..., 1/2^{p-1})$. For any $\sigma \in \Omega$, the Hopfield Hamiltonian with patterns $\boldsymbol{\xi}'$ is

$$\begin{aligned} H_{N,\boldsymbol{\xi}'}(\sigma) &= -\frac{N}{2}\boldsymbol{m}^\intercal(\sigma) Q \boldsymbol{m}(\sigma) \\ &= -\frac{N}{2}\left(\frac{1}{2^{p-1}}\tilde{\boldsymbol{m}}^\intercal(\tilde{\sigma})A^\intercal\right) Q \left(\frac{1}{2^{p-1}}A\tilde{\boldsymbol{m}}(\tilde{\sigma})\right) \\ &= -\frac{N}{2}\tilde{\boldsymbol{m}}^\intercal(\tilde{\sigma})\mathcal{X}\mathcal{M}\mathcal{X}\tilde{\boldsymbol{m}}(\tilde{\sigma}) = \tilde{H}_{N,\boldsymbol{X}'}(\tilde{\sigma}). \end{aligned} \tag{9}$$

This is the Hamiltonian of a 2^{p-1}-gCW network with relative partition sizes vector $\boldsymbol{X}'$. It immediately follows that the Hopfield partition function with patterns $\boldsymbol{\xi}'$ is

$$\begin{aligned} Z_{N,\beta,\boldsymbol{\xi}'} &= \sum_{\sigma \in \Omega} e^{-\beta H_{N,\boldsymbol{\xi}'}(\sigma)} \\ &= \sum_{\sigma \in \Omega} e^{-\beta \tilde{H}_{N,\boldsymbol{X}'}(\tilde{\sigma})} \\ &= \sum_{\tilde{\sigma} \in \Omega} e^{-\beta \tilde{H}_{N,\boldsymbol{X}'}(\tilde{\sigma})} = \tilde{Z}_{N,\beta,\boldsymbol{X}'}, \end{aligned} \tag{10}$$

where the second to last step comes from the fact that the gauge transformation is a bijection. This is the partition function of a 2^{p-1}-gCW network with relative partition sizes vector $\boldsymbol{X}'$.

Combining (9) and (10), we see that the Hopfield Gibbs measure with patterns $\boldsymbol{\xi}'$ is

$$\mu_{N,\beta,\boldsymbol{\xi}'}(\sigma) = \nu_{N,\beta,\boldsymbol{X}'}(\tilde{\sigma}),$$

which is the Gibbs measure of a 2^{p-1}-gCW network with relative partition sizes vector $\boldsymbol{X}'$.

□

Proof of Corollary 3.2.2. Fix any $\boldsymbol{\xi}' \in \Omega_{\perp}^p$. By Theorem 3.2,

$$\begin{aligned}\mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) \in \cdot|\boldsymbol{\xi} = \boldsymbol{\xi}') &= \mu_{N,\beta,\boldsymbol{\xi}'}(\{\sigma \in \Omega : \boldsymbol{m}(\sigma) \in \cdot\}) \\ &= \nu_{N,\beta,\boldsymbol{X}'}(\{\tilde{\sigma} \in \Omega : \boldsymbol{m}(\sigma) \in \cdot\}) \\ &= \nu_{N,\beta,\boldsymbol{X}'}\left(\left\{\tilde{\sigma} \in \Omega : \frac{1}{2^{p-1}} A\tilde{\boldsymbol{m}}(\tilde{\sigma}) \in \cdot\right\}\right) \\ &= \nu_{N,\beta,\boldsymbol{X}'}\left(\{\tilde{\sigma} \in \Omega : \tilde{\boldsymbol{m}}(\tilde{\sigma}) \in \mathcal{A}(\cdot)\}\right) \\ &= \mathbb{P}_{N,\beta}(\tilde{\boldsymbol{m}}(\tilde{\sigma}) \in \mathcal{A}(\cdot)|\boldsymbol{X} = \boldsymbol{X}').\end{aligned}$$

As $\boldsymbol{\xi}'$ was any arbitrary orthogonal p-tuple of patterns, we conclude that

$$\mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) \in \cdot|\boldsymbol{\xi} \in \Omega_{\perp}^p) = \mathbb{P}_{N,\beta}(\tilde{\boldsymbol{m}}(\tilde{\sigma}) \in \mathcal{A}(\cdot)|\boldsymbol{X} = \boldsymbol{X}').$$

□

Proof of Corollary 3.2.1. For any $i \in \{1, ..., 2^{p-1}\}$, we can find the values $\tilde{m}_i$ is allowed to take by flipping spins of the partition $\mathcal{V}_i$, starting the configuration on the partition in the all-negative configuration; $\tilde{\sigma}^i = (-1)_{x\in\mathcal{V}_i}$. The spins outside the partition do not influence the value of this order parameter. We find

$$\text{Im}(\tilde{m}_i) = \{-1 + \frac{2k}{n_i} \ : \ k = 0, ..., n_i\} \quad \forall i \in \{1, ..., 2^{p-1}\},$$

and as the values of the order parameters are only affected by the spins in their own partition (and the partitions are disjoint),

$$\text{Im}(\tilde{\boldsymbol{m}}) = \bigotimes_{i=1}^{2^{p-1}} \text{Im}(\tilde{m}_i).$$

By Theorem 3.2, we conclude that the allowed values of $\boldsymbol{m}$ are

$$\text{Im}(\boldsymbol{m}) = \left\{\frac{1}{2^{p-1}} A\tilde{\boldsymbol{m}}' \ : \ \tilde{\boldsymbol{m}}' \in \text{Im}(\tilde{\boldsymbol{m}})\right\}.$$

□

Proof of Theorem 3.3. The proof is very similar to the proof of Theorem 3.2. We partition the sites of our network into two groups:

$$\begin{aligned}\mathcal{V}_1 &:= \{x \in V_N : \xi_x^1 = \xi_x^2\}, \\ \mathcal{V}_2 &:= \{x \in V_N : \xi_x^1 = -\xi_x^2\}.\end{aligned}$$

Denote by $X := n_1/N$ the fraction of the amount of sites at which the two patterns agree. The relative partition sizes vector is then $\boldsymbol{X}' = (X, 1 - X)$. Denote by $\tilde{\sigma}$ the gauge transformation of a configuration $\sigma \in \Omega$ as before. Define the order parameters

$$\tilde{m}_1(\tilde{\sigma}) := \frac{1}{n_1} \sum_{x\in\mathcal{V}^1} \tilde{\sigma}_x, \qquad \tilde{m}_2(\tilde{\sigma}) := \frac{1}{n_2} \sum_{x\in\mathcal{V}^2} \tilde{\sigma}_x.$$

Note that

$$\begin{aligned}m_1(\sigma) &= \frac{1}{N} \sum_{x\in V_N} \xi_x^1 \sigma_x \\ &= \frac{1}{N} \sum_{x\in\mathcal{V}_1} \xi_x^1 \sigma_x + \frac{1}{N} \sum_{x\in\mathcal{V}_2} \xi_x^1 \sigma_x \\ &= X \frac{1}{n_1} \sum_{x\in\mathcal{V}_1} \xi_x^1 \sigma_x + (1 - X) \frac{1}{n_2} \sum_{x\in\mathcal{V}_2} \xi_x^2 \sigma_x \\ &= X\tilde{m}_1(\tilde{\sigma}) + (1 - X)\tilde{m}_2(\tilde{\sigma}),\end{aligned}$$

and similarly

$$\begin{aligned}
m_2(\sigma) &= \frac{1}{N}\sum_{x\in V_N}\xi_x^2\sigma_x \\
&= X\frac{1}{n_1}\sum_{x\in\mathcal{V}_1}\xi_x^1\sigma_x + (1-X)\frac{1}{n_2}\sum_{x\in\mathcal{V}_2}(-\xi_x^1)\sigma_x \\
&= X\tilde{m}_1(\tilde{\sigma}) - (1-X)\tilde{m}_2(\tilde{\sigma}).
\end{aligned}$$

Define the 2×2 diagonal matrix $\mathcal{X}' = \mathrm{diag}(X, 1-X)$. We can then write

$$\boldsymbol{m}(\sigma) = A\mathcal{X}\tilde{\boldsymbol{m}}(\tilde{\sigma}).$$

Again, the matrix $\mathcal{M} := A^{\intercal}QA$ is an interaction matrix of a 2-gCW network, by Lemma 3.1. With exactly the same procedure as in the proof of Theorem 3.2, we can obtain the rest of the claims that were to be proven.

□

Proof of Corollary 3.3.1. Fix an arbitrary tuple of patterns $\boldsymbol{\xi}' = (\xi^1, \xi^2) \in \Omega^2$ and let $\boldsymbol{X}' = (X, 1-X)$. As $\mathcal{X}'$ and A are both invertible (with $A^{-1} = \frac{1}{2}A$) and the linear map $A\mathcal{X}'$ is a bijection, we have by Theorem 3.3

$$\begin{aligned}
\mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma)\in\cdot|\boldsymbol{\xi}=\boldsymbol{\xi}') &= \mu_{N,\beta,\boldsymbol{\xi}'}(\{\sigma\in\Omega : \boldsymbol{m}(\sigma)\in\cdot\}) \\
&= \nu_{N,\beta,\boldsymbol{X}'}(\{\tilde{\sigma}\in\Omega : \boldsymbol{m}(\sigma)\in\cdot\}) \\
&= \nu_{N,\beta,\boldsymbol{X}'}(\{\tilde{\sigma}\in\Omega : A\mathcal{X}'\tilde{\boldsymbol{m}}(\tilde{\sigma})\in\cdot\}) \\
&= \nu_{N,\beta,\boldsymbol{X}'}(\{\tilde{\sigma}\in\Omega : \tilde{\boldsymbol{m}}(\tilde{\sigma})\in\mathcal{A}(\cdot)\}) \\
&= \mathbb{P}_{N,\beta}(\tilde{\boldsymbol{m}}(\tilde{\sigma})\in\mathcal{A}(\cdot)|\boldsymbol{X}=\boldsymbol{X}').
\end{aligned}$$

□

Proof of Lemma 3.2. As $\boldsymbol{\xi}$ is drawn uniformly at random from Ω^p, for all $i\in\{1,..,p\}$ and $x\in V_N$, the pattern spins ξ_x^i are i.i.d. Ber(1/2) random variables on $\{-1,+1\}$.

Then for any $(i,j)\in I_p$, $(k,l)\in I_p$ and $x,y\in V_N$ (except for the case that both $(i,j)=(k,l)$ and $x=y$), the random variables $\xi_x^i\xi_x^j$ and $\xi_y^k\xi_y^l$ are i.i.d. Ber(1/2) on $\{-1,+1\}$, so the random variables $O_{(i,j)}(\boldsymbol{\xi}) = |\frac{1}{N}\sum_{x\in V_N}\xi_x^i\xi_x^j|$ and $O_{(k,l)}(\boldsymbol{\xi}) = |\frac{1}{N}\sum_{x\in V_N}\xi_x^k\xi_x^l|$ are also i.i.d. Therefore all random variables $O_t(\boldsymbol{\xi}), t\in I_p$ are i.i.d., which we denote as $O_t(\boldsymbol{\xi})\sim O, \forall t\in I_p$; here O is the absolute value of the empirical average of an i.i.d. Ber(1/2) process of N trials with outcomes $\{-1,+1\}$.

We have that by the weak law of large numbers, for any $\delta>0, \kappa>0$ there exists N large enough such that

$$\mathbb{P}_N\left(O\geq\delta\right)\leq\kappa.$$

We find

$$\begin{aligned}
\mathbb{P}_{N,\beta}\left(d(\boldsymbol{\xi},\Omega_{\perp}^p)\geq\delta\right) &= 1-\mathbb{P}_{N,\beta}\left(d(\boldsymbol{\xi},\Omega_{\perp}^p)<\delta\right) \\
&= 1-\mathbb{P}_{N,\beta}\left(O_t(\boldsymbol{\xi})<\delta\ \forall t\in I_p\right) \\
&= 1-\left(\mathbb{P}_{N,\beta}(O<\delta)\right)^{\frac{p(p-1)}{2}} \\
&= 1-\left(1-\mathbb{P}_{N,\beta}(O\geq\delta)\right)^{\frac{p(p-1)}{2}} \\
&\leq 1-(1-\kappa)^{\frac{p(p-1)}{2}} := \varepsilon.
\end{aligned}$$

For N sufficiently large, for any arbitrarily small ε, we can choose κ so that the bound that was to be proven holds.

□

Proof of Lemma 3.3. We start with the upper bound. First note that

$$\begin{aligned}
\mathbb{P}_{N,\beta}(m\in E) &= \mathbb{P}_{N,\beta}(m\in E|\boldsymbol{\xi}\in\Delta_\delta)\mathbb{P}_N(\boldsymbol{\xi}\in\Delta_\delta) + \mathbb{P}_{N,\beta}(m\in E|\boldsymbol{\xi}\notin\Delta_\delta)\mathbb{P}_N(\boldsymbol{\xi}\notin\Delta_\delta) \\
&\leq \mathbb{P}_{N,\beta}(m\in E|\boldsymbol{\xi}\in\Delta_\delta) + \mathbb{P}_N(\boldsymbol{\xi}\notin\Delta_\delta).
\end{aligned}$$

By Lemma 3.2, for any $\delta>0, \kappa>0$ there is N large enough such that

$$\mathbb{P}_N(\boldsymbol{\xi}\notin\Delta_\delta)\leq\kappa.$$

This implies that for any $\delta > 0, \kappa > 0$ there is N large enough such that

$$\begin{aligned} \log\left[\mathbb{P}_{N,\beta}(m \in E)\right] &\leq \log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta) + \mathbb{P}_{N,\beta}(\boldsymbol{\xi} \notin \Delta_\delta)\right] \\ &\leq \log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta) + \kappa\right]. \end{aligned} \tag{11}$$

We now want to get this small κ out of the logarithm. For this, observe the following. For any small $\varepsilon > 0$, as $\log(x)$ is continuous and monotone increasing for $x \in \mathbb{R}^+$, there exists $0 < \lambda < 1$ such that $\log(x+\lambda) < \log(x)+\varepsilon$. The upper bound $\lambda < 1$ is later needed.

So fix an $\varepsilon > 0$. As κ in (47) was arbitrary, let us set $\kappa = \lambda$. Then, for any $\delta > 0$ there is N large enough such that

$$\log\left[\mathbb{P}_{N,\beta}(m \in E)\right] < \log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\right] + \varepsilon. \tag{12}$$

Now we work on the lower bound. For N large enough,

$$\begin{aligned} \log\left[\mathbb{P}_{N,\beta}(m \in E)\right] &\geq \log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\mathbb{P}_N(\boldsymbol{\xi} \in \Delta_\delta)\right] \\ &= \log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\right] + \log\left[\mathbb{P}_N(\boldsymbol{\xi} \in \Delta_\delta)\right] \\ &\geq \log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\right] + \log\left[1-\kappa\right]. \end{aligned} \tag{13}$$

Remember we set $\kappa = \lambda$, and so we want to bound $\log[1-\lambda]$ from below. $(1-\lambda) \in \mathbb{R}^+$, so by our definition of λ we have $\log[(1-\lambda)+\lambda] < \log[1-\lambda] + \varepsilon$, which implies $\log[1-\lambda] > -\varepsilon$.

Together with (13) this implies that for any $\delta > 0$, there is N large enough so that

$$\log\left[\mathbb{P}_{N,\beta}(m \in E)\right] > \log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\right] - \varepsilon. \tag{14}$$

Combining bounds (12) and (14) then gives the result.

□

Proof of Lemma 3.1. We will use the bounds of Lemma 3.3 to 'squeeze' the limit of our desired marginal distribution function. Lemma 3.3 assures that for any arbitrarily small $\delta > 0$ and $\varepsilon > 0$, there exisits N_0 large enough such that

$$\frac{1}{N}\log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\right] - \frac{\varepsilon}{N} < \frac{1}{N}\log\left[\mathbb{P}_{N,\beta}(m \in E)\right] < \frac{1}{N}\log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\right] + \frac{\varepsilon}{N},$$

for all $N \in \{N_0, N_0+1, ...\}$.

$$\begin{aligned} \lim_{N\to\infty}\left(\frac{1}{N}\log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\right] - \frac{\varepsilon}{N}\right) &= \lim_{N\to\infty}\left(\frac{1}{N}\log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\right] + \frac{\varepsilon}{N}\right) \\ &= \lim_{N\to\infty}\left(\frac{1}{N}\log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\right]\right), \end{aligned}$$

and so by the squeeze theorem,

$$\lim_{N\to\infty}\frac{1}{N}\log\left[\mathbb{P}_{N,\beta}(m \in E)\right] = \lim_{N\to\infty}\frac{1}{N}\log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Delta_\delta)\right]. \tag{15}$$

In the limit $N \to \infty$, for any small $\delta > 0$, N will be large enough for the bounds to work, and so (15) will hold. As δ can thus be arbitrarily small,

$$\lim_{N\to\infty}\frac{1}{N}\log\left[\mathbb{P}_{N,\beta}(m \in E)\right] = \lim_{N\to\infty}\frac{1}{N}\log\left[\mathbb{P}_{N,\beta}(m \in E|\boldsymbol{\xi} \in \Omega_\perp^p)\right],$$

where we use that $\boldsymbol{\xi} \in \Omega_\perp^p$ iff $d(\boldsymbol{\xi}, \Omega_\perp^p) = 0$.

□

Proof of Theorem 3.5. The result follows from combining Proposition 3.1, Corollary 3.2.2 and Corollary 3.4.1.

□

3.4 Examples

Here we consider a few example Hopfield networks and explicitly construct the 2^{p-1}-gCW networks that correspond to them. First we will show the structure of the 4-gCW network that corresponds to the 3-pattern Hopfield network. After that, we analyze the 2-pattern Hopfield network using its corresponding 2-gCW network.

Example 3.6 (3-pattern Hopfield network)**.** *For* $p = 3$*, set of all column vectors with elements* $-1, +1$ *of length* $p - 1 = 2$ *is*

$$\{-1,+1\}^2 = \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right\}.$$

The matrix A *is then*

$$(A)_{ij} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{pmatrix}.$$

The partitions are

$$\begin{aligned} \mathcal{V}_1 &= \{x \in V_N \ : \ \xi_x^1 = \xi_x^2 = \xi_x^3\}, \\ \mathcal{V}_2 &= \{x \in V_N \ : \ \xi_x^1 = \xi_x^2 = -\xi_x^3\}, \\ \mathcal{V}_3 &= \{x \in V_N \ : \ \xi_x^1 = -\xi_x^2 = \xi_x^3\}, \\ \mathcal{V}_4 &= \{x \in V_N \ : \ \xi_x^1 = -\xi_x^2 = -\xi_x^3\}. \end{aligned}$$

The order parameters of the 3*-pattern Hopfield network with orthogonal patterns and the order parameters of the* 4*-gCW network are related as follows:*

$$\begin{aligned} m_1(\sigma) &= \frac{1}{4}(\tilde{m}_1(\tilde{\sigma}) + \tilde{m}_2(\tilde{\sigma}) + \tilde{m}_3(\tilde{\sigma}) + \tilde{m}_4(\tilde{\sigma})), \\ m_2(\sigma) &= \frac{1}{4}(\tilde{m}_1(\tilde{\sigma}) + \tilde{m}_2(\tilde{\sigma}) - \tilde{m}_3(\tilde{\sigma}) - \tilde{m}_4(\tilde{\sigma})), \\ m_3(\sigma) &= \frac{1}{4}(\tilde{m}_1(\tilde{\sigma}) - \tilde{m}_2(\tilde{\sigma}) + \tilde{m}_3(\tilde{\sigma}) - \tilde{m}_4(\tilde{\sigma})). \end{aligned}$$

Consider the 3*-pattern Hopfield network with the* 3×3 *identity matrix as interaction matrix. The interaction matrix of the corresponding* 4*-gCW network is*

$$(\mathcal{M})_{ij} = (A^\intercal Q A)_{ij} = \begin{pmatrix} 3 & 1 & 1 & -1 \\ 1 & 3 & -1 & 1 \\ 1 & -1 & 3 & 1 \\ -1 & 1 & 1 & 3 \end{pmatrix}.$$

We see that sites couple most strongly to sites in the same partition, and more weakly to sites in other partitions. The coupling for two sites in different partitions can be positive or negative.

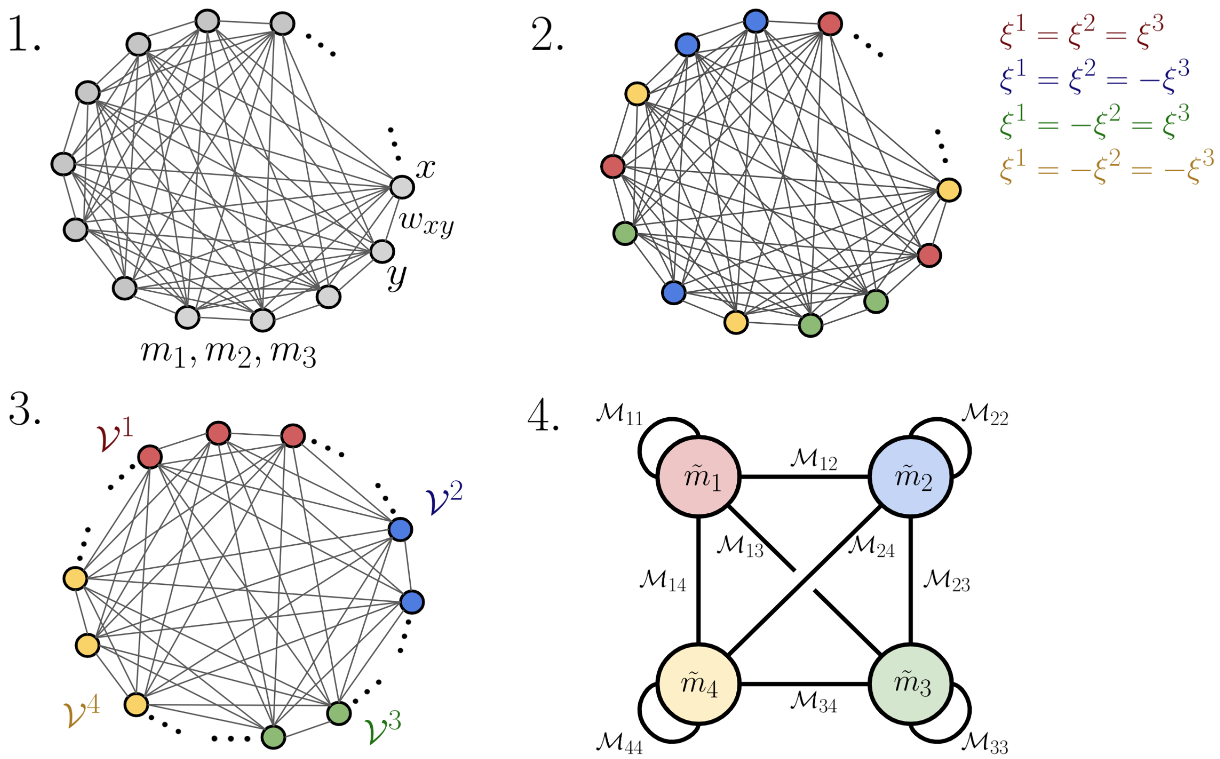


Figure 13: Example of the partitioning procedure for three patterns. The network is a complete graph. 1: The 3-pattern Hopfield network has order parameters m_1, m_2, m_3 and the weight between two sites x and y is w_{xy}, whose value depends on the pattern spins at x and y. 2: The sites of the Hopfield network are coloured based on the spins of the patterns at that site. Each pattern spin at a site is compared with the spin of the first pattern at that site. For three patterns, this gives four possible colours. 3: Sites with the same colour are grouped together into a partition. For three orthogonal patterns, the sites are grouped into four distinct partitions of equal size. 4: We define an order parameter on each partition, thus creating order parameters $\tilde{m}_1, \tilde{m}_2, \tilde{m}_3, \tilde{m}_4$. The value of the weights w_{xy} only depends on what partitions x and y are in; the weight values are encoded in the matrix $\mathcal{M}$. This is a 4-group Curie-Weiss network.

Example 3.7 (2-pattern standard Hopfield network)**.** *The* 2*-pattern case is curious. Consider the* 2*-pattern Hopfield network with orthogonal patterns* $\boldsymbol{\xi}'$*, and as interaction matrix the* 2×2 *indentity matrix. We have* $\boldsymbol{X}' = (1/2, 1/2)$*, and one can compute that*

$$(A)_{ij} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad (\mathcal{M})_{ij} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

The partitions are

$$\mathcal{V}_1 = \{x \in V_N \ : \ \xi^1_x = \xi^2_x\},$$
$$\mathcal{V}_2 = \{x \in V_N \ : \ \xi^1_x = -\xi^2_x\}.$$

If we compute the Hamiltonian, we find

$$H_{N,\boldsymbol{\xi}'}(\sigma) = \tilde{H}_{N,\boldsymbol{X}'}(\tilde{\sigma}) = -\frac{N}{2}\left(\frac{1}{2}\tilde{m}_1(\tilde{\sigma})^2 + \frac{1}{2}\tilde{m}_2(\tilde{\sigma})^2\right).$$

The Boltzmann factor in the Gibbs measure thus is a product of two Boltzmann factors:

$$e^{-\beta H_{N,\boldsymbol{\xi}'}(\sigma)} = e^{\beta \frac{N}{4}\tilde{m}_1(\tilde{\sigma})^2} e^{\beta \frac{N}{4}\tilde{m}_2(\tilde{\sigma})^2}.$$

Defining the 'standard' (1-group) Curie-Weiss partition function and Curie-Weiss Gibbs measure on both partitions

$$\tilde{Z}^i_{n,\beta} = \sum_{\tilde{\sigma}^i \in \mathcal{V}_i} e^{\beta \frac{n}{2}\tilde{m}_i(\tilde{\sigma}^i)^2}, \qquad \rho^i_{n,\beta}(\tilde{\sigma}) = \frac{1}{\tilde{Z}^i_{n,\beta}} e^{\beta \frac{n}{2}\tilde{m}_i(\tilde{\sigma}^i)^2},$$

one can show that

$$\mu_{N,\beta,\boldsymbol{\xi}'}(\sigma) = \nu_{N,\beta,\boldsymbol{X}'}(\tilde{\sigma}) = \rho^1_{N/2,\beta}(\tilde{\sigma}^1) \cdot \rho^2_{N/2,\beta}(\tilde{\sigma}^2),$$

which is nothing more than the product measure for two independent Curie-Weiss networks, where one network contains all sites at which $\xi_x^1 = \xi_x^2$, and the other contains all sites at which $\xi_x^1 = -\xi_x^2$. The standard 2-pattern Hopfield network with orthogonal patterns is just two independent equally sized copies of a Curie-Weiss network!

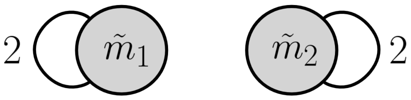


Figure 14: The 2-gCW network that corresponds to the 2-pattern Hopfield network is just two independent Curie-Weiss networks, both with coupling $\mathcal{M}_{11} = \mathcal{M}_{22} = 2$, and no cross-couplings.

Example 3.8 (Van Hemmen network)**.** *The last network we will analyze through the correspondence is a network proposed by Van Hemmen in 1982 [53] and further analyzed in [54]; it was introduced with the intention to study its spin-glass properties. We will consider his network with the ferromagnetic interaction set to zero, and we assume the patterns to be orthogonal.*

The interaction matrix is

$$(Q)_{ij} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

so the weights of this network only consist of crossterms of the form $\xi_x^1 \xi_y^2$ and $\xi_x^2 \xi_y^1$.

The partitions and the matrix A are the same as in Example 3.7, and so the interaction matrix of the corresponding 2-gCW network is

$$(\mathcal{M})_{ij} = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}.$$

We could do exactly the same analysis as done in Example 3.7, with the only difference that $\mathcal{M}_{22}$ is now negative instead of positive. Again, the network thus consists of two independent and equally sized copies of a Curie-Weiss (CW) network, but the first CW network is a ferromagnet (positive coupling) while the second CW network is an antiferromagnet (negative coupling).

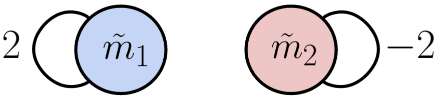


Figure 15: The 2-gCW network that corresponds to the Van Hemmen network (with ferromagnetic interaction set to zero) consists of two independent Curie-Weiss networks, one with coupling $\mathcal{M}_{11} = 2$, and the other with coupling $\mathcal{M}_{22} = -2$.

We saw that Hopfield networks with only two patterns usually don't have interesting properties, as they appear to just be products of independent Curie-Weiss models. But we considered only either self-couplings or cross-couplings; what if we combine both the 'standard' 2-pattern Hopfield network and the Van Hemmen network? In the next Chapter, we will investigate such networks, where both the self-couplings and cross-couplings are present.

4 Embedding a two-state Markov chain in an ANN

We wish to embed an arbitrary two-state Markov chain into an attractor neural network. An arbitrary two-state Markov chain has two states, which we will simply call `state 1` and `state 2`, and four possible transitions: `state 1` → `state 1`, `state 1` → `state 2`, `state 2` → `state 1` and `state 2` → `state 2`. These four transitions will have their own probabilities, which we denote as P_{11}, P_{12}, P_{21} and P_{22} respectively; these are the entries of the 2×2 transition matrix P. For both `state 1` and `state 2`, the outgoing probabilities sum to 1.

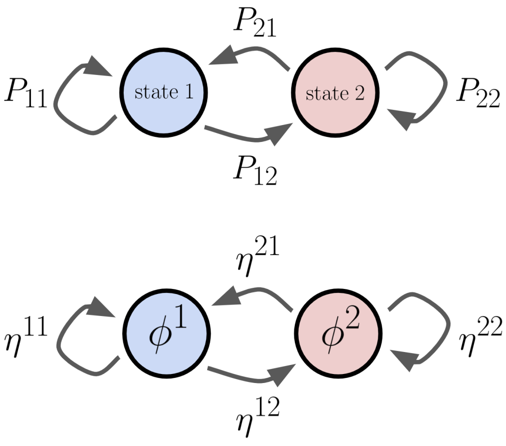


Figure 16: The most general two-state Markov chain has two states, and four possible transitions. Top: the Markov chain with two states, and transition matrix P. Bottom: the attractor states associated with each part of the Markov chain.

The Markov chain will be embedded in an ANN as follows. `state 1` and `state 2` are stored in the ANN as attractor states ϕ^1 and ϕ^2 respectively. Each edge is also stored as an attractor state: for $i, j = 1, 2$, the edge `state` i → `state` j will be stored as an attractor η^{ij}. We call these patterns *edge patterns.*

Each timestep of the embedded Markov chain looks as follows.

1. The current state of the chain is `state` i: the ANN is in attractor state ϕ^i. A single 'push' parameter $\delta^{\uparrow}$ (independent of the current chain state) is turned on, and the ANN transitions from attractor state ϕ^i to η^{ii}. The parameter is turned off again.

2. The ANN will stochastically transition between attractor states η^{ii} and η^{ij} (where $j \neq i$) for a fixed time T. We call this time the *switching time*, and we say the network *switches between edge patterns* η^{ii} and η^{ij}.

3. After time T, a single 'pull' parameter $\delta^{\downarrow}$ (independent of the current chain state) is turned on. If the ANN is in attractor state η^{ii}, the ANN transitions to ϕ^i. If the ANN is in attractor state η^{ij}, the ANN transitions to ϕ^j. The parameter is turned off again. The new state of the chain is `state` i if the ANN is in ϕ^i, or `state` j if the ANN is in ϕ^j.

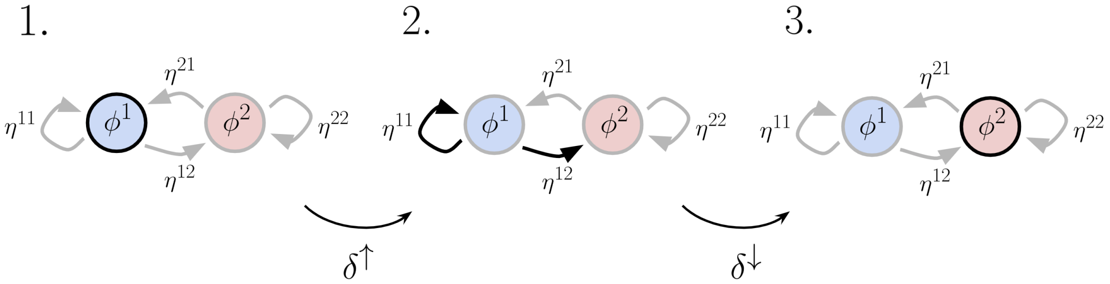


Figure 17: Example of a timestep in the embedded Markov chain. In this case, the embedded Markov chain jumps from `state 1` to `state 2`.

We can implement this behaviour into the ANNs considered in this work, by defining the *embedding weights* of the networks as follows:

$$w_{xy} = w_{xy}^{\text{states}} + \delta^{\uparrow} w_{xy}^{\text{push}} + \delta^{\downarrow} w_{xy}^{\text{pull}} + w_{xy}^{\text{switch1}} + w_{xy}^{\text{switch2}}, \tag{16}$$

where

$$w_{xy}^{\text{states}} = \phi_x^1 \phi_y^1 + \phi_x^2 \phi_y^2$$

stores the attractor states corresponding to `state 1` and `state 2`,

$$w_{xy}^{\text{push}} = \eta_x^{11} \phi_y^1 + \eta_x^{22} \phi_y^2$$

is an asymmetric weight that transitions the network to attractor state η^{11} if the network is in ϕ^1 or to η^{22} if the network is in ϕ^2,

$$w_{xy}^{\text{pull}} = \phi_x^1(\eta_y^{11} + \eta_y^{21}) + \phi_x^2(\eta_y^{22} + \eta_y^{12})$$

is an asymmetric weight that transitions the network to attractor state ϕ^1 if the network is in η^{11} or η^{21}, and to ϕ^2 if the network is in η^{22} or η^{12}.

We will define the precise form of the weights w_{xy}^{switch1} and w_{xy}^{switch2} in the next Chapter. While w_{xy}^{switch1} contains attractor states η^{11}, η^{12}, and w_{xy}^{switch2} contains attractor states η^{21}, η^{22}, both weights are constructed in the same way. We will write this as $w_{xy}^{\text{switch1}} = w_{xy}^{\text{switch}}[\eta^{11}, \eta^{12}]$ and $w_{xy}^{\text{switch2}} = w_{xy}^{\text{switch}}[\eta^{21}, \eta^{22}]$, where w_{xy}^{switch} is the general form of the weights, and the attractor states used to make the weights are explicitly denoted.

The weights w_{xy}^{switch} make the network stochastically transition (or 'switch') between the attractor states η used to construct them, whenever the network is in one of such attractor states. These weights can be biased to make the network favor one attractor state over the other. More explicitly, the probability to find the network in the favoured attractor state during the switching time T is larger than the probability to find the network in the less favoured attractor state. This allows us to embed a Markov chain with a custom transition matrix P.

The switching time T should be long enough for the stochastic transitioning process between the η atrractor states to reach equilibrium. We might for example consider the situation when the embedded Markov chain is in `state 1`, that is, the network is in attractor state ϕ^1. To start the next timestep of the Markov chain, we turn $\delta^{\uparrow}$ on and then off again, and the network transitions to η^{11}. If T is very small, we would now almost immediately complete the timestep again and turn on $\delta^{\downarrow}$. The network will not have had time to transition at least once between η^{11} and η^{12}, and we end up in ϕ^1 (`state 1`) for certain. Like most stochastic processes, the stochastic transitioning process between the η attractor states needs a burn-in period in order for us to fully use its stochastic behaviour.

If we forget about all other weights, the weights w_{xy}^{switch} on themselves also define an ANN. We will first fully analyze a Hopfield network with weights w_{xy}^{switch}, which we call the '2-pattern switching network'. One might then hope that much of the results we find for this network still hold when we combine two of such weight terms ('switch1', 'switch2') and add the 'states', 'push' and 'pull' weights of (16). Unfortunately this is not the case for Hopfield networks; we will see why in Section 5.5. However, the results we obtain about the behaviour of the 2-pattern switching network will still be very valuable as guidelines for the design process of a sparse switching network. The sparse switching network *does* keep its behaviour when the other weights of (16) are added, which will be demonstrated with a heuristic mathematical argument. Unfortunately the sparse switching network is more complicated mathematically, so in this work we only use simulations to further analyze the sparse switching network.

5 2-pattern switching network

We saw in the Introduction that the standard two-pattern Hopfield network has two phases: an *ordered* phase and a *disordered* phase. In the ordered phase, once the Hopfield network has obtained largest overlap with one of the patterns, it will keep its largest overlap with that pattern indefinitely.

We present weights, which are defined through an interaction matrix, that introduces a new phase to the network: a *mixed* phase. The resulting network will transition stochastically between the patterns when the network parameters are just below the critical values of the phase transition between the ordered and the mixed phase. We show what the behaviour of the network looks like in this regime in Section 5.3.

We abuse our nomenclature a bit, and say that the 2-pattern switching network is in state 1 if the state of the network $\boldsymbol{m}$ satisfies $|m_1| \geq |m_2|$ (the network configuration has larger overlap with the first pattern than with the second pattern) and it is in state 2 if $|m_2| > |m_1|$ (we break the tie arbitrarily in favor of state 1).

5.1 Definition

The 2-pattern switching network is given by the interaction matrix

$$(Q)_{ij} = \begin{pmatrix} 1+\gamma & \alpha \\ \alpha & 1 \end{pmatrix},$$

where $\alpha \in [0,1)$ is the *crossterm parameter*, and $\gamma \in [0,1)$ is the *energy perturbation* (which is typically very small). The weights of the network are

$$N \cdot w_{xy}^{\text{switch}}[\xi^1, \xi^2] = \underbrace{\xi_x^1\xi_y^1 + \xi_x^2\xi_y^2}_{\text{standard Hopfield}} + \underbrace{\gamma\xi_x^1\xi_y^1}_{\text{perturb.}} + \underbrace{\alpha(\xi_x^1\xi_y^2 + \xi_x^2\xi_y^1)}_{\text{crossterm}}.$$

The intuition behind introducing the crossterm α for attractor state switching is as follows: as asymmetric terms of the form $\xi_x^j\xi_y^i$ make a network in attractor state ξ^i transition to attractor state ξ^j, a combination of two asymmetric terms might make the network oscillate between attractor states ξ^i and ξ^j. Most importantly, this implementation results in synaptic weights that are symmetric in x and y, and this allows us to use a Hamiltonian (and as a consequence, statistical mechanics) to analyze the model.

The energy perturbation γ also has an intuitive interpretation. γ decreases the energy of attractor state ξ^1, and so the probability of finding the network in state 1 increases under the Gibbs distribution. Of course, if one would like the probability of finding state 2 to be larger than the probability of finding state 1, the perturbation term in w_{xy}^{switch} can be chosen to be $\gamma\xi_x^2\xi_y^2$ instead of $\gamma\xi_x^1\xi_y^1$.

The existence of a mixed phase can be directly observed from the weights. If we set $\gamma = 0$, we can immediately see that for $\alpha = 0$, the attractors are ξ^1 and ξ^2, while for $\alpha = 1$ the weights become

$$N \cdot w_{xy}^{\text{switch}} = \xi_x^1\xi_y^1 + \xi_x^2\xi_y^2 + \xi_x^1\xi_y^2 + \xi_x^2\xi_y^1 = (\xi_x^1 + \xi_x^2)(\xi_y^1 + \xi_y^2),$$

and so the attractor appears to be $\xi^1 + \xi^2$; it is a *mixture* of the first and the second pattern. Of course $\xi^1 + \xi^2$ is not a valid configuration, as $\xi_x^1 + \xi_x^2$ can be zero when the patterns don't agree on site x. However, it *is* an attractor in the sense that at the sites y where the two patterns do agree, the spins σ_y converge to $\frac{1}{2}(\xi_y^1 + \xi_y^2) = \xi_y^1 = \xi_y^2$. Spins at sites where the patterns agree will align with the pattern spins, and at sites where the patterns don't agree the spins will randomly flip. We will show this when we analyze the network using its corresponding 2-group Curie-Weiss network.

5.2 The corresponding 2-group Curie-Weiss network

Parts of the analysis of the 2-pattern switching network will rely on the Hopfield/gCW correspondence. By Theorem 3.3, the interaction matrix of the corresponding 2-gCW network is

$$(\mathcal{M})_{ij} = (A^\intercal Q A)_{ij} = \begin{pmatrix} 2+2\alpha+\gamma & \gamma \\ \gamma & 2-2\alpha+\gamma \end{pmatrix}. \tag{17}$$

Again, we denote by $X = |\{x \in V_N : \xi_x^1 = \xi_x^2\}|/N$ the fraction of sites at which the two patterns agree.

Before we start the analysis, let us already introduce the *free energy* of the corresponding 2-gCW network. For $\tilde{\boldsymbol{m}}' \in \operatorname{Im}(\tilde{\boldsymbol{m}})$,

$$\tilde{F}_{\alpha,\beta,\gamma}(\tilde{\boldsymbol{m}}') := -\frac{1}{8}\tilde{\boldsymbol{m}}'^\intercal \mathcal{M} \tilde{\boldsymbol{m}}' - \frac{1}{2\beta}\left(S\left(\tilde{m}_1'\right) + S\left(\tilde{m}_2'\right)\right) \tag{18}$$

where

$$S\left(\tilde{m}_i'\right) := -\frac{1+\tilde{m}_i'}{2}\log\frac{1+\tilde{m}_i'}{2} - \frac{1-\tilde{m}_i'}{2}\log\frac{1-\tilde{m}_i'}{2} \tag{19}$$

is the *entropy* of a partition. Note that the free energy is exactly the function $\tilde{F}$ used to construct the rate function of the Hopfield network in Theorem 3.5. Furthermore, we will use the notation $\tilde{F}_{\alpha,\beta} := \tilde{F}_{\alpha,\beta,\gamma=0}$.

Consider the case when $\gamma = 0$, and $X = 1/2$. We see that the matrix $\mathcal{M}$ is diagonal, just as in Example 3.7, and in exactly the same way as in the example we can deduce that our 2-pattern switching network becomes two independent copies of a Curie-Weiss network, which are defined on their own site partitions.

The first partition contains all sites where $\xi_x^1 = \xi_x^2$ and has self-coupling $\mathcal{M}_{11} = 2 + 2\alpha$, and the other partition contains all sites at which $\xi_x^1 = -\xi_x^2$ and has self-coupling $\mathcal{M}_{11} = 2 - 2\alpha$. The Curie-Weiss Gibbs measure of the first network is then

$$\rho^1_{N/2,\beta}(\tilde{\sigma}^1) = \frac{1}{\tilde{Z}^1_{N/2,\beta}} e^{\beta \frac{N}{4}(1+\alpha)\tilde{m}_1(\tilde{\sigma}^1)^2}, \tag{20}$$

and of the second network is

$$\rho^2_{N/2,\beta}(\tilde{\sigma}^2) = \frac{1}{\tilde{Z}^2_{N/2,\beta}} e^{\beta \frac{N}{4}(1-\alpha)\tilde{m}_2(\tilde{\sigma}^2)^2}, \tag{21}$$

where the notation is as in Example 3.7.

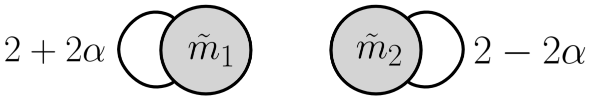


Figure 18: The 2-gCW network that corresponds to the 2-pattern switching network with $\gamma = 0$ consists of two independent Curie-Weiss networks, one with coupling $\mathcal{M}_{11} = 2 + 2\alpha$, and the other with coupling $\mathcal{M}_{22} = 2 - 2\alpha$.

Let us get some intuition. If $\alpha = 0$, we get back the standard 2-pattern Hopfield network. However, if $\alpha = 1$, then the self-coupling of the second network is zero, and $\rho^2_{N/2,\beta}$ becomes the uniform probability measure (all states are equally likely, no matter the value of β). This means that any site for which $\xi_x^1 = -\xi_x^2$ will randomly flip during dynamics. If also β is large enough (temperatures are low), then the first network (containing all sites at which $\xi_x^1 = \xi_x^2$) will be a Curie-Weiss network with strong self-coupling in the ordered phase, and so $\tilde{m}_1$ will tend to 1, which means that $\tilde{\sigma}^1$ will tend to the all-one configuration, which in turn means that σ^1 will tend to the attractor ξ^1 ($= \xi^2$ on the first partition). We see hints of a new phase in the network, namely a mixed phase, in which half of the sites randomly flip, and the other half freeze in the attractor state. We will rigorously analyze the phase diagram of our 2-pattern switching network in the next subsection.

5.3 Phase diagram

To investigate the effect of the parameter α more thoroughly, we derive the phase diagram of the 2-pattern switching network through analytical methods and simulation. To simplify analysis we set the (small) energy perturbation $\gamma = 0$ in this Section. The exact derived phase diagram can be seen in figure 19, along with typical behaviour of the order parameters below. In the disordered phase (I), the expected overlaps of the network with patterns 1 and 2 are both zero. In the mixed phase (II), the expected overlaps of the network with patterns 1 and 2 are equal, and between 0 and 0.5. In the ordered phase (III), the expected overlaps of the network with pattern 1 is larger than the expected overlap with pattern 2, but both are nonzero. Of particular interest to us will be the region of parameter values just below the boundary between the mixed and the ordered phase. Here, simulations reveal stochastic switching between the overlaps of patterns 1 and 2 (see the 'simulation'-subsection).

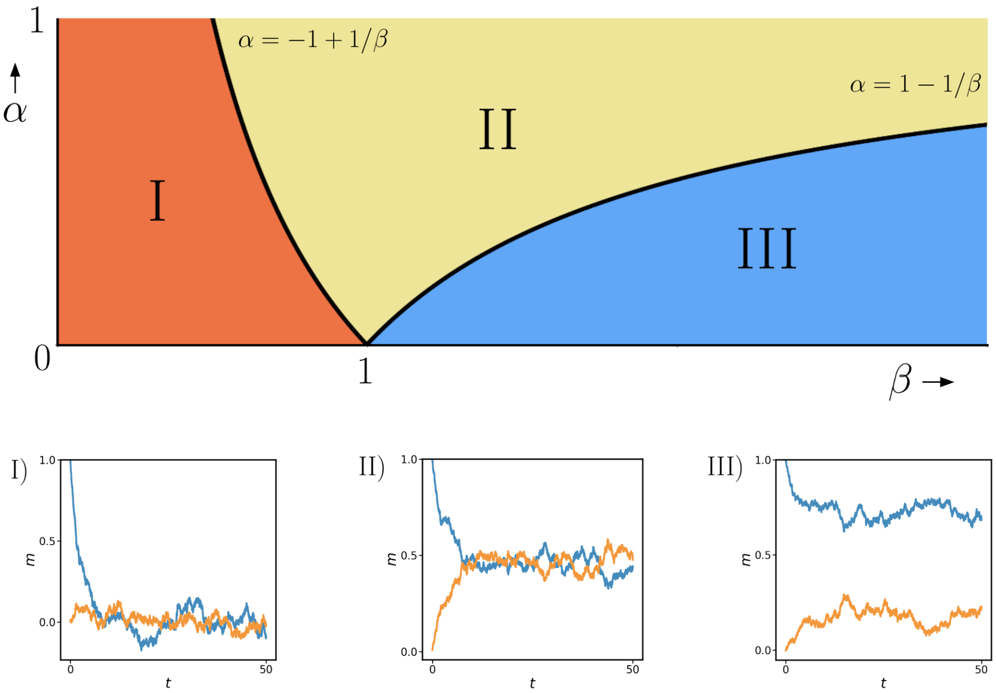


Figure 19: Top: the three phases of the 2-pattern switching network, and their regimes. Bottom: time series of the order parameters m_1 (blue) and m_2 (orange), which show the typical behaviour of the network order parameters in the corresponding phases, where the dynamics is started in full overlap with state 1 and zero overlap with state 2. I) *Disordered phase.* The order parameters fluctuate around 0, so the network is in neither state 1 or 2. II) *Mixed phase.* The network has equal overlap with both state 1 and 2 at the same time. III) *Ordered phase.* The network always has the largest overlap with the state in which the dynamics was started (which was state 1 in this case).

The phase diagram of the 2-pattern switching network can be derived in multiple ways. The first is to make a small generalization of the techniques used for the standard Hopfield model, which is to do a Hubbard-Stratanovich transformation of the partition function, and to extract mean-field equations from the free energy using Laplace's method [15] [33]. The other technique that we will consider is to make use of the Hopfield/gCW correspondence. The phase diagram of the 2-pattern switching network can directly be extracted from Theorem 3.5. If we ignore the technicalities of the convergence of patterns to orthogonality in the large N limit, but just assume they will be orthogonal for large N, then we can actually immediately obtain the phase diagram by just looking at the measures (20) and (21) without using Theorem 3.5. Let's start with this heuristic approach, and after that we derive the phase diagram in full rigour with the strategies described above.

5.3.1 Heuristic approach

It is well-known that the Curie-Weiss network with the Gibbs distribution

$$\rho_{n,\beta}(\tilde{\sigma}) = \frac{1}{\tilde{Z}_{n,\beta}} e^{\beta \frac{n}{2} \tilde{m}(\tilde{\sigma})^2}$$

in the large N limit has a phase transition at $\beta = 1$; $\beta < 1$ is the disordered phase and $\beta > 1$ is the ordered phase. If we define $\beta_1' := (1+\alpha)\beta$, the distribution (20) is exactly the Curie-Weiss Gibbs distribution but with inverse temperature β_1', instead of β. The first network will thus be disordered when $\beta_1' < 1$, i.e. $\alpha < \frac{1}{\beta} - 1$, and ordered when $\alpha > \frac{1}{\beta} - 1$. Similarly, we define $\beta_2' := (1-\alpha)\beta$. The second network will be disordered when $\beta_2' < 1$, i.e. $\alpha > 1 - \frac{1}{\beta}$, and ordered when $\alpha < 1 - \frac{1}{\beta}$.

As $\alpha \geq 0$, we cannot have that α is smaller than both $\frac{1}{\beta} - 1$ and $1 - \frac{1}{\beta}$ at the same time, i.e, $\alpha < \min\{\frac{1}{\beta} - 1, 1 - \frac{1}{\beta}\}$. We are left with three options:

	Network 1	Network 2
$\frac{1}{\beta} - 1 < \alpha < 1 - \frac{1}{\beta}$	disordered	disordered
$\alpha > \max\{\frac{1}{\beta} - 1, 1 - \frac{1}{\beta}\}$	ordered	disordered
$1 - \frac{1}{\beta} < \alpha < \frac{1}{\beta} - 1$	ordered	ordered

and we neatly see the three phases of the 2-pattern switching network, and their regimes. We will now look at the derivation of the phase diagram more carefully.

5.3.2 Using Hubbard-Stratanovich

We're going to find the average values of the order parameters m_1, m_2 by formulating a *self-consistency equation.* This is an equation of the form $\boldsymbol{m} = \text{func}(\boldsymbol{m})$, which can have a unique or multiple solutions for $\boldsymbol{m}$. We find such an equation by evaluating the partition function and the corresponding free energy.

Let us first introduce a little bit of notation. Define

$$\boldsymbol{b} := \left(\sum_{x \in V_N} \xi^1_x \sigma_x, \sum_{x \in V_N} \xi^2_x \sigma_x \right)^\intercal,$$

where we realize that $\boldsymbol{b}$ depends on σ (we will not note this down explicitly, to lighten notation), and that it thus is a random variable.

$$H^0_{N,\alpha}(\sigma) := -\frac{1}{2N} \boldsymbol{b}^\intercal Q \boldsymbol{b}$$

and

$$Z^0_{N,\alpha,\beta} := \sum_{\sigma \in \Omega} e^{-\beta H^0_{N,\alpha}(\sigma)},$$

where in Q, the energy perturbation parameter $\gamma = 0$.

Furthermore, we introduce an *external field*

$$\boldsymbol{h} := (h_1, h_2)^\intercal \in \mathbb{R}^2,$$

and

$$H^{\text{ext}}_{N,\boldsymbol{h}}(\sigma) := -\boldsymbol{h}^\intercal \cdot \boldsymbol{b}.$$

Let the combined Hamiltonian be

$$\mathcal{H}_{N,\alpha,\boldsymbol{h}}(\sigma) := H^0_{N,\alpha}(\sigma) + H^{\text{ext}}_{N,\boldsymbol{h}}(\sigma),$$

and the partition function of this Hamiltonian is

$$\mathcal{Z}_{N,\alpha,\beta,\boldsymbol{h}} := \sum_{\sigma \in \Omega} e^{-\beta \mathcal{H}_{N,\alpha,\boldsymbol{h}}(\sigma)} = \sum_{\sigma \in \Omega} e^{-\beta H^0_{N,\alpha}(\sigma) - \beta(h_1 b_1 + h_2 b_2)}$$

Lastly, we define the *free energy*

$$F_{N,\alpha,\beta,\boldsymbol{h}} = -\frac{1}{\beta N} \log \mathcal{Z}_{N,\alpha,\beta,\boldsymbol{h}}. \tag{22}$$

Now, observe the following key property of the free energy:

$$\begin{aligned}
\left[\frac{\partial}{\partial h_i} F_{N,\alpha,\beta,\boldsymbol{h}} \right]_{\boldsymbol{h}=0} &= \left[-\frac{\partial}{\partial h_i} \frac{1}{\beta N} \log \mathcal{Z}_{N,\alpha,\beta,\boldsymbol{h}} \right]_{\boldsymbol{h}=0} \\
&= \left[\sum_{\sigma \in \Omega} \left(\frac{1}{N} \sum_{x \in V_N} \xi^i_x \sigma_x \right) \frac{e^{-\beta \mathcal{H}_{N,\alpha,\boldsymbol{h}}(\sigma)}}{\mathcal{Z}_{N,\alpha,\beta,\boldsymbol{h}}} \right]_{\boldsymbol{h}=0} \\
&= \sum_{\sigma \in \Omega} m_i(\sigma) \frac{e^{-\beta \mathcal{H}^0_{N,\alpha(\sigma)}}}{Z^0_{N,\alpha,\beta}} \\
&:= \langle m_i \rangle_{N,\alpha,\beta},
\end{aligned}$$

where $\langle \cdot \rangle$ is the expected value.

$\mathcal{Z}$ is difficult to compute, as it contains terms of the form $\exp\left[(\sum_x \xi^i_x \sigma_x)^2 \right]$, which is not straightforward to sum over. We linearize these terms with a generalization of the *Hubbard-Stratanovich transformation.* We use that for any $\boldsymbol{v} \in \mathbb{R}^2$ and $A \in \mathbb{R}^{2\times 2}$,

$$\underbrace{e^{\frac{1}{2}\boldsymbol{a}^\intercal A \boldsymbol{a}}}_{\text{quadratic in } \boldsymbol{a}} = \frac{1}{\sqrt{\det(2\pi A)}} \int_{\mathbb{R}^2} \mathrm{d}^2 v \, \underbrace{\exp\left[-\frac{1}{2} \boldsymbol{v}^\intercal A^{-1} \boldsymbol{v} + \boldsymbol{v}^\intercal \cdot \boldsymbol{a} \right]}_{\text{linear in } \boldsymbol{a}}. \tag{23}$$

Now we evaluate.

$$
\begin{aligned}
\mathcal{Z}_{N,\alpha,\beta,\boldsymbol{h}} &= \sum_{\sigma\in\Omega} e^{-\frac{1}{2N}\beta\boldsymbol{b}^\intercal Q\boldsymbol{b}-\beta\boldsymbol{h}^\intercal\cdot\boldsymbol{b}} \\
&= \sum_{\sigma\in\Omega} \underbrace{e^{-\frac{1}{2}(\beta\boldsymbol{b}^\intercal)\left(\frac{1}{\beta N}Q\right)(\beta\boldsymbol{b})}}_{*} e^{-\beta\boldsymbol{h}^\intercal\cdot\boldsymbol{b}}.
\end{aligned}
$$

Define $B=\frac{1}{\beta N}Q$. Then,

$$
B^{-1}=\frac{N\beta}{(\alpha^2-1)}\begin{pmatrix}-1 & \alpha\\ \alpha & -1\end{pmatrix}
$$

Simplify notation and let the constant $c=\frac{1}{\sqrt{\det(2\pi B)}}$. Using the transformation (23) on $(*)$,

$$
\begin{aligned}
\mathcal{Z}_{N,\alpha,\beta,\boldsymbol{h}} &= \sum_{\sigma\in\Omega} c\int_{\mathbb{R}^2} \mathrm{d}^2v\ \exp\left[-\frac{1}{2}\boldsymbol{v}^\intercal B^{-1}\boldsymbol{v}+\beta\boldsymbol{v}^\intercal\cdot\boldsymbol{b}+\beta\boldsymbol{h}^\intercal\cdot\boldsymbol{b}\right] \\
&= c\int_{\mathbb{R}^2} \mathrm{d}^2v\ \exp\left[-\frac{1}{2}\boldsymbol{v}^\intercal B^{-1}\boldsymbol{v}\right] \underbrace{\sum_{\sigma\in\Omega}\exp\left[\beta(\boldsymbol{v}^\intercal+\boldsymbol{h}^\intercal)\cdot\boldsymbol{b}\right]}_{**}.
\end{aligned} \tag{24}
$$

Defining $\boldsymbol{\xi}_x := (\xi_x^1,\xi_x^2)$, and recalling the definition of $\boldsymbol{b}$,

$$
\begin{aligned}
(**) &= \sum_{\sigma\in\Omega}\exp\left[\beta(\boldsymbol{v}^\intercal+\boldsymbol{h}^\intercal)\cdot\left(\sum_{x\in V_N}\boldsymbol{\xi_x}\sigma_x\right)\right] \\
&= \sum_{\sigma\in\Omega}\prod_{x=1}^{N}\exp\left[\beta(\boldsymbol{v}^\intercal+\boldsymbol{h}^\intercal)\cdot\boldsymbol{\xi_x}\sigma_x\right] \\
&= \prod_{x=1}^{N}\sum_{\sigma_x=\pm1}\exp\left[\beta(\boldsymbol{v}^\intercal+\boldsymbol{h}^\intercal)\cdot\boldsymbol{\xi_x}\sigma_x\right] \\
&= \prod_{x=1}^{N}2\cosh\left[\beta(\boldsymbol{v}^\intercal+\boldsymbol{h}^\intercal)\cdot\boldsymbol{\xi_x}\right].
\end{aligned}
$$

Plugging this back into (24), we get

$$
\mathcal{Z}_{N,\alpha,\beta,\boldsymbol{h}} = c\int_{\mathbb{R}^2}\mathrm{d}^2v\ \exp\left[-\beta N g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v})\right],
$$

where

$$
g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v}) := \frac{1}{2}\boldsymbol{v}^\intercal B^{-1}\boldsymbol{v}-\frac{1}{\beta N}\sum_{x=1}^{N}\log\left[2\cosh[\beta(\boldsymbol{v}^\intercal+\boldsymbol{h}^\intercal)\cdot\boldsymbol{\xi_x}]\right].
$$

We can simplify this using the fact that the patterns *self-average* over sites. Recall that the patterns ξ_x^1,ξ_x^2 are i.i.d. Ber(1/2) random variables on $\{-1,1\}$ for all $x\in V_N$, and so for all $x\in V_N$ the four possible values $\Xi=\{(-1,-1)^\intercal,(-1,1)^\intercal,(1,-1)^\intercal,(1,1)^\intercal\}$ of $\boldsymbol{\xi}_x$ (which are i.i.d.) occur each with probability 1/4. Therefore, if we average over all sites, we expect to see each pattern with probability 1/4, and so for any bounded function f of the patterns at a site x,

$$
\frac{1}{N}\sum_{x=1}^{N}f(\boldsymbol{\xi}_x)=\frac{1}{N}N\left(\frac{1}{4}f((-1,-1)^\intercal)+...+\frac{1}{4}f((1,1)^\intercal)\right)=\sum_{\boldsymbol{\xi}\in\Xi}\mathbb{P}(\boldsymbol{\xi})f(\boldsymbol{\xi}) := \langle\!\langle f(\boldsymbol{\xi})\rangle\!\rangle,
$$

where the last expression is called the *self-average.*

Thus,

$$
g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v}) = \frac{1}{2}\boldsymbol{v}^\intercal B^{-1}\boldsymbol{v}-\frac{1}{\beta}\langle\!\langle\log\left[2\cosh[\beta(\boldsymbol{v}^\intercal+\boldsymbol{h}^\intercal)\cdot\boldsymbol{\xi_x}]\right]\rangle\!\rangle.
$$

To evaluate the free energy (22) in the limit $N\to\infty$, we use Laplace's method. In our case it states that

$$
\lim_{N\to\infty}-\frac{1}{N\beta}\log\int_{\mathbb{R}^2}\mathrm{d}^2v\ e^{-N\beta g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v})} = g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v}_0),
$$

where $\boldsymbol{v}_0 \in \mathbb{R}^2$ is the unique minimum of $g_{\alpha_\beta,\boldsymbol{h}}$. One can check that $g_{\alpha_\beta,\boldsymbol{h}}$ indeed has a unique minimum. So,

$$F_{\alpha,\beta,\boldsymbol{h}} := \lim_{N\to\infty} F_{N,\alpha,\beta,\boldsymbol{h}} = s,$$

where $\boldsymbol{v}^0$ is the solution of

$$\begin{cases} \dfrac{\partial}{\partial v_1} g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v}) = 0 \\ \dfrac{\partial}{\partial v_2} g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v}) = 0 \end{cases} \tag{25}$$

Filling in the expression of g, we find the set of equations

$$\begin{cases} \dfrac{1}{1-\alpha^2}(v_1^0 - \alpha v_2^0) - \langle\langle \xi_1 \tanh[\beta(\boldsymbol{v}^\intercal + \boldsymbol{h}^\intercal)\cdot\boldsymbol{\xi}]\rangle\rangle = 0 \\ \dfrac{1}{1-\alpha^2}(v_2^0 - \alpha v_1^0) - \langle\langle \xi_2 \tanh[\beta(\boldsymbol{v}^\intercal + \boldsymbol{h}^\intercal)\cdot\boldsymbol{\xi}]\rangle\rangle = 0 \end{cases} \tag{26}$$

The average of the order parameters can then be found from the free energy:

$$\begin{aligned} \langle m_1\rangle_{\alpha,\beta} &:= \lim_{N\to\infty} \langle m_1\rangle_{N,\alpha,\beta} \\ &= \left[\frac{\partial}{\partial h_1} F_{\alpha,\beta,\boldsymbol{h}}\right]_{\boldsymbol{h}=0} \\ &= \left[\frac{\partial}{\partial h_1} g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v}^0)\right]_{\boldsymbol{h}=0} \\ &= \langle\langle \xi_1 \tanh[\beta(\boldsymbol{v}^0)^\intercal \cdot \boldsymbol{\xi}]\rangle\rangle \\ &= \frac{1}{1-\alpha^2}(v_1^0 - \alpha v_2^0). \end{aligned}$$

Similarly,

$$\langle m_1\rangle_{\alpha,\beta} = \frac{1}{1-\alpha^2}(v_2^0 - \alpha v_1^0).$$

The equations in (26) are coupled. We can decouple them with by introducing new variables (note the parallels with the gCW order parameters!)

$$\tilde{m}_1 = m_1 + m_2, \qquad \tilde{m}_2 = m_1 - m_2.$$

Then,

$$\langle \tilde{m}_1\rangle_{\alpha,\beta} = \langle m_1\rangle_{\alpha,\beta} + \langle m_2\rangle_{\alpha,\beta} = \frac{1}{1+\alpha}(v_1^0 + v_2^0),$$

and similarly

$$\langle \tilde{m}_2\rangle_{\alpha,\beta} = \frac{1}{1-\alpha}(v_1^0 - v_2^0).$$

From (25) we get

$$\begin{cases} \dfrac{\partial}{\partial v_1} g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v}) + \dfrac{\partial}{\partial v_2} g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v}) = 0 \\ \dfrac{\partial}{\partial v_1} g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v}) - \dfrac{\partial}{\partial v_2} g_{\alpha_\beta,\boldsymbol{h}}(\boldsymbol{v}) = 0 \end{cases}$$

resulting in the set of equations

$$\begin{cases} \dfrac{1}{1+\alpha}(v_1^0 + v_2^0) - \langle\langle (\xi_1 + \xi_2) \tanh[\beta(v_1^0\xi_1 + v_2^0\xi_2)]\rangle\rangle = 0 \\ \dfrac{1}{1-\alpha}(v_1^0 - v_2^0) - \langle\langle (\xi_1 - \xi_2) \tanh[\beta(v_1^0\xi_1 + v_2^0\xi_2)]\rangle\rangle = 0 \end{cases} \tag{27}$$

Now we evaluate the self-averages.

$$\begin{aligned} \langle\langle (\xi_1 + \xi_2) \tanh[\beta(v_1^0\xi_1 + v_2^0\xi_2)]\rangle\rangle &= \frac{1}{4}\sum_{\xi_1=\pm1}\sum_{\xi_2=\pm1} (\xi_1 + \xi_2) \tanh[\beta(v_1^0\xi_1 + v_2^0\xi_2)] \\ &= \tanh[\beta(v_1^0 + v_2^0)]. \end{aligned}$$

Similarly,

$$\langle\langle (\xi_1 + \xi_2) \tanh[\beta(v_1^0\xi_1 + v_2^0\xi_2)]\rangle\rangle = \tanh[\beta(v_1^0 - v_2^0)].$$

We combine this with the definitions of $\tilde{m}_1, \tilde{m}_2$ and (27) to find the mean-field equations:

$$\begin{cases} \langle \tilde{m}_1 \rangle_{\alpha,\beta} = \tanh[\beta(1+\alpha)\langle \tilde{m}_1 \rangle_{\alpha,\beta}] \\ \langle \tilde{m}_2 \rangle_{\alpha,\beta} = \tanh[\beta(1-\alpha)\langle \tilde{m}_2 \rangle_{\alpha,\beta}] \end{cases} \tag{28}$$

Both mean-field equations will always have a trivial solution: $\langle \tilde{m}_1 \rangle_{\alpha,\beta} = \langle \tilde{m}_2 \rangle_{\alpha,\beta} = 0$.

I) If both $\beta(1+\alpha) \le 1$ and $\beta(1-\alpha) \le 1$, the mean-field equations only have a trivial solution.

II) If $\beta(1+\alpha) > 1$ but $\beta(1-\alpha) \le 1$, only $\langle \tilde{m}_1 \rangle_{\alpha,\beta}$ can take nontrivial values. Moreover, as $m_1 = \frac{1}{2}(\tilde{m}_1 + \tilde{m}_2)$, $m_2 = \frac{1}{2}(\tilde{m}_1 - \tilde{m}_2)$, and $\langle \tilde{m}_2 \rangle_{\alpha,\beta} = 0$, we get $\langle m_1 \rangle_{\alpha,\beta} = \langle m_2 \rangle_{\alpha,\beta} = \frac{1}{2}\langle \tilde{m}_1 \rangle_{\alpha,\beta} \neq 0$.

III) If both $\beta(1+\alpha) > 1$ and $\beta(1-\alpha) > 1$, $\langle \tilde{m}_1 \rangle_{\alpha,\beta}$ and $\langle \tilde{m}_2 \rangle_{\alpha,\beta}$ can both take nontrivial values: $\langle m_1 \rangle_{\alpha,\beta} \neq \langle m_2 \rangle_{\alpha,\beta} \neq 0$.

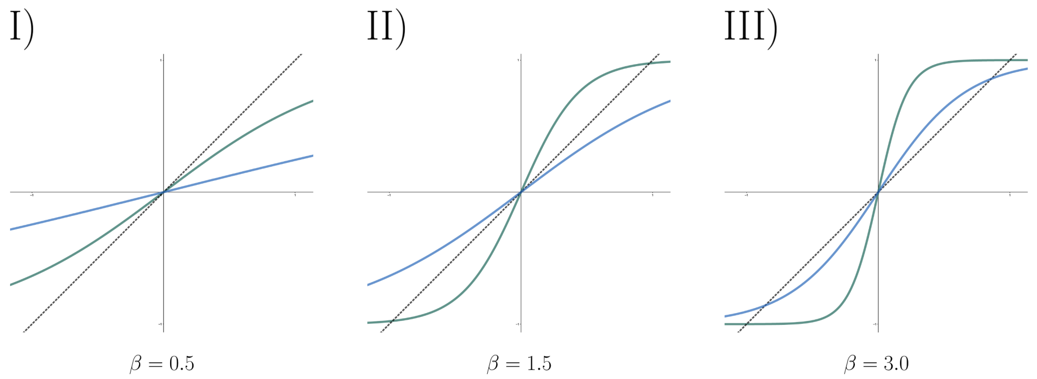


Figure 20: The three different scenarios for the mean-field equations. Green: $y = \tanh[\beta(1+\alpha)x]$, blue: $y = \tanh[\beta(1-\alpha)x]$, black dashed: $y = x$. $\alpha = 2$ for this example.

5.3.3 Using Hopfield/gCW correspondence

We can also show that in different parameter regimes, the marginal probability density function of $\boldsymbol{m}$ concentrates on different sets of states in the limit $N \to \infty$.

By Theorem 3.5, for any $\varepsilon > 0$ there is N large enough such that

$$\frac{1}{N} \log \mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) \in \cdot\,) \le -\inf_{\tilde{\boldsymbol{m}}' \in \mathcal{A}(\cdot)} R_\beta(\tilde{\boldsymbol{m}}') + \varepsilon,$$

which implies that

$$\mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) \in \cdot\,) \le \exp\left[-N\beta\left(-\varepsilon + \inf_{\tilde{\boldsymbol{m}}' \in \mathcal{A}(\cdot)} \tilde{F}_{\alpha,\beta}(\tilde{\boldsymbol{m}}') - \inf_{\tilde{\boldsymbol{m}}' \in [-1,1]^2} \tilde{F}_{\alpha,\beta}(\tilde{\boldsymbol{m}}')\right)\right].$$

Let $\boldsymbol{m}^*$ be the location of any global minimum of $\tilde{F}_{\alpha,\beta}$, and let $\delta > 0$; $|\cdot|$ is any norm on $\mathbb{R}^2$. Now let $\mathcal{A}_\delta = \mathcal{A}(\{\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m}) : |\boldsymbol{m}' - \boldsymbol{m}^*| \ge \delta\})$, and

$$c = c(\beta, \delta, \epsilon) := \beta\left(-\varepsilon + \inf_{\tilde{\boldsymbol{m}}' \in \mathcal{A}_\delta} \tilde{F}_{\alpha,\beta}(\tilde{\boldsymbol{m}}') - \tilde{F}_{\alpha,\beta}(\tilde{\boldsymbol{m}}^*)\right),$$

which is larger than 0 for small enough ε, and increases as δ increases. This shows that there exists a constant $c > 0$ such that

$$\mathbb{P}_{N,\beta}(|\boldsymbol{m}(\sigma) - \boldsymbol{m}^*| \ge \delta) \le e^{-cN}.$$

The probability of finding a state outside the neighborhood of a minimum of $\tilde{F}_{\alpha,\beta}$ vanishes as $N \to \infty$. This means that the probability density function concentrates on the locations of the minima of $\tilde{F}_{\alpha,\beta}$.

A quick analysis shows that in the region $\frac{1}{\beta} - 1 < \alpha < 1 - \frac{1}{\beta}$, $\tilde{F}_{\alpha,\beta}$ has a unique minimum at $(0, 0)$. When $\alpha > \max\{\frac{1}{\beta} - 1, 1 - \frac{1}{\beta}\}$, $\tilde{F}_{\alpha,\beta}$ has two minima, and when $1 - \frac{1}{\beta} < \alpha < \frac{1}{\beta} - 1$, $\tilde{F}_{\alpha,\beta}$ has four minima.

We conclude that the 2-pattern switching network has three different phases. In each phase, the expected value of the network state is different, and the probability concentrates on small parts of the space of allowed states.

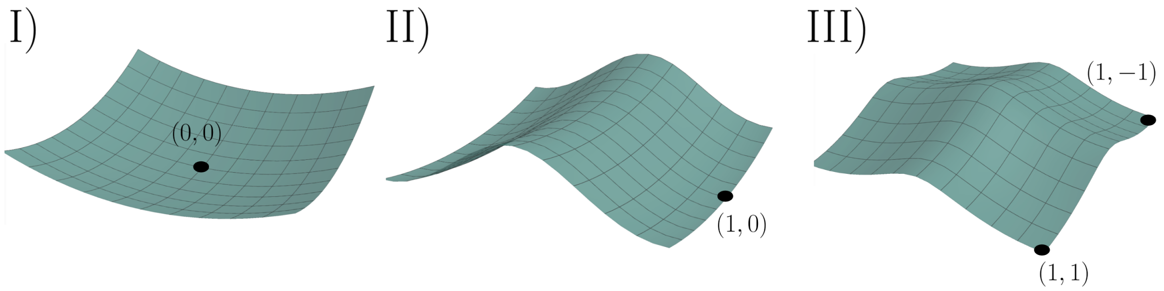


Figure 21: The three different scenarios for the free energy $\tilde{F}_{\alpha,\beta}$.

5.3.4 Simulation

The phase diagram on figure 19 has been simulated, by running a time series of m_1 and m_2 for 50 different values of both α and β, giving a total of $50 \times 50 = 2500$ simulations. The simulations were performed with synchronous dynamics, and $N = 1000$ neurons. For each point (α, β), a new set of two patterns were generated (with maximum overlap of 0.01) and a time series of 1000 timesteps per neuron was run, starting the network in state $m_1 = 1, m_2 = 0$. A burn-in period of 500 timesteps was used, and during the remaining 500 timesteps the values of m_1 and m_2 were stored. In figure 22, on the left, the average of the stored values of $\tilde{m}_1 = m_1 + m_2$ (blue) and $\tilde{m}_2 = m_1 - m_2$ (green) are shown for each point (α, β).

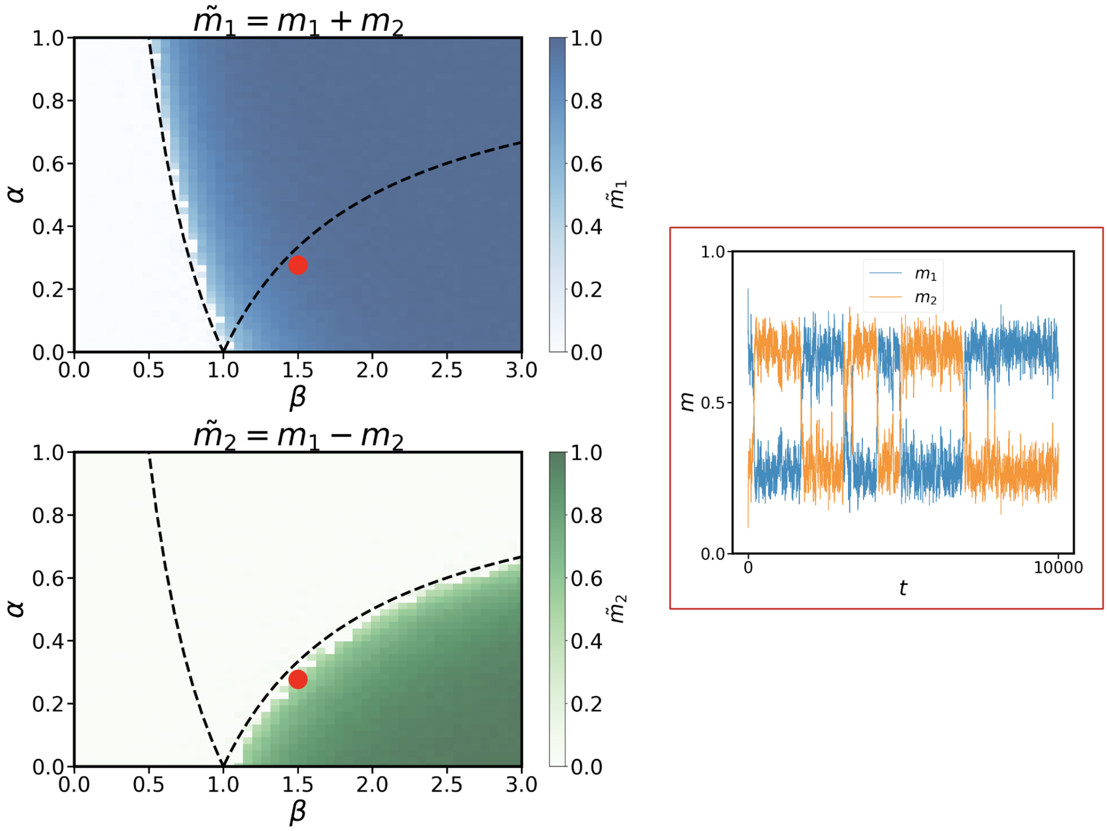


Figure 22: Left: Network simulations for 50 values of α from 0 to 1, and 50 values of β from 0 to 3; in total $50 \times 50 = 2500$ simulations. For each combination of parameters, a simulation of a time series of m_1 and m_2 is performed, and the average overlap is plotted in colour. The black dotted lines are the mathematically derived bounds of the three different phases. Right: time series of the order parameters m_1 (blue) and m_2 (orange), which show the typical behaviour of the network order parameters just below the phase transition of phases II and III. This particular simulation corresponds to $\alpha = 0.29, \beta = 1.5$ (red dot).

The order parameter $\tilde{m}_1$ clearly has two different regimes: it is (almost) zero on the left of the leftmost black dotted line ($\alpha = -1 + 1/\beta$) and larger than zero on the right. This agrees very nicely with theory. The small variations ('rough edges') along the boundary are believed to be due to the finite size of N and the finite simulation time.

The same holds for the order parameter $\tilde{m}_2$: it has two regimes, and the boundary of these regimes agrees with theory, which predicts that the boundary is $\alpha = 1 - 1/\beta$. Again, the small variations along the boundary are believed to be due to the finite size of N and finite simulation time.

This variation in overlap of $\tilde{m}_2$ between points just below the boundary between the mixed and ordered phase is desirable. It means that even for very close neighboring points in this part of the phase diagram, the average overlaps taken over 500 timesteps per neuron can differ quite a lot. This hints at the behaviour we seek for our switching network: a stochastic switching between the two patterns, where in every simulation there is only one pattern with highest overlap (and not a fifty/fifty mixing of both patterns). Furthermore, this switching takes place on timescales longer than 1000 timesteps per neuron, as otherwise the switching would average out and we wouldn't see the same contrast between neighboring points.

In figure 22 on the right, the plot of a single time series at the parameter point ($\alpha = 0.29, \beta = 1.5$) shows the switching behaviour of the network just below the boundary. Notice the timescale: the time series was simulated up to 10000 timesteps per neuron, and we only see the network switch its overlap 6 times.

5.4 Estimate of state probabilities

We derive an estimate for the equilibrium probability density function of the states of the 2-pattern switching network with orthogonal patterns. This estimate becomes accurate in the limit $N \to \infty$. After the derivation, we will use this estimate to obtain a rough expression for the probability that in equilibrium we find the network in either state 1 or 2. From this analysis we se more concretely how the parameter γ influences the network behaviour.

We will assume for our estimate that the two stored patterns ξ^1, ξ^2 are orthogonal, or in other words, the tuple $\boldsymbol{\xi'} := (\xi^1, \xi^2)$ is orthogonal (see Definition 3.1). This simplifies the formulas a bit, and for large N it is a good estimate.

5.4.1 Probability density function estimate

An estimate of the probability density function is most easily obtained using the Hopfield/gCW correspondence. As the two patterns are orthogonal, we have a relative partition sizes vector $\boldsymbol{X'} = (1/2, 1/2)$ and $\mathcal{X} = \text{diag}(1/2, 1/2)$, so that $\mathcal{X}\mathcal{M}\mathcal{X} = \frac{1}{4}\mathcal{M}$. For any state $\boldsymbol{m'} \in \text{Im}(\boldsymbol{m}(\sigma))$, let $\tilde{\boldsymbol{m}}' = A\boldsymbol{m'}$ (with A as in Theorem 3.3), and by Corollary 3.3.1,

$$\begin{aligned} f^{N,\beta}_{\boldsymbol{m}|\boldsymbol{\xi}}(\boldsymbol{m'}, \boldsymbol{\xi'}) &= \mathbb{P}_{N,\beta}(\boldsymbol{m}(\sigma) = \boldsymbol{m'} | \boldsymbol{\xi} = \boldsymbol{\xi'}) \\ &= \mathbb{P}_{N,\beta}\left(\tilde{\boldsymbol{m}}(\tilde{\sigma}) = \tilde{\boldsymbol{m}}' | \boldsymbol{X} = \boldsymbol{X'}\right) \\ &= |\{\tilde{\sigma} \in \Omega \ : \ \tilde{\boldsymbol{m}}(\tilde{\sigma}) = \tilde{\boldsymbol{m}}'\}| \frac{1}{Z_{N,\beta,\boldsymbol{X'}}} e^{-\frac{N}{8}\tilde{\boldsymbol{m}}'^{\intercal}\mathcal{M}\tilde{\boldsymbol{m}}'}. \end{aligned} \tag{29}$$

To find an expression for the factor in front, we need to find the amount of configurations $\tilde{\sigma}$ that give rise to the same state $\tilde{\boldsymbol{m}}'$. The fact that we deal with gCW states here (and not Hopfield states) is very convenient, as the values of the gCW order parameters are only influenced by the spins of their own partition. Let $\tilde{\boldsymbol{m}}' = (\tilde{m}'_1, \tilde{m}'_2)$.

$$\begin{aligned} |\{\tilde{\sigma} \in \Omega \ : \ \tilde{\boldsymbol{m}}(\tilde{\sigma}) = \tilde{\boldsymbol{m}}'\}| &= |\{\tilde{\sigma} \in \Omega \ : \ \tilde{m}_1(\tilde{\sigma}^1) = \tilde{m}'_1, \ \tilde{m}_2(\tilde{\sigma}^2) = \tilde{m}'_2\}| \\ &= |\{\tilde{\sigma}^1 \in \Omega_1 \ : \ \tilde{m}_1(\tilde{\sigma}^1) = \tilde{m}'_1\}| \cdot |\{\tilde{\sigma}^2 \in \Omega_2 \ : \ \tilde{m}_2(\tilde{\sigma}^2) = \tilde{m}'_2\}|. \end{aligned} \tag{30}$$

For $i \in \{1, 2\}$, we look for the amount of configurations $\tilde{\sigma}^i$ that give rise to the same state $\tilde{m}'_i$. If in some configuration $\tilde{\sigma}^i$ we have k times a $+1$ spin and $\frac{N}{2} - k$ times a -1 spin, then $\tilde{m}_i(\tilde{\sigma}^i) = \frac{2}{N}(2k - \frac{N}{2})$. Therefore, for $\tilde{m}_i(\tilde{\sigma}^i) = \tilde{m}'_i$ we need $k = \frac{N}{4}(1 + \tilde{m}'_i)$.

The amount of unique ways to choose k spins in the partition $\mathcal{V}_i$ to set to $+1$ is given by a binomial coefficient, and we find

$$|\{\tilde{\sigma}^i \in \Omega_i \ : \ \tilde{m}_i(\tilde{\sigma}^i) = \tilde{m}'_i\}| = \binom{\frac{1}{2}N}{\frac{1+\tilde{m}'_i}{4}N}.$$

As N is usually very large, it is convenient to take a Stirling approximation.

$$\log \binom{\frac{1}{2}N}{\frac{1+\tilde{m}'_i}{2}\frac{1}{2}N} \approx -\frac{1}{2}NS\left(\tilde{m}'_i\right) - \mathcal{O}\left(\log \sqrt{\frac{1}{2}N}\right), \tag{31}$$

where S was the entropy of a partition (19). Note that $S(-1) = S(1) = 0$ (by L'Hôpital's rule on $\lim_{x \to 0^+} x \log x$).

We arrive at our estimate of the probability density function of the network states by combining (29), (30), (31) and (19).

$$f^{N,\beta}_{\boldsymbol{m}|\boldsymbol{\xi}}(\boldsymbol{m'}, \boldsymbol{\xi'}) \approx \frac{1}{Z_{N,\beta,\boldsymbol{X'}}} e^{-N\beta\tilde{F}_{\alpha,\beta,\gamma}(\tilde{\boldsymbol{m}}')}, \tag{32}$$

where $\tilde{F}_{\alpha,\beta,\gamma}$ is the free energy (18)

with $\mathcal{M}$ as in (17). Similarly, using the Stirling approximation for the partition function yields

$$Z_{N,\beta,\boldsymbol{X}'} \approx \sum_{\tilde{\boldsymbol{m}}' \in \mathrm{Im}(\tilde{\boldsymbol{m}})} e^{-N\beta \tilde{F}_{\alpha,\beta,\gamma}(\tilde{\boldsymbol{m}}')}.$$

5.4.2 Two-state system approximation

Next up, we want to estimate the probability of finding the equilibrium network in either state 1 or 2. The probability to find the network in state 1 is

$$p_1 := \mathbb{P}_{N,\beta}(|m_1(\sigma)| \geq |m_2(\sigma)|) = \sum_{\substack{\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m}): \\ |m_1'| \geq |m_2'|}} f^{N,\beta}_{\boldsymbol{m}|\boldsymbol{\xi}}(\boldsymbol{m}', \boldsymbol{\xi}'),$$

and the probability of finding the network in state 2 is

$$p_2 := \mathbb{P}_{N,\beta}(|m_2(\sigma)| > |m_1(\sigma)|) = \sum_{\substack{\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m}): \\ |m_2'| > |m_1'|}} f^{N,\beta}_{\boldsymbol{m}|\boldsymbol{\xi}}(\boldsymbol{m}', \boldsymbol{\xi}') = 1 - p_1.$$

From (32) we see that for very large N, the only states that have significant probability to occur are those whose corresponding 2-gCW state minimizes the free energy. Indeed, any 2-gCW state whose free energy differs from the minimum will have a small Boltzmann factor in comparison with states that minimize the free energy, as the very large N blows up any small differences in free energy. To see this, bound the (approximation of the) partition function

$$Z_{N,\beta,\boldsymbol{X}'} \geq e^{-N\beta \min_{\tilde{\boldsymbol{m}}} \tilde{F}_{\alpha,\beta,\gamma}(\tilde{\boldsymbol{m}})},$$

and then

$$f^{N,\beta}_{\boldsymbol{m}|\boldsymbol{\xi}}(\boldsymbol{m}', \boldsymbol{\xi}') \leq \exp\left[-N\beta\left(\tilde{F}_{\alpha,\beta,\gamma}(\tilde{\boldsymbol{m}}') - \min_{\tilde{\boldsymbol{m}}} \tilde{F}_{\alpha,\beta,\gamma}(\tilde{\boldsymbol{m}})\right)\right].$$

If $\tilde{\boldsymbol{m}}'$ does not minimize F, this goes to zero as $N \to \infty$. For a more rigorous treatment, see Section 5.3.3.

Our next approximation is to consider only large values of β. Let

$$\begin{aligned} \tilde{F}_{\alpha,\gamma}(\tilde{\boldsymbol{m}}') &:= \lim_{\beta\to\infty} F_{\alpha,\beta,\gamma}(\tilde{\boldsymbol{m}}') \\ &= -\frac{1}{8}\left[\left((2+2\alpha+\gamma)\tilde{m}_1'^2 + (2-2\alpha+\gamma)\tilde{m}_2'^2\right) + 2\gamma \tilde{m}_1'\tilde{m}_2'\right]. \end{aligned}$$

Consider the case $\gamma = 0$ for the moment. As $\beta \to \infty$, the free energy becomes

$$\tilde{F}_{\alpha,0}(\tilde{\boldsymbol{m}}') = -\frac{1}{4}\left((1+\alpha)\tilde{m}_1'^2 + (1-\alpha)\tilde{m}_2'^2\right),$$

which is minimized by the states

$$\tilde{\boldsymbol{q}} \in \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right\}.$$

Let $\tilde{\boldsymbol{q}}_1 = (1,1)^\intercal, \tilde{\boldsymbol{q}}_2 = (1,-1)^\intercal$. The corresponding Hopfield states are $\boldsymbol{q}_1 = (1,0)^\intercal, \boldsymbol{q}_2 = (0,1)^\intercal$ respectively. The unique configuration that gives rise to the state $\boldsymbol{q}_1$ is ξ^1, and for $\boldsymbol{q}_2$ it is ξ^2. For large N and at low temperatures, the network has a high probability to be in one of the attractor states.

States other than those that minimize te free energy have negligible probability when N is large, and for large β the states that minimize the free energy are approximately $\tilde{\boldsymbol{q}}_1$ and $\tilde{\boldsymbol{q}}_2$ (we will not consider the other two minima $-\tilde{\boldsymbol{q}}_1$ and $-\tilde{\boldsymbol{q}}_2$). We can therefore approximate our network as a *two-state system*, consisting only of the states $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$.

The probability of finding the network in state 1 or state 2 in the two-state approximation is

$$p_1 \approx \max_{\substack{\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m}): \\ |m_1'| \geq |m_2'|}} f^{N,\beta}_{\boldsymbol{m}|\boldsymbol{\xi}}(\boldsymbol{m}', \boldsymbol{\xi}') \approx f^{N,\beta}_{\boldsymbol{m}|\boldsymbol{\xi}}(\boldsymbol{q}_1, \boldsymbol{\xi}'), \qquad p_2 \approx \max_{\substack{\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m}): \\ |m_2'| > |m_1'|}} f^{N,\beta}_{\boldsymbol{m}|\boldsymbol{\xi}}(\boldsymbol{m}', \boldsymbol{\xi}') \approx f^{N,\beta}_{\boldsymbol{m}|\boldsymbol{\xi}}(\boldsymbol{q}_2, \boldsymbol{\xi}'). \tag{33}$$

This two-state system is not interesting at the moment, as the states $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ are *degenerate*, i.e. their energies are the same, and so they always have equal probability of occurring. We break this degeneracy by decreasing the energy of state 1 a little bit using the parameter γ. From now on, we will keep the previous approximation but allow γ to be positive and small. We have

$$\tilde{F}_{\alpha,\gamma}(\tilde{\boldsymbol{q}}_1) = -\frac{1}{2}(1+\gamma), \qquad \tilde{F}_{\alpha,\gamma}(\tilde{\boldsymbol{q}}_2) = -\frac{1}{2}.$$

(the α parameter cancels out). In this two-state system approximation,

$$Z_{N,\beta,\gamma} \approx e^{-N\beta\tilde{F}_{\alpha,\gamma}(\tilde{\boldsymbol{q}}_1)} + e^{-N\beta\tilde{F}_{\alpha,\gamma}(\tilde{\boldsymbol{q}}_2)} = e^{\frac{1}{2}N\beta(1+\gamma)} + e^{\frac{1}{2}N\beta},$$

and

$$f^{N,\beta}_{\boldsymbol{m}|\boldsymbol{\xi}}(\boldsymbol{q}_1,\boldsymbol{\xi}) \approx \frac{e^{\frac{1}{2}N\beta(1+\gamma)}}{e^{\frac{1}{2}N\beta(1+\gamma)} + e^{\frac{1}{2}N\beta}} = \frac{e^{\frac{1}{2}N\beta\gamma}}{e^{\frac{1}{2}N\beta\gamma}+1}, \qquad f^{N,\beta}_{\boldsymbol{m}|\boldsymbol{\xi}}(\boldsymbol{q}_2,\boldsymbol{\xi}) \approx \frac{1}{e^{\frac{1}{2}N\beta\gamma}+1}. \tag{34}$$

After combining (33) and (34), we conclude

$$p_1 \approx \frac{1}{1+e^{-\frac{1}{2}N\beta\gamma}}, \qquad p_2 \approx \frac{1}{1+e^{\frac{1}{2}N\beta\gamma}}. \tag{35}$$

This is a rough estimate. However, it still allows us to draw some quick conclusions about our 2-pattern state switching network.

1. Larger values of γ increase the probability of finding the network in state 1, and decrease the probability of finding the network in state 2. This is expected, as increasing γ lowers the energy of state 1.

2. At $\gamma = 0$, both state 1 and 2 have probability 1/2 of occurring, and for large γ we will always find the network in state 1. One should take $\gamma \sim 1/(N\beta)$ to get a network that has more interesting state probabilities.

3. The probability to find the network in state 1 (or 2) is given by a Boltzmann distribution (also called Gibbs distribution), with effective state energies that can be approximated by $\epsilon_1 = -\frac{1}{2}N\beta\gamma$ for state 1 and $\epsilon_2 = 0$ for state 2.

The three simulations in figure 24 show statements 1 and 2. We will investigate statement 3 only for the sparse switching network.

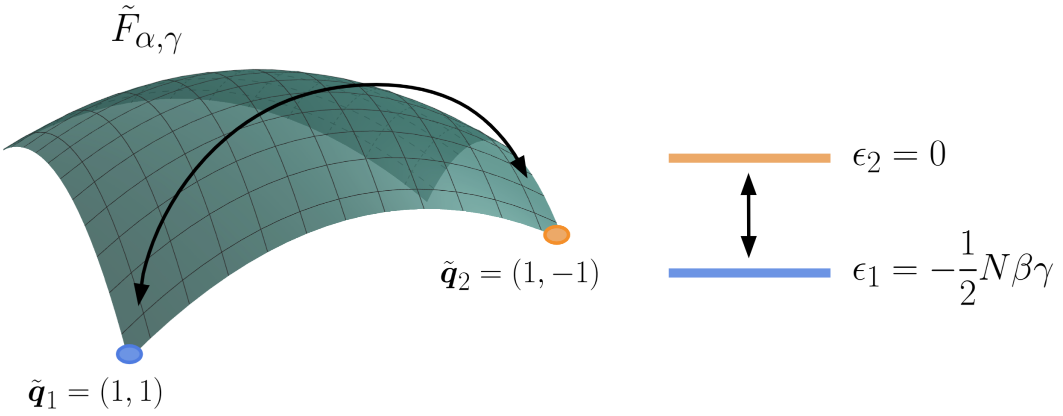
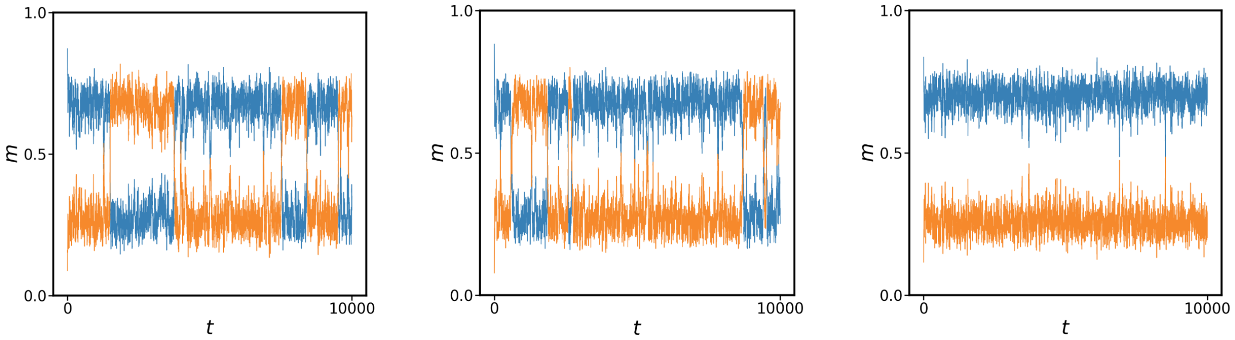

$\tilde{F}_{\alpha,\gamma}$
$\epsilon_2 = 0$
$\tilde{\boldsymbol{q}}_2 = (1,-1)$
$\epsilon_1 = -\frac{1}{2}N\beta\gamma$
$\tilde{\boldsymbol{q}}_1 = (1,1)$

Figure 23: The two-state system approximation. In the low temperature limit, we can approximate the system to only consist of two states: $\tilde{\boldsymbol{q}}_1$ (state 1) and $\tilde{\boldsymbol{q}}_2$ (state 2), which correspond to the configurations ξ^1 and ξ^2 respectively. The probability of finding the system in either state 1 or 2 can be approximated by a Boltzmann distribution with effective energies ϵ_1 and ϵ_2.



$N\beta\gamma = 0$ $N\beta\gamma = 1.5$ $N\beta\gamma = 15$

Figure 24: Time series of the order parameters m_1 (blue) and m_2 (orange), which show the typical behaviour of the network order parameters for different values of γ. In all three simulations, $N = 1000, \alpha = 0.29, \beta = 1.5$, and the simulations are started in the state $m_1 = 1, m_2 = 0$. As we increase γ, at any given timestep we are more likely to see the network in state 1 than in state 2.

There is an important remark here. The fact that p_1 and p_2 are 'nontrivial' (not equal to 1 or 0) does not necessarily mean that the network really switches states on reasonable timescales (although we have seen evidence in simulations). It may also be the case that at the start of the dynamics, the network has for example 50/50 chance to converge to state 1 or state 2, but once it's in either state 1 or 2, it will stay there indefinitely. We need to apply theory of *non-equilibrium* statistical mechanics to properly show with theory that the network really switches states on reasonable timescales. The network then has a *metastability phenomenon*: the network will stay in state 1 or 2 for some time, but it will for sure leave the state again within a reasonable time interval and switch to the other state. We will explore this in Chapter 6.

5.5 Two 2-pattern switching networks have spurious minima

In the embedding weights matrix (16) that we use to embed the two-state Markov chain, we have two copies of a 2-pattern switching network, namely from the weights w_{xy}^{switch1} and w_{xy}^{switch2}. The analysis of a single 2-pattern switching network shows that it is a promising way to implement stochastic transitions between attractor states. First, using theory, we will quickly investigate a network with four patterns $\boldsymbol{\eta} = (\eta^{11}, \eta^{12}, \eta^{21}, \eta^{22})$, and weights

$$w_{xy} = w_{xy}^{\text{switch}}[\eta^{11}, \eta^{12}] + w_{xy}^{\text{switch}}[\eta^{21}, \eta^{22}], \tag{36}$$

and we will see that such networks lead to trouble: they contain minima in their energy landscape whose existence can be detrimental for the proper embedding of a two-state Markov chain. After that, we show simulations of the full embedding weights matrix implemented into a Hopfield network, and we see how these energy minima lead to wrong behaviour.

During a single timestep of the embedded Markov chain, the network switches between two patterns (η^{11} and η^{12}, or η^{21} and η^{22}, see figures 16 and 17 for a refresher). It is essential that there is zero probability that the network switches to a pattern that should not take part in the switching. For example, if the embedded Markov chain is in `state 1`, during the next timestep the network switches between patterns η^{11} and η^{12}. If pattern η^{21} also joins in the switching, the probability to go to pattern ϕ^1 at the end of the timestep increases, as both η^{11} and η^{21} 'point' to ϕ^1. Similarly, if η^{22} joins in the switching, the probability to end up in pattern ϕ^2 increases. In any case, the presence of these unwanted patterns in the switching alters the embedded Markov chain transition matrix P in a complicated way, and we lose controllability over the embedded chain transition probabilities.

5.5.1 Theoretical argument

A quick test to see if unwanted patterns join in the switching is to check what the energy landscape in the mixed phase of the network looks like. If there are only minima at states which only have overlap with the two wanted patterns, the network will very likely not switch to an unwanted pattern. However, if the minima are at states which also have overlap with unwanted patterns, there is the possibility that unwanted patterns join in the switching.

To simplify the analysis of weights matrix (36), we will study this network with $\alpha = 1$ and $\gamma = 0$ for both $w_{xy}^{\text{switch}}[\eta^{11}, \eta^{12}]$ and $w_{xy}^{\text{switch}}[\eta^{21}, \eta^{22}]$. We assume that this situation already gives a good indication of trouble that might arise during switching: if there is overlap with unwanted patterns in a mixed phase with $\alpha = 1$, these unwanted patterns will probably also be present at lower but nonzero values of α.

We analyze the energy landscape through the Hopfield/gCW correspondence. This greatly simplifies the analysis: the space of allowed states in the Hopfield network is complicated, and so it can be unclear where minima of the energy are located, even if the Hamiltonian is known. The space of allowed states of the multi-group Curie-Weiss network is much easier to work with, and the analysis becomes more straightforward. See Corollary 3.2.1 for the space of allowed states of both models. We will assume equal partition sizes; this is a good approximation for large N.

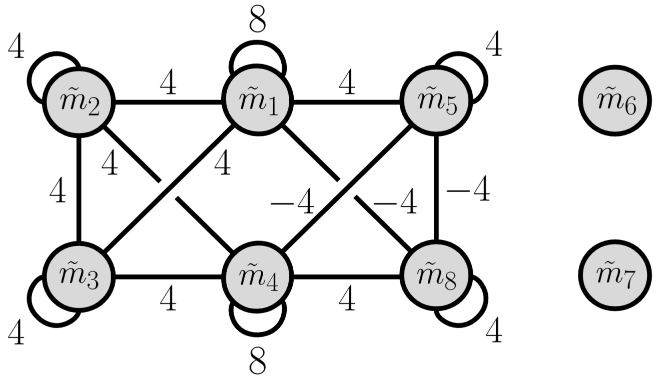
The interaction matrix of the 4-pattern Hopfield network we analyze here (with $\alpha = 1, \gamma = 0$ for both weights) is

$$(Q)_{ij} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

The interaction matrix of the corresponding 8-gCW network is

$$(\mathcal{M})_{ij} = \begin{pmatrix} 8 & 4 & 4 & 0 & 4 & 0 & 0 & -4 \\ 4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 \\ 4 & 4 & 4 & 4 & 0 & 0 & 0 & 0 \\ 0 & 4 & 4 & 8 & -4 & 0 & 0 & 4 \\ 0 & 4 & 4 & 8 & -4 & 0 & 0 & 4 \\ 4 & 0 & 0 & -4 & 4 & 0 & 0 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4 & 0 & 0 & 4 & -4 & 0 & 0 & 4 \end{pmatrix}.$$

We can most easily check for minima of the Hamiltonian through the following graphical representation of the interactions between order parameters, where the coupling at an edge between $\tilde{m}_i$ and $\tilde{m}_j$ is given by $\mathcal{M}_{ij}$:



When we write out the linear form $\tilde{\boldsymbol{m}}^\intercal \mathcal{M} \tilde{\boldsymbol{m}}$ (leaving out the relative partition sizes matrix $\mathcal{X}$, which only contributes multiplication by a constant), we get

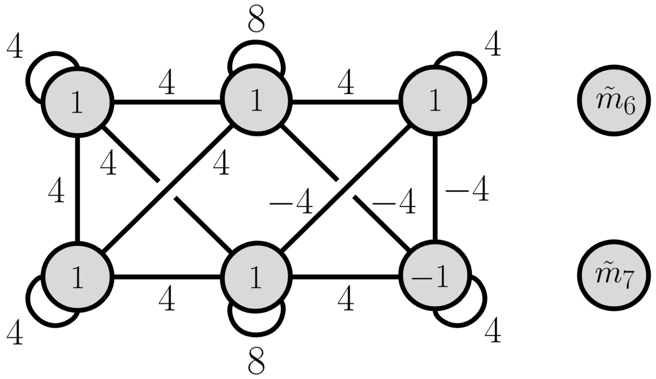
$$\tilde{\boldsymbol{m}}^\intercal \mathcal{M} \tilde{\boldsymbol{m}} = \sum_{i=1}^{8} \sum_{j=1}^{8} \mathcal{M}_{ij} \tilde{m}_i \tilde{m}_j,$$

and we see that each term of this sum can be derived from the graphical representation: any term is a multiplication of the value of two adjacent vertices, times the number at their shared edge. Note that each edge between two different vertices appears twice in the sum: once for $\tilde{m}_i \tilde{m}_j$ and once for $\tilde{m}_j \tilde{m}_i$. Self-interaction edges appear only once. The interaction between order parameters whose vertices do not share an edge is zero. In this case, $\tilde{m}_6$ and $\tilde{m}_7$ also have zero self-interaction, and so their value is completely irrelevant for the energy of the state.

Now, consider the following 8-gCW state:

$$\tilde{\boldsymbol{u}}' = (1, 1, 1, 1, 1, \tilde{m}_6, \tilde{m}_7, -1)^\intercal,$$

where $\tilde{m}_6$ and $\tilde{m}_7$ can take any value in $[-1, 1]$.



If we want to change the value of any order parameter that's set to 1, we can only decrease it. Neglecting the self-couplings, if we change $\tilde{m}_1$ or $\tilde{m}_4$ from 1 to $1-\epsilon$ (ϵ is a small number), the energy changes by a negative amount, as we lose $2 \cdot 3 \cdot 4\epsilon$ from the three positively coupled edges but only gain $2 \cdot 1 \cdot 4\epsilon$ from the negatively coupled edge (note that each edge appears twice in the sum, hence the factor 2). If we change $\tilde{m}_2$ or $\tilde{m}_3$ from 1 to $1-\epsilon$, we lose $2 \cdot 3 \cdot 4\epsilon$ as the edges between these order parameters only have positively coupling. If we change $\tilde{m}_3$ from 1 to $1-\epsilon$, we lose $2 \cdot 2 \cdot 4\epsilon$ from the edge with $\tilde{m}_1$ and $\tilde{m}_8$, but only gain $2 \cdot 1 \cdot 4\epsilon$ from the coupling with $\tilde{m}_4$.

If we want to change the value of any order parameter that's set to -1, we can only increase it. If we change $\tilde{m}_8$ from -1 to $-1+\epsilon$, we lose $2 \cdot 2 \cdot 4\epsilon$ from the coupling with $\tilde{m}_1$ and $\tilde{m}_5$, but only gain $2 \cdot 1 \cdot 4\epsilon$ from the coupling with $\tilde{m}_4$.

Any small change to any order parameter thus decreases the linear form $\tilde{\boldsymbol{m}}^\intercal \mathcal{M} \tilde{\boldsymbol{m}}$. This still holds when we also consider the self-interactions (can be checked with a very small computation). As the Hamiltonian of the 8-gCW network is $-\frac{N}{2}\tilde{\boldsymbol{m}}^\intercal \mathcal{X}\mathcal{M}\mathcal{X}\tilde{\boldsymbol{m}}$, the energy thus increases with any small variation of the order parameter values. We see that the state $\tilde{\boldsymbol{u}}'$ is a local minimum of the Hamiltonian. Furthermore, $\tilde{H}_{N,\boldsymbol{X}}(\tilde{\boldsymbol{u}}') = -\frac{5}{8}N$.

The Hopfield state that corresponds to this 8-gCW state is

$$\boldsymbol{u}' := \frac{1}{8}A\tilde{\boldsymbol{u}}' = (1/2, 1/2, 1/4, 1/4)^\intercal, \qquad \text{where } (A)_{ij} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{pmatrix}.$$

By Theorem 3.2, we thus find $H_{N,\boldsymbol{\xi}}(\boldsymbol{u}') = -\frac{5}{8}N$.

Now, consider a Hopfield state in the mixed phase, such that there is only maximal overlap between two patterns, and the overlap with the other two patterns is zero. The Hopfield state that minimizes the energy and satisfies this property has the form

$$\boldsymbol{v}' := (1/2, 1/2, 0, 0)^\intercal$$

(we could also pick m_3, m_4 to be the nonzero order parameters and $m_1, m_2 = 0$, or consider $-\boldsymbol{v}'$ for example, but these states all have the same energy). The energy of $\boldsymbol{v}'$ is quickly computed to be $H_{N,\boldsymbol{\xi}}(\boldsymbol{v}') = -\frac{4}{8}N$, and so it is $\frac{1}{8}N$ higher than the energy of the state $\boldsymbol{u}'$.

This Hopfield network with four patterns and two sets of switch weights thus has a rather surprising state that minimizes the energy locally, namely the state $(1/2, 1/2, 1/4, 1/4)^\intercal$, in which patterns 1 and 2 mix fifty/fifty, but the 'undesired' patterns 3 and 4 also join in the mixing (although less prominently). The energy of this state is lower than that of the state $(1/2, 1/2, 0, 0)^\intercal$, which is the state of the network when patterns 1 and 2 mix fifty/fifty, and the unwanted patterns are not present. Therefore, as our desired state (a 'clean' mixing of patterns 1 and 2) is for sure not a global minimum of the Hamiltonian, it can only be metastable at best. If the network ever achieves this desired state, it will likely shortly after transition to the nearby minimum of the energy, which is the undesired state $(1/2, 1/2, 1/4, 1/4)^\intercal$.

5.5.2 Simulations

The embedding weights matrix (16) that we would like to use to embed the two-state Markov chain has been implemented in a Hopfield network and has been simulated. During simulation, three different characteristic behaviours appeared. These different behaviours of the network are reason enough to abandon the Hopfield network as the ANN in which we will embed the Markov chain. We therefore only show and explain these different behaviours, and we won't investigate the properties of this network any further.

Recall the events that make up a single timestep of the embedded Markov chain: 1) turn the 'push' parameter $\delta^{\uparrow}$ on and off, 2) let the edge patterns switch for fixed time, 3) turn the 'pull' paramter $\delta^{\downarrow}$ on and off. In figures 25, 26 and 27, we show three complete timesteps of the embedded Markov chain. In each figure, the embedded Markov chain starts in `state 1`: the network has largest overlap with pattern ϕ^1 (blue). The roles that the different patterns play in the embedded Markov chain can be found in figure 16. Each simulation has been performed with $N = 1000$ neurons, $\alpha = 0.32, \beta = 1.5$. While new realizations of the random patterns $\phi^i, i = 1, 2$ and $\eta^{ij}, i, j = 1, 2$ are generated for each simulation, the overlap between any given two patterns used in the simulations is no larger than 0.02 (the patterns could be made perfectly orthogonal by changing only at most 2% of the neuron values).

Figure 25 shows a simulation in which the network had the desired behaviour. During the switching time of the embedded chain timestep, only the correct two patterns switch. We see that after the network is in pattern ϕ^1, the network transitions to pattern η^{11} (light blue), which starts to switch with pattern η^{12} (purple). As the other edge patterns do not interfere, the probability to find that the network has largest overlap with η^{11} is P_{11}, and that it has largest overlap with η^{12} is P_{12}. The other edge patterns (η^{21} and η^{22}) also fluctuate in overlap, but the network never has strong overlap with any of these patterns. At any given timestep of the network itself (not embedded chain timestep!) it is clear which pattern the network has the largest overlap with: the correct edge patterns oscillate in overlap with always one pattern clearly on top.

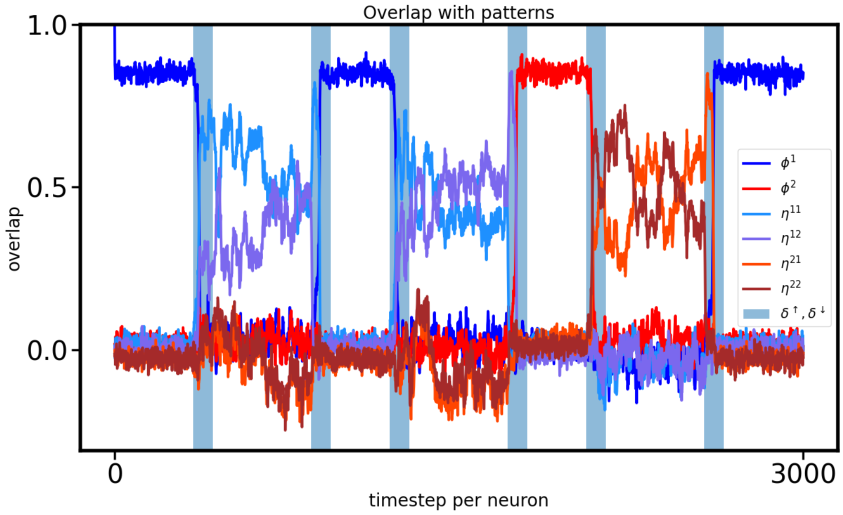


Figure 25: A simulation of the embedded two-state Markov chain, which shows the desired behaviour of the network.

Figure 26 shows a simulation in which during switching, the network attains the state $(1/2, 1/2, 1/4, 1/4)$. In the first timestep of the embedded Markov chain, the network has half overlap with the 'right' edge patterns η^{11}, η^{12}, but also a quarter overlap with the 'wrong' edge patterns η^{21}, η^{22}. In contrast to the situation in figure 25, the edge patterns do not oscillate in overlap with one pattern clearly on top, but rather fully enter the mixed phase in which the oscillations in overlap are very small and hardly distinguishable.

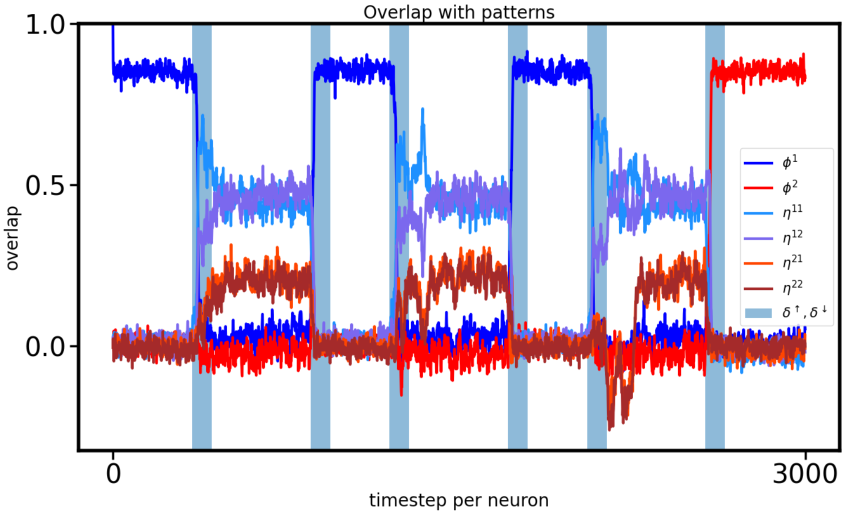


Figure 26: A simulation of the embedded two-state Markov chain, which shows the presence of the state $(1/2, 1/2, 1/4, 1/4)$ during switching.

Figure 27 shows a simulation in which there is catastrophic failure of the embedded two-state Markov chain. The embedded chain starts in `state 1`: the network has largest overlap with ϕ^1. The first timestep of the embedded Markov chain starts when the parameter $\delta^\uparrow$ is turned on and off (first blue region). At first, the edge patterns η^{11} and η^{12} start to switch cleanly, but a little later the network enters the state $(1/2, 1/2, -1/4, -1/4)$, in which the wrong edge patterns η^{21}, η^{22} are clearly present. Close to the end of the first embedded chain timestep (near the beginning of second blue region), the overlap of the network with the wrong edge patterns is larger (in absolute value) than the overlap with the correct ones. When the parameter $\delta^\downarrow$ is turned on and off, the next state of the embedded chain is decided. However, as the network has largest overlap with the wrong edge patterns η^{21}, η^{22}, the embedded chain will enter `state 1` if η^{21} is on top, or `state 2` is η^{22} is on top. Clearly, the probabilities of the next state of the embedded chain are not given by P_{11} and P_{22} anymore. Note that as η^{21} and η^{22} had negative overlap, the network transitions to the pattern $-\phi^2$, which is also an attractor state of the network.

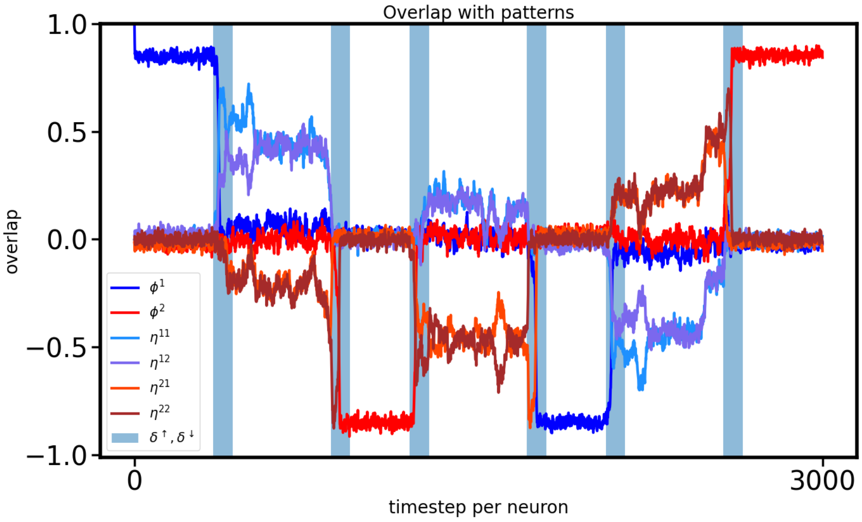


Figure 27: A simulation of the embedded two-state Markov chain, which shows catastrophic failure. This failure is a consequence of the presence of the third and fourth edge pattern during the switching between the first and second edge pattern.

6 Metastability of 2-pattern switching network

In this Chapter, we show that the 2-pattern switching network really is able to switch states on reasonable timescales. As the state switching is a proper dynamical phenomenon, we can no longer work in the equilibrium setting, and we need to introduce *non-equilibrium* statistical mechanics theory. Furthermore, we expect that there are some states in which the 2-pattern switching network stays for an exponentially long time before it switches; these would be *metastable* states. We therefore will investigate if the network has metastability phenomena.

The goal is to identify three main quantities: what are the (meta)stable states, the mean transition time from metastable to stable state, and the states visited with high probability during the transition, which are called *gates*. To find these quantities is usually an exercise in probability theory. However, using well-established theory, we can turn these problems into an exercise in analysis.

The non-equilibrium statistical mechanics theory applied here is the *pathwise approach* to metastability [55] [56] [57]. In this approach, the analysis of metastable phenomena comes down to analyzing the energy differences between different paths that the system can take through the state space. The central object to study is the *maximum stability level*: it is the largest energy barrier the system has to overcome if it wants to go from a metastable to a stable state. After computing this quantity, we obtain the metastable states, transition times and gates almost immediately, using the results of Cirillo et al. [57]. A catch is that their results only work in the large β limit. The Glauber dynamics (Section 2.2.1) satisfies the *Freidlin Wentzell assumptions*, which is needed to apply the theory of Cirillo et al. More specifically, for any $\sigma, \eta \in \Omega$ the transition probabilities satisfy

$$\lim_{\beta \to \infty} -\frac{1}{\beta} \log(\pi_{N,\beta}(\sigma, \eta)) = \Delta_{N,\beta}(\sigma, \eta),$$

where Δ is the rate function: $\Delta_{N,\beta} = \infty$ if σ, η differ in more than one spin, and otherwise

$$\Delta_{N,\beta}(\sigma, \eta) = \begin{cases} o(1) & \text{if } H(\eta) \leq H(\sigma), \\ H(\eta) - H(\sigma) + o(1) & \text{if } H(\eta) > H(\sigma), \end{cases}$$

as $N \to \infty$.

We equip the 2-pattern switching network with the dynamics as defined in Section 2.2.1. We said that the 2-pattern state-switching network switches states if the network transitions from state 1 ($|m_1| \geq |m_2|$) to state 2 ($|m_2| > |m_1|$, see Chapter 5), or from state 2 to state 1. However, in the large β limit, we saw that the network essentially only stays in either $\boldsymbol{m} = (1, 0)$ or $\boldsymbol{m} = (0, 1)$ (or their negatives), and so in this Chapter we say that the network has transitioned from state 2 to state 1 if the network transitions from $\boldsymbol{m} = (0, 1)$ to $\boldsymbol{m} = (1, 0)$. Transitions from $\boldsymbol{m} = (1, 0)$ to $\boldsymbol{m} = (0, 1)$ are not covered by the theory, as we will see that such transitions would go from a stable to a metastable state (although we know from simulations that they do occur).

At the end of this Chapter, we present an approximation for the mean transition time, and some qualitative insights into state switching that follow from it.

6.1 Magnetization chains

The transition probabilities $\pi_{N,\beta}(\sigma, \eta)$ that describe the dynamics of our networks only depend on the energies of the configurations σ, η. That is, σ and η only enter the expression of $\pi_{N,\beta}$ through the Hamiltonian H. For both networks that we have introduced in Chapter 2, the Hamiltonian only depends on the state $\boldsymbol{m}$ of the network, and it thus appears that it is not necessary to keep track of the exact configuration of the network to compute the transition probabilities.

One might hope that instead of working with the Markov chain $\sigma(t) \in \Omega$, we only need to consider the resulting *magnetization chain* $\boldsymbol{m}(t) := \boldsymbol{m}(\sigma(t)) \in \mathrm{Im}(\boldsymbol{m})$. Unfortunately for us, as is pointed out in [58], the dynamics of $\boldsymbol{m}(t)$ is not always Markovian, in the sense that sometimes the knowledge of $\boldsymbol{m}(t)$ alone (and not knowing the configuration the network is in) is not enough to determine the possible transition probabilities from $\boldsymbol{m}(t)$.

In contrast to Hopfield networks, the magnetization chain of a q-gCW network *is* a Markov chain. For convenience we first introduce the following

Definition 6.1 (Energy of a state). *Let $\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m})$ be a state, and let $\sigma \in \Omega$ be any configuration such that $\boldsymbol{m}(\sigma) = \boldsymbol{m}'$. The* ***energy of $\boldsymbol{m}'$*** *is*

$$H_{N,\boldsymbol{X}}(\boldsymbol{m}') := H_{N,\boldsymbol{X}}(\sigma).$$

In this Chapter, all Hamiltonians and order parameters are those of a multi-group Curie-Weiss network, unless explicitly mentioned otherwise. All tilde symbols have been removed to lighten notation.

Proposition 6.1 (q-gCW magnetization chain)**.** *Consider a q-gCW network with N sites, relative partition sizes vector $\boldsymbol{X}$ and Hamiltonian $H_{N,\boldsymbol{X}}$, equipped with the asynchronous dynamics as defined in 2.2.1, which is the Markov chain $\sigma(t) \in \Omega$, $t \in \mathbb{N}$ with transition matrix $\pi_{N,\beta,\boldsymbol{X}}$.*
Denote the set of states that can be reached from state $\boldsymbol{m}'' \in \mathrm{Im}(\boldsymbol{m})$ by flipping only one spin:

$$D^+(\boldsymbol{m}'') := \{\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m}) : \exists i \in \{1, ..., q\} \text{ s.t. } m_i' = m_i'' + 2/n_i \text{ and } m_i' = m_i'' \ \forall j \neq i\},$$
$$D^-(\boldsymbol{m}'') := \{\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m}) : \exists i \in \{1, ..., q\} \text{ s.t. } m_i' = m_i'' - 2/n_i \text{ and } m_i' = m_i'' \ \forall j \neq i\}.$$

For any two states $\boldsymbol{m}'', \boldsymbol{m}'$, if $\boldsymbol{m}'' \neq \boldsymbol{m}'$, define

$$p_{N,\beta,\boldsymbol{X}}(\boldsymbol{m}'', \boldsymbol{m}') := \begin{cases} \frac{1}{q}\left(\frac{1 \mp m_i''}{2}\right) \frac{e^{-\beta H_{N,\boldsymbol{X}}(\boldsymbol{m}')}}{e^{-\beta H_{N,\boldsymbol{X}}(\boldsymbol{m}')} + e^{-\beta H_{N,\boldsymbol{X}}(\boldsymbol{m}'')}} & \text{if } \boldsymbol{m}' \in D^{\pm}(\boldsymbol{m}''), \\ 0 & \text{otherwise,} \end{cases}$$

and if $\boldsymbol{m}'' = \boldsymbol{m}'$ define

$$p_{N,\beta,\boldsymbol{X}}(\boldsymbol{m}'', \boldsymbol{m}'') := 1 - \sum_{\boldsymbol{u} \in \mathrm{Im}(\boldsymbol{m}) \setminus \{\boldsymbol{m}''\}} p_{N,\beta,\boldsymbol{X}}(\boldsymbol{m}'', \boldsymbol{u}).$$

*The **magnetization chain** of this network $\boldsymbol{m}(t) := \boldsymbol{m}(\sigma(t)) \in \mathrm{Im}(\boldsymbol{m})$ is a Markov chain with transition matrix $p_{N,\beta,\boldsymbol{X}}$.*

Proof of Proposition 6.1. As only one spin gets updated at each timestep, and each spin belongs to a unique partition, only one of the order parameters $m_1, ..., m_q$ gets updated at each timestep. Let the updated spin be in partition $\mathcal{V}_i$, where $i \in \{1, ..., q\}$. Then, $m_j(t+1) = m_j(t)$ for all $j \in \{1, ..., q\} \setminus \{i\}$, and $m_i(t+1) = m_i(t) + W_i(t)$, where $W_i(t) \in \{-\frac{2}{n_i}, 0, \frac{2}{n_i}\}$. $W_i(t) = -2/n_i$ if the updated spin flips from negative to positive, $W_i(t) = 2/n_i$ if it flips from positive to negative and $W_i(t) = 0$ otherwise. By definition 2.8, the probability of the updated spin flipping or not is only dependent on $\boldsymbol{m}(t)$, and it is time homogeneous. $W_i(t)$ is thus independent of $\boldsymbol{m}(t-1), ..., \boldsymbol{m}(0)$, and so $\boldsymbol{m}(t+1)$ is only dependent on $\boldsymbol{m}(t)$. The magnetization chain is a Markov chain.

Next, we show that the magnetization chain has the transition matrix $p_{N,\beta,\boldsymbol{X}}$. Let $S' = \{\sigma \in \Omega : \boldsymbol{m}(\sigma) = \boldsymbol{m}'\}$, and $S'' = \{\sigma \in \Omega : \boldsymbol{m}(\sigma) = \boldsymbol{m}''\}$.

$$\begin{aligned} \mathbb{P}_{N,\beta}(\boldsymbol{m}(t+1) = \boldsymbol{m}' \mid \boldsymbol{m}(t) = \boldsymbol{m}'') &= \mathbb{P}_{N,\beta}(\sigma(t+1) \in S' \mid \sigma(t) \in S'') \\ &= \frac{\mathbb{P}_{N,\beta}(\sigma(t+1) \in S', \sigma(t) \in S'')}{\mathbb{P}_{N,\beta}(\sigma(t) \in S'')} \\ &= \underbrace{\frac{\sum_{\eta' \in S'} \sum_{\eta'' \in S''} \pi_{N,\beta,\boldsymbol{X}}(\eta'', \eta') \mathbb{P}_{N,\beta}(\sigma(t) = \eta'')}{\mathbb{P}_{N,\beta}(\sigma(t) \in S'')}}_{(*)}. \end{aligned}$$

For $\eta'' \in S', \eta'' \in S''$, $\pi_{N,\beta,\boldsymbol{X}}(\eta'', \eta') = 0$ if η', η'' differ in more than one spin, and if η', η'' differ in exactly one spin,

$$\pi_{N,\beta,\boldsymbol{X}}(\eta'', \eta') = \frac{1}{N} \frac{e^{-\beta H_{N,\boldsymbol{X}}(\boldsymbol{m}')}}{e^{-\beta H_{N,\boldsymbol{X}}(\boldsymbol{m}')} + e^{-\beta H_{N,\boldsymbol{X}}(\boldsymbol{m}'')}}.$$

So we can write

$$(*) = \frac{1}{N} \frac{e^{-\beta H_{N,\boldsymbol{X}}(\boldsymbol{m}')}}{e^{-\beta H_{N,\boldsymbol{X}}(\boldsymbol{m}')} + e^{-\beta H_{N,\boldsymbol{X}}(\boldsymbol{m}'')}} \sum_{\eta'' \in S''} \left(\frac{\mathbb{P}_{N,\beta}(\sigma(t) = \eta'')}{\mathbb{P}_{N,\beta}(\sigma(t) \in S'')} \sum_{\eta' \in S'} \mathbb{1}_{|\{x \in V_N : \eta_x' \neq \eta_x''\}| = 1} \right).$$

Any configuration $\sigma \in \Omega$ such that $m_i(\sigma) = u_i$ has $\frac{N}{q}(\frac{1+u_i}{2})$ positive spins, and $\frac{N}{q}(\frac{1-u_i}{2})$ negative spins in partition $\mathcal{V}_i$. So, given the state $\boldsymbol{m}''$, if we want to *decrease* m_i'' by $2/n_i$ for some $i \in \{1, ..., q\}$ and leave all other m_j'' with $j \in \{1, ..., q\} \setminus \{i\}$ unchanged, we can choose a positive spin to flip in $\frac{N}{q}(\frac{1+m_i''}{2})$ different ways. If instead we *increase* one order parameter m_i'' by $2/n_i$ and leave the rest unchanged, we have $\frac{N}{q}(\frac{1-m_i''}{2})$ ways to do so.

So, given $\eta'' \in S''$, if $m_i' = m_i'' \pm 2/n_i$,

$$\sum_{\eta' \in S'} \mathbb{1}_{|\{x \in V_N : \eta_x' \neq \eta_x''\}| = 1} = \frac{N}{q} \left(\frac{1 \mp m_i''}{2} \right),$$

and we conclude that

$$(*) = p_{N,\beta,\boldsymbol{X}}(\boldsymbol{m}'', \boldsymbol{m}').$$

Lastly, if $\boldsymbol{m}'' = \boldsymbol{m}'$, we have $S'' = S'$. For all $\boldsymbol{u} \in \mathrm{Im}(\boldsymbol{m})$, define partitions of the configuration space: $W(\boldsymbol{u}) = \{\eta \in \Omega : \boldsymbol{m}(\eta) = \boldsymbol{u}\}$. Then,

$$
\begin{aligned}
(*) &= \frac{\sum_{\eta',\eta''\in S''} \pi_{N,\beta,\boldsymbol{X}}(\eta'',\eta')\mathbb{P}_{N,\beta}(\sigma(t)=\eta'')}{\mathbb{P}_{N,\beta}(\sigma(t)\in S'')} \\
&= \frac{\sum_{\eta''\in S''} \pi_{N,\beta,\boldsymbol{X}}(\eta'',\eta'')\mathbb{P}_{N,\beta}(\sigma(t)=\eta'')}{\mathbb{P}_{N,\beta}(\sigma(t)\in S'')} \\
&= \sum_{\eta''\in S''}\left(1-\sum_{\eta'\in\Omega\setminus\{\eta''\}} \pi_{N,\beta,\boldsymbol{X}}(\eta'',\eta')\right)\frac{\mathbb{P}_{N,\beta}(\sigma(t)=\eta'')}{\mathbb{P}_{N,\beta}(\sigma(t)\in S'')} \\
&= 1-\sum_{\boldsymbol{u}\in\mathrm{Im}(\boldsymbol{m})\setminus\{\boldsymbol{m}''\}}\left(\sum_{\eta''\in S''}\sum_{\eta'\in W(\boldsymbol{u})} \pi_{N,\beta,\boldsymbol{X}}(\eta'',\eta')\frac{\mathbb{P}_{N,\beta}(\sigma(t)=\eta'')}{\mathbb{P}_{N,\beta}(\sigma(t)\in S'')}\right) \\
&= 1-\sum_{\boldsymbol{u}\in\mathrm{Im}(\boldsymbol{m})\setminus\{\boldsymbol{m}''\}} p_{N,\beta,\boldsymbol{X}}(\boldsymbol{m}'',\boldsymbol{u}).
\end{aligned}
$$

□

In the pathwise approach to metastability, all relevant quantities are constructed from the analysis of paths through the configuration space Ω. A path through configuration space is a sequence of configurations $(\sigma(1),...,\sigma(n))\in\Omega^n$ such that there is nonzero probability to transition from $\sigma(t)$ to $\sigma(t+1)$ for all $t=1,...,n-1$.

For Hopfield states, knowledge of only the current state is sometimes not enough to determine what other states can be transitioned to with nonzero probability. If we therefore try to define a notion of paths through the Hopfield state space $\mathrm{Im}(\boldsymbol{m})$ in a similar way as done for configurations, we get stuck. We can only construct paths through the state space by keeping track of the underlying configuration; this is equally labourous as working with paths through the configuration space itself.

For multi-group Curie-Weiss networks, the situation is better. Given any gCW state, we know all possible transition directions, and we do not need to know the exact underlying configuration. We can easily construct all paths through the gCW state space $\mathrm{Im}(\boldsymbol{m})$, of which there are a lot less than the amount of paths through configuration space. We will show this concretely for 2-gCW networks.

6.2 State lattice

Consider a general 2-gCW network, with N sites and relative partition sizes vector $\boldsymbol{X}=(X,1-X)$ (where $X\in S_N$, see Section 2.1.2). From Corollary 3.2.1, we find that the space of allowed states of this network is

$$
\mathrm{Im}(\boldsymbol{m})=\left\{-1+\frac{2k}{n_1^{(i)}}:k=0,...,n_1^{(i)}\right\}\otimes\left\{-1+\frac{2l}{n_2^{(i)}}:l=0,...,n_2^{(i)}\right\}\subset[-1,1]^2,
$$

where $n_1^{(i)}=XN$ and $n_2^{(i)}=(1-X)N$.

Definition 6.2 (State lattice)**.** *The* ***state lattice*** $\mathcal{L}_{N,\boldsymbol{X}}:=(V,U)$ *is a graph with vertex set* $V:=\mathrm{Im}(\boldsymbol{m})\subset\mathbb{R}^2$*, and edge set*

$$
U:=\{\{\boldsymbol{m}',\boldsymbol{s}'\}\in\mathrm{Im}(\boldsymbol{m})^2:(|m_1'-s_1'|,|m_2'-s_2'|)\in\{(2/n_1^{(i)},0),(0,2/n_2^{(i)})\}\}.
$$

Two states $\boldsymbol{m}^1,\boldsymbol{m}^2\in\mathcal{L}_{N,\boldsymbol{X}}$ *are called* ***neighbors*** *iff* $\{\boldsymbol{m}^1,\boldsymbol{m}^2\}\in U$.
The ***boundary*** *of the state lattice is is the set of states*

$$
\partial\mathcal{L}_{N,\boldsymbol{X}}:=\{v\in V:\deg(v)<4\}
$$

and the ***corners*** *of the state lattice is the set of states*

$$
\mathcal{C}:=\{v\in V:\deg(v)=2\}
$$

(see figure 28).

When the context is clear, we will sometimes abuse notation and write the set of allowed states as $\mathcal{L}_{N,\boldsymbol{X}}$ instead of $\mathrm{Im}(\boldsymbol{m})$; this is in order to lighten notation in the proofs.

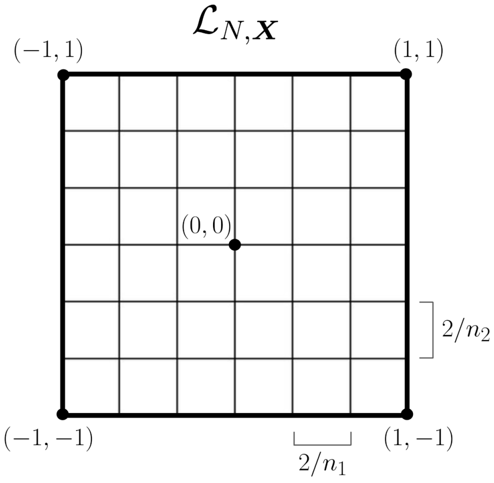


Figure 28: Example of a state lattice, where all neighbors are connected by an edge.

Definition 6.3 (Path). *A path of length $n > 1$ is a sequence of states $(\boldsymbol{r}(1), ..., \boldsymbol{r}(n)) \in \mathrm{Im}(\boldsymbol{m})^n$ such that $p_{N,\beta,\boldsymbol{X}}(\boldsymbol{r}(t), \boldsymbol{r}(t+1)) > 0$ for any $t \in \{1, ..., n-1\}$.*

Corollary 6.3.1. *For any path $\boldsymbol{r}$ of length $n > 1$, and any $t = 0, ..., n-1$, either $\boldsymbol{r}(t), \boldsymbol{r}(t+1)$ are neighbors on the state lattice, or $\boldsymbol{r}(t+1) = \boldsymbol{r}(t)$.*

The magnetization chain of a 2-gCW network is thus a (lazy) random walk on the state lattice. This is very convenient: in the pathwise approach to metastability, we work with paths through the space of allowed states. By interpreting the paths as random walks, we can use geometric arguments in our analysis.

6.3 Pathwise approach to metastability

Here we introduce the definitions and results from [57], [56] that play a part in the pathwise approach to metastability. While these definitions were introduced for the Markov chain of configurations $\sigma(t)$, we will immediately adapt them to be used for our magnetization chain $\boldsymbol{m}(t)$ (with the dynamics of Proposition 6.1). For the remainder of this Section, all definitions and results we introduce are for a general q-gCW network with N sites and relative partition sizes vector $\boldsymbol{X}$, equipped with the asynchronous dynamics.

Let $\mathcal{F}$ be the set of all loop-free paths, and for nonempty sets of configurations $E, E' \subset \mathrm{Im}(\boldsymbol{m})$, let $\mathcal{F}(E, E')$ be the set of all loop-free paths with the first state in E and the last state in E'.

Definition 6.4 (Communication height). *For any nonempty $E, E' \subset \mathrm{Im}(\boldsymbol{m})$, the **communication height** between E and E' is*

$$\Phi_{N,\boldsymbol{X}}[E, E'] := \min_{\boldsymbol{r} \in \mathcal{F}(E,E')} \max_{t=1,...,n} H_{N,\boldsymbol{X}}(\boldsymbol{r}(t)),$$

i.e. it is the highest energy that any path between E and E' must necessarily attain.

Remark 6.4.1. *In the definition of the communication height in [57], one considers paths through configuration space instead of paths through the gCW state space. For any path $\omega = \omega(1), ..., \omega(n)$ through configuration space, its magnetization chain $\boldsymbol{r} = \boldsymbol{m}(\omega(1)), ..., \boldsymbol{m}(\omega(n))$ has exactly the same energy at each $t = 1, .., n$. As $\boldsymbol{m} : \Omega \to \mathrm{Im}(\boldsymbol{m})$ is a surjective function, all metastability properties of the Hamiltonian defined on configuration space will be the same as for the Hamiltonian defined on the state space (6.1). We can thus use the magnetization chain to find the metastability properties of the 2-gCW network.*

For any state $\boldsymbol{m}'$ with energy $H_{N,\boldsymbol{X}}(\boldsymbol{m}')$, the *stability level of* $\boldsymbol{m}'$ is the largest energy barrier that has to be overcome by any path that connects $\boldsymbol{m}'$ to any state with energy lower than $H_{N,\boldsymbol{X}}(\boldsymbol{m}')$. Let $E^{\mathrm{s}} \subset \mathrm{Im}(\boldsymbol{m})$ be the set of absolute minima of the Hamiltonian $H_{N,\boldsymbol{X}}$ (also called the *stable states*, or *ground states* of the network).

Definition 6.5 (Stability level)**.** *Let $\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m}) \setminus E^{\mathrm{s}}$, and denote the set of states with energy lower than $H_{N,\boldsymbol{X}}(\boldsymbol{m}')$ as*

$$E^{-}(\boldsymbol{m}') := \{\boldsymbol{s}' \in \mathrm{Im}(\boldsymbol{m}) : H_{N,\boldsymbol{X}}(\boldsymbol{s}') < H_{N,\boldsymbol{X}}(\boldsymbol{m}')\}.$$

The ***stability level*** *of $\boldsymbol{m}'$ is*

$$V_{N,\boldsymbol{X}}(\boldsymbol{m}') := \Phi\left[\{\boldsymbol{m}'\}, E^{-}(\boldsymbol{m}')\right] - H_{N,\boldsymbol{X}}(\boldsymbol{m}').$$

We are now in a position to define metastability.

Definition 6.6 (Metastability)**.** *The* ***maximum stability level*** *of the network is*

$$\Gamma_{N,\boldsymbol{X}} := \max_{\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m}) \setminus E^{\mathrm{s}}} V_{N,\boldsymbol{X}}(\boldsymbol{m}') > 0.$$

The set of ***metastable states*** *of the network is*

$$E^{\mathrm{m}} := \{\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m}) \setminus E^{\mathrm{s}} : V_{N,\boldsymbol{X}}(\boldsymbol{m}) = \Gamma_{N,\boldsymbol{X}}\}.$$

Define the random variable $\tau_{\mathrm{s}} := \inf\{t \geq 0 : \boldsymbol{m}(t) \in E^{\mathrm{s}}\}$ to be the *first hitting time* of the set of stable states. It is proven by Cirillo et al. in [57] that, if the chain is started in a metastable state in E^{m}, τ_{s} is almost surely of the order $\exp[\beta\Gamma]$, and so we say that the lifespan of any metastable state in E^{m} is exponentially long.

Cirillo et al. also give an estimate for the expected value of τ_{s}. Let $\mathbb{E}^{\boldsymbol{m}'}_{N,\boldsymbol{X}}[\tau_s]$ be the expected value of τ_{s}, with the chain $\boldsymbol{m}(t)$ started at $\boldsymbol{m}(0) = \boldsymbol{m}'$. It is proven that, for any metastable state $\boldsymbol{m}' \in E^{\mathrm{m}}$,

$$\lim_{\beta\to\infty} \frac{1}{\beta} \log \mathbb{E}^{\boldsymbol{m}'}_{N,\boldsymbol{X}}[\tau_{\mathrm{s}}] = \Gamma_{N,\boldsymbol{X}}.$$

We then have the estimate

$$\mathbb{E}^{\boldsymbol{m}'}_{N,\boldsymbol{X}}[\tau_{\mathrm{s}}] \approx e^{\beta\Gamma_{N,\boldsymbol{X}}}. \tag{37}$$

We will use this estimate in our analysis of the 2-pattern switching network to show that the network transitions on reasonable timescales.

The last definitions that we will introduce are those that describe the states that must necessarily be visited during a transition from a metastable state to a stable state. We start with *optimal paths*: they are paths between two sets of states, whose highest energy is equal to the communication height between the two sets. They are optimal in the sense that there are no other paths between the two sets that have lower maximum energy.

Definition 6.7 (Optimal paths, saddles)**.** *For any nonempty $E, E' \subset \mathrm{Im}(\boldsymbol{m})$, the set of* ***optimal paths*** *connecting E and E' are*

$$\mathcal{F}_{\mathrm{o}}(E, E') := \left\{\boldsymbol{r} \in \mathcal{F}(E, E') : \max_{t=0,\ldots,n-1} H_{N,\boldsymbol{X}}(\boldsymbol{r}(t)) = \Phi_{N,\boldsymbol{X}}[E, E']\right\}.$$

The set of ***saddles*** *$\mathcal{S}[E, E']$ between E and E' is the set of states where the optimal paths in $\mathcal{F}_{\mathrm{o}}(E, E')$ attain their highest energy.*

Some saddles might be unique to only a few optimal paths, while other saddles might be present in all optimal paths. We call the set of the latter saddles a *gate*.

Definition 6.8 (Gate)**.** *Given $\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m})$ and $E \subset \mathrm{Im}(\boldsymbol{m})$, a subset of saddles $W \subset \mathcal{S}[\{\boldsymbol{m}'\}, E]$ is a* ***gate*** *and $\boldsymbol{m}'$ to E iff every optimal path in $\mathcal{F}_{\mathrm{o}}(\{\boldsymbol{m}'\}, E)$ intersects W.*
A gate is ***minimal*** *iff for any subset of gates $W' \subset W$, there exists an optimal path in $\mathcal{F}_{\mathrm{o}}(\{\boldsymbol{m}'\}, E)$ that does not intersect W'.*

Gates are interesting, because at low temperatures they must be necessarily visited during the transition from a metastable state to a stable state. It is proven in [57] that if we start the chain $\boldsymbol{m}(t)$ in a metastable state $\boldsymbol{m}' \in E^{\mathrm{m}}$, there exists $c > 0$ such that

$$\mathbb{P}_{N,\beta}(\tau_W > \tau_{\mathrm{s}}) \leq e^{-c\beta},$$

with τ_W the first hitting time of the gate between $\boldsymbol{m}'$ and E^{s}.

Gates are defined for networks with a fixed and finite amount of sites N. However, we are interested in the limiting behaviour of the network as N becomes large. The network that we treat consists of two partitions, with sequence of sizes $(n_1^{(i)})_{i\in\mathbb{N}}, (n_2^{(i)})_{i\in\mathbb{N}}$ respectively, and we wish to have a notion of gates in the limit $i \to \infty$.

Definition 6.9 (Limit gate)**.** *Let $\mathcal{B}_\kappa(\boldsymbol{s}) \subset \mathbb{R}^2$ be the open ball centered at $\boldsymbol{s} \in \mathbb{R}^2$ with radius $\kappa > 0$. For a set $K \subset \mathbb{R}^2$, let $\mathcal{B}_\kappa(K) := \bigcup_{\boldsymbol{s}\in K} \mathcal{B}_\kappa(\boldsymbol{s})$ be an open neighborhood.*

Given $\boldsymbol{m}' \in \mathrm{Im}(\boldsymbol{m})$, $E \subset \mathrm{Im}(\boldsymbol{m})$, and an increasing sequence of partition sizes $(n^{(i)})_{i\in\mathbb{N}} = (n_1^{(i)}, n_2^{(i)})_{i\in\mathbb{N}}$, let $W_{n^{(i)}}[\{\boldsymbol{m}\}, E]$ be the minimal gate for $\boldsymbol{m}'$ and E, when the partitions have sizes $n^{(i)}$.

*The **limit gate** for $\boldsymbol{m}'$ and E is the set $K' \in \mathbb{R}^2$, such that for any $\kappa > 0$, there exists $i \in \mathbb{N}$ large enough such that $W_{n^{(j)}}[\{\boldsymbol{m}\}, E] \subset \mathcal{B}_\kappa(K')$ for all $j \geq i$.*

*The **minimal limit gate** for $\boldsymbol{m}'$ and E is the intersection of all limit gates for $\boldsymbol{m}'$ and E.*

Remark 6.9.1. *It is not possible to use the standard definition of limits of sets, as often an element in the limit gate is not an element in any of the minimal gates of the network, for any finite n_1, n_2.*

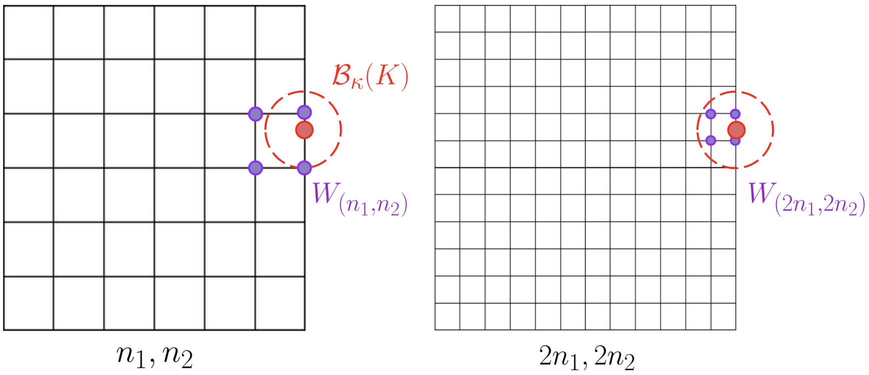


Figure 29: Schematic overview of Definition 6.9. An example of a set K (red dot) and its open neighborhood (red circle) is shown. Left: For some n_1, n_2 the minimal gate might not be contained in the open neighborhood. Right: If there is large enough n_1', n_2' such that the minimal gate is contained in the open neighborhood of the set K for arbitrary $\kappa > 0$, then K is the limit gate (in this example, $n_1' = 2n_1, n_2' = 2n_2$).

6.4 Metastability of the 2-group Curie-Weiss switching network

Recall from Section 5.2 that the interaction matrix of the 2-gCW network corresponding to the 2-pattern switching network is

$$(\mathcal{M})_{ij} = \begin{pmatrix} 2+2\alpha+\gamma & \gamma \\ \gamma & 2-2\alpha+\gamma \end{pmatrix}, \tag{38}$$

and $\mathcal{X} = \mathrm{diag}(X, 1-X)$, with $X \in S_N$ (see Section 2.1.2). The energy of a state $\boldsymbol{m} = (m_1, m_2) \in \mathrm{Im}(\boldsymbol{m})$ of this network is then given by

$$\begin{aligned} H_{N,\boldsymbol{X}}(\boldsymbol{m}') &= -\frac{N}{2}\boldsymbol{m}'^{\intercal}\mathcal{X}\mathcal{M}\mathcal{X}\boldsymbol{m}' \\ &= -\frac{N}{2}\left[(2+2\alpha+\gamma)X^2m_1^2 + (2-2\alpha+\gamma)(1-X)^2 m_2^2 + 2\gamma X(1-X)\,m_1m_2\right]. \end{aligned} \tag{39}$$

Lemma 6.1. *If $X \in S_N \setminus \{0,1\}$ and $\alpha \in [0,1)$, the Hamiltonian $H_{N,\boldsymbol{X}} : [-1,1]^2 \to \mathbb{R}$ (from (39)) is strictly concave on $[-1,1]^2 \subset \mathbb{R}^2$.*

Proof of Lemma 6.1. The Hessian matrix of the Hamiltonian $H_{N,\boldsymbol{X}}$ is $-N(\mathcal{X}\mathcal{M}\mathcal{X})$. As $\alpha \neq 1$, $\mathcal{M}$ is positive definite. Furthermore, $X \neq \{0,1\}$, and so $\mathcal{X}$ is diagonal with positive entries, which implies $\mathcal{X}\mathcal{M}\mathcal{X}$ is positive definite. The Hessian matrix of $H_{N,\boldsymbol{X}}$ is then negative definite, which implies that $H_{N,\boldsymbol{X}}$ is strictly concave. □

Lemma 6.2. *Consider a 2-gCW network with N sites, relative partition sizes vector $\boldsymbol{X} = (X, 1-X)$ with $X \in S_N \setminus \{0,1\}$, and the interaction matrix $\mathcal{M}$ from (38). If also $\alpha \in [0,1)$, then a state of this network is stable or metastable only if it is on the boundary of the state lattice. That is,*

$$E^{\mathrm{s}} \subseteq \partial\mathcal{L}_{N,\boldsymbol{X}}, \qquad E^{\mathrm{m}} \subseteq \partial\mathcal{L}_{N,\boldsymbol{X}}.$$

Proof of Lemma 6.2. Fix any state $\boldsymbol{m}' \in \mathcal{L}_{N,\boldsymbol{X}} \setminus \partial\mathcal{L}_{N,\boldsymbol{X}}$ (it is not on the boundary of $\mathcal{L}_{N,\boldsymbol{X}}$), and let $B \subset \partial\mathcal{L}_{N,\boldsymbol{X}}$ be the set of boundary states to which $\boldsymbol{m}'$ is closest (minimizing the Euclidean distance). $\boldsymbol{m}'$ has at least one neighbor $\boldsymbol{s}' \in \mathcal{L}_{N,\boldsymbol{X}}$ (possibly in B itself) such that $\boldsymbol{s}'$ is closer to B than $\boldsymbol{m}'$ is to B.

By Lemma 6.1 $H_{N,\boldsymbol{X}}$ is strictly concave. Therefore the stationary point of $H_{N,\boldsymbol{X}}$ at $(0,0)$ is unique, and it is also the unique maximum. The transition $\boldsymbol{m}' \to \boldsymbol{s}'$ must then decrease the energy, as we move further away from $(0,0)$. Therefore, $\boldsymbol{s}' \in E^-(\boldsymbol{m}')$, and $\boldsymbol{m}'$ cannot be a stable state.

The 'trivial' path $\{\boldsymbol{m}', \boldsymbol{s}'\}$ connects the sets $\{\boldsymbol{m}'\}$ and $\{\boldsymbol{s}'\}$, so the communication height is bounded by the maximum energy on this path:

$$\Phi_{N,\boldsymbol{X}}\left[\{\boldsymbol{m}'\}, \{\boldsymbol{s}'\}\right] \leq H_{N,\boldsymbol{X}}(\boldsymbol{m}').$$

As

$$0 \leq \Phi_{N,\boldsymbol{X}}\left[\{\boldsymbol{m}'\}, E^{-}(\boldsymbol{m}')\right] \leq \Phi_{N,\boldsymbol{X}}\left[\{\boldsymbol{m}'\}, \{\boldsymbol{s}'\}\right],$$

we have that the stability level

$$0 \leq V_{N,\boldsymbol{X}}(\boldsymbol{m}') \leq \Phi_{N,\boldsymbol{X}}\left[\{\boldsymbol{m}'\}, \{\boldsymbol{s}'\}\right] - H_{N,\boldsymbol{X}}(\boldsymbol{m}') \leq 0.$$

$V_{N,\boldsymbol{X}}(\boldsymbol{m}') = 0$, and so by definition $\boldsymbol{m}'$ cannot be a metastable state.

□

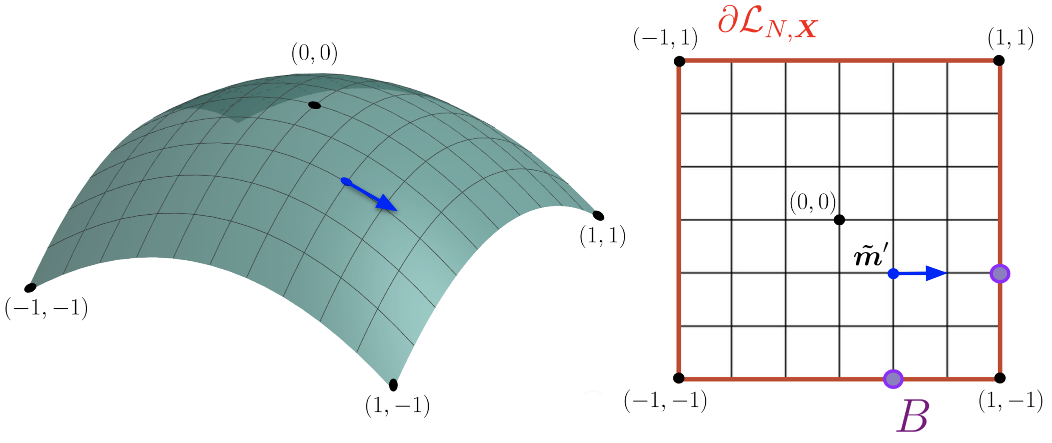

(0, 0)
(1, 1)
(−1, −1)
(1, −1)
$\partial\mathcal{L}_{N,\boldsymbol{X}}$
(−1, 1)
(1, 1)
(0, 0)
$\tilde{\boldsymbol{m}}'$
(−1, −1)
B
(1, −1)

Figure 30: A schematic overview of the proof of Lemma 6.2. Any path in a 2-gCW network is a walk on $\mathcal{L}_{N,X}$. The Hamiltonian is concave, and has unique maximum at $(0, 0)$. If a walk is not on the boundary of $\mathcal{L}_{N,X}$, it is possible to move towards the nearest boundary, and this decreases the energy (the blue arrow is an example of this).

Theorem 6.10 (Metastability of 2-gCW switching network)**.** *Consider a 2-gCW network with N sites, interaction matrix $\mathcal{M}$ from (38) and increasing partition sizes $(n_1^{(i)})_{i\in\mathbb{N}}, (n_2^{(i)})_{i\in\mathbb{N}}$ which are a sequence such that $n_1^{(i+1)} \geq n_1^{(i)} + 1$ and $n_2^{(i+1)} \geq n_2^{(i)} + 1$ for all $i \in \mathbb{N}$.*
Define

$$\boldsymbol{X}_i = (X_i, 1 - X_i) := \left(\frac{n_1^{(i)}}{n_1^{(i)} + n_2^{(i)}}, \frac{n_2^{(i)}}{n_1^{(i)} + n_2^{(i)}} \right),$$

and $\boldsymbol{Y} := \lim_i \boldsymbol{X}_i$. Furthermore, let $(n_1^{(i)})_{i\in\mathbb{N}}, (n_2^{(i)})_{i\in\mathbb{N}}$ be such that $\boldsymbol{X}_i \notin \{0, 1\}$ for all $i \in \mathbb{N}$.
Let $\alpha \in [0, 1)$, and assume that

$$0 \leq \frac{\gamma}{2 + 2\alpha + \gamma} \frac{1 - X_i}{X_i} < \frac{1}{3}, \qquad 0 \leq \frac{\gamma}{2 - 2\alpha + \gamma} \frac{X_i}{1 - X_i} < \frac{1}{3}, \qquad \forall i \in \mathbb{N},$$

i.e. γ is not too large.

1. *If $\gamma = 0$, for any $i \in \mathbb{N}$ we have $E^{\mathrm{s}} = \mathcal{C}$ and $E^{\mathrm{m}} = \varnothing$; there is no metastability for any sequence of increasing partition sizes.*
2. *If $\gamma > 0$ and*
$$\frac{X_i^2}{(1 - X_i)^2} < \frac{2 - 2\alpha + \gamma}{2 + 2\alpha + \gamma} \qquad \forall i \in \mathbb{N},$$
the maximum stability level of the network is
$$\lim_{i\to\infty} \frac{1}{N_i} \Gamma_{N_i,\boldsymbol{X}_i} = \frac{1}{2} \left[(2 + 2\alpha + \gamma) Y^2 - 2\gamma Y (1 - Y) + \frac{\gamma^2}{2 + 2\alpha + \gamma} (1 - Y)^2 \right]. \tag{40}$$
$E^{\mathrm{s}} = \{(1,1)^\intercal, (-1,-1)^\intercal\}$, $E^{\mathrm{m}} = \{(-1,1)^\intercal, (1,-1)^\intercal\}$. The minimal limit gate for $(-1, 1)$ and E^- is $\{(u_1^+, 1)\}$, and the limit gate for $(1, -1)$ and E^- is $\{(u_1^-, -1)\}$.
3. *If $\gamma > 0$ and*
$$\frac{X_i^2}{(1 - X_i)^2} > \frac{2 - 2\alpha + \gamma}{2 + 2\alpha + \gamma} \qquad \forall i \in \mathbb{N},$$
the maximum stability level of the network is
$$\lim_{i\to\infty} \frac{1}{N_i} \Gamma_{N_i,\boldsymbol{X}_i} = \frac{1}{2} \left[(2 - 2\alpha + \gamma)(1 - Y)^2 - 2\gamma Y (1 - Y) + \frac{\gamma^2}{2 - 2\alpha + \gamma} Y^2 \right]. \tag{41}$$
$E^{\mathrm{s}} = \{(1,1)^\intercal, (-1,-1)^\intercal\}$, $E^{\mathrm{m}} = \{(-1,1)^\intercal, (1,-1)^\intercal\}$. The minimal limit gate for $(-1, 1)$ and E^- is $\{(1, u_2^-)\}$, and the limit gate for $(1, -1)$ and E^- is $\{(1, u_2^+)\}$.
4. *If $\gamma > 0$ and*
$$\frac{X_i^2}{(1 - X_i)^2} = \frac{2 - 2\alpha + \gamma}{2 + 2\alpha + \gamma} \qquad \forall i \in \mathbb{N},$$
then (40) and (41) are equal, and the maximum stability level is the same as in these equations. $E^{\mathrm{s}} = \{(1,1)^\intercal, (-1,-1)^\intercal\}$, $E^{\mathrm{m}} = \{(-1,1)^\intercal, (1,-1)^\intercal\}$. The minimal limit gate for $(-1, 1)$ and E^- is $\{(u_1^+, 1), (1, u_2^-)\}$, and the minimal limit gate for $(1, -1)$ and E^- is $\{(u_1^-, -1), (1, u_2^+)\}$.

Remark 6.10.1. *The network shows different metastable behaviour depending on how the partitions are grown as $N \to \infty$. For $Y \geq 1/2$, we will find the maximum stability level to be that of (41) for all $\alpha \in [0, 1)$ and $\gamma > 0$.*

However, when one has a network in which the first partition is smaller than the second, but not too small, for example $Y = 0.40$, two different regimes of α appear, and the metastable behaviour is radically different for both. Neglecting the very small γ, if $\alpha < 0.86$ we have

$$\lim_{i\to\infty} \frac{1}{N_i} \Gamma_{N_i,\boldsymbol{X}_i} = 0.16 \cdot (1 + \alpha),$$

and the maximum stability level (and expected transition time) increases with α; for large N, there are almost no transitions. If $\alpha > 0.86$ we have

$$\lim_{i\to\infty} \frac{1}{N_i} \Gamma_{N_i,\boldsymbol{X}_i} = 0.36 \cdot (1 - \alpha),$$

and the maximum stability level (and expected transition time) decreases with α. The maximum stability level has a discontinuity at $\alpha = 0.86$, at which there is a jump of size 0.25 in units of energy per site. There is *a nontrivial value of α for which transitions between states 1 and 2 suddenly start to occur. Lastly, when $Y \leq (\sqrt{3}-1)/2$ (neglecting γ), we have the maximum stability level of (40) for all $\alpha \in [0,1)$.*

Remark 6.10.2. *Metastability is a finite-size effect. In the limit $\beta \to \infty$, for infinitely large systems ($N \to \infty$) ergodicity is broken, and a network in state $(1,-1)$ will almost surely stay in that state; the metastability disappears. Yet, in Theorem 6.10, we take limits $N \to \infty$. As we divide by N first, we actually compute a stability **energy density** rather than a stability **energy**. As in our model the maximum stability level is a linear function in N, its 'density' $\frac{1}{N}\Gamma_{N,\boldsymbol{X}}$ does not depend on N at all; however, we can still derive all metastability properties from knowledge of the density.*

We still need to take a limit $N \to \infty$, as we first compute the maximum stability level density for a continuous state space (which is easier), and then want to compare it to the discrete state space case; in the limit these become equal. The limit is thus taken to be able to compare maximum stability level densities for the discrete and continuous case: we approximate the large finite system with an infinite system.

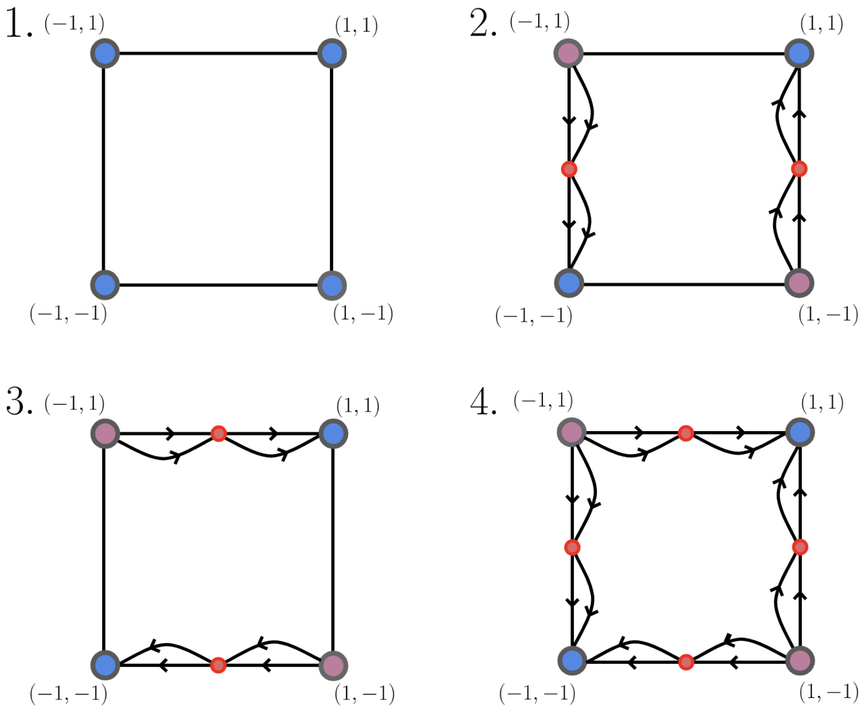


Figure 31: The four metastability scenarios in Theorem 6.10. Blue: stable states, purple: metastable states, red: limit gates. The arrows show schematically what sort of paths the network takes from metastability to stability in each scenario. Notice that all paths must pass through a gate.

Proof of 6.10. For the first part of the proof, fix any $i \in \mathbb{N}$. By Lemma 6.2, we know that (meta)stable states can only be found at the boundaries of the state lattice. First we analyze the Hamiltonian on the boundaries.

$$H_{N,X_i}((m_1,\pm1)^\intercal) = -\frac{N}{2}\left[(2+2\alpha+\gamma)X_i^2 m_1^2 + (2-2\alpha+\gamma)\left(1-X_i\right)^2 \pm 2\gamma X_i\left(1-X_i\right)m_1\right],$$
$$H_{N,X_i}((\pm1,m_2)^\intercal) = -\frac{N}{2}\left[(2+2\alpha+\gamma)X_i^2 + (2-2\alpha+\gamma)\left(1-X_i\right)^2 m_2^2 \pm 2\gamma X_i\left(1-X_i\right)m_2\right].$$

Let $\alpha \in [0,1)$ and $X_i \in S_N \setminus \{0,1\}$. We compute the locations of the maxima of the Hamiltonian on the intervals $J_1^\pm := \mathbb{R} \times \{\pm1\}$, and $J_2^\pm := \{\pm1\} \times \mathbb{R}$; we denote these locations as $u_1^\pm, u_2^\pm$ respectively.

$$\begin{aligned} \frac{\partial H_{N,X_i}}{\partial m_1}((u_1^\pm,\pm1)^\intercal) = 0 \qquad &\Leftrightarrow \qquad u_1^\pm = \mp\frac{\gamma}{2+2\alpha+\gamma}\frac{1-X_i}{X_i} \\ \frac{\partial H_{N,X_i}}{\partial m_2}((\pm1,u_2^\pm)^\intercal) = 0 \qquad &\Leftrightarrow \qquad u_2^\pm = \mp\frac{\gamma}{2-2\alpha+\gamma}\frac{X_i}{1-X_i}. \end{aligned} \tag{42}$$

We know these are the unique maxima and not minima on $J_1^\pm, J_2^\pm$, as the Hamiltonian is strictly concave. Note that by our starting assumptions, $u_1^\pm, u_2^\pm \in (-1,1) \subset \mathbb{R}$, so on all four intervals $[-1,1]\times\{\pm1\}$, $\{\pm1\}\times[-1,1]$ the Hamiltonian has a unique maximum. As the Hamiltonian is strictly concave, it follows that any minimum of the Hamiltonian on $\mathcal{L}_{N,\boldsymbol{X}_i}$ must also be a corner of $\mathcal{L}_{N,\boldsymbol{X}_i}$. The (meta)stable states can only be found on the corners of the lattice: $E^{\mathrm{s}} \subseteq \mathcal{C}$ and $E^{\mathrm{m}} \subseteq \mathcal{C}$.

We are now in a good position to determine which states are stable and metastable, and what the optimal paths are that connects these states. When $\gamma = 0$, all states in $\mathcal{C}$ have the same energy, and they were the locations of the minima of the Hamiltonian on the state lattice, so $E^{\mathrm{s}} = \mathcal{C}$ and $E^{\mathrm{m}} = \varnothing$.

When $\gamma > 0$, the intuition is as follows. The energy perturbation decreases the energy of the Hopfield states $(1,0)^\intercal$ and $(-1,0)^\intercal$, which correspond to the 2-gCW states $(1,1)^\intercal$ and $(-1,-1)^\intercal$ respectively. Of all four states in $\mathcal{C}$, $(1,1)^\intercal$ and $(-1,-1)^\intercal$ have the same energy, and their energy is lower than that of the states $(1,-1)^\intercal$ and $(-1,1)^\intercal$, so $E^{\mathrm{s}} = \{(1,1)^\intercal, (-1,-1)^\intercal\}$. The states $(1,-1)^\intercal$ and $(-1,1)^\intercal$ have higher energy, but they could still be metastable. We make the *Ansatz* that they are, and investigate the optimal paths from the presumed metastable states to the stable states.

Consider the line segment $\ell^+ = \ell_1^+ \cup \ell_2^+ \subset \mathbb{R}^2$, where

$$\ell_1^+ := \{u_1^+\} \times [u_2^+, 1],$$
$$\ell_2^+ := [u_1^+, 1] \times \{u_2^+\}.$$

This line segment can be constructed by starting one line at the maximum of the Hamiltonian on the top boundary, and one at the maximum on the right boundary, and moving downward or to the left respectively, until the lines meet.

We also construct the set of states $L^+ \subset \mathrm{Im}(\boldsymbol{m})$ that are directly to the left, or below the line ℓ^+. $L^+ = L_1^+ \cup L_2^+$, with

$$L_1^+ := \mathcal{L}_{N,\boldsymbol{X}_i} \cap ([u_1^+ - 2/n_1^{(i)}, u_1^+] \times [u_2^+, 1]),$$
$$L_2^+ := \mathcal{L}_{N,\boldsymbol{X}_i} \cap ([u_1^+, 1] \times [u_2^+ - 2/n_2^{(i)}, u_2^+]),$$

and note that these sets are nonempty for large enough $n_1^{(i)}, n_2^{(i)}$. To lighten notation, we will not explicitly denote the dependence of L^+ on $n_1^{(i)}, n_2^{(i)}$, but keep this dependence in mind.

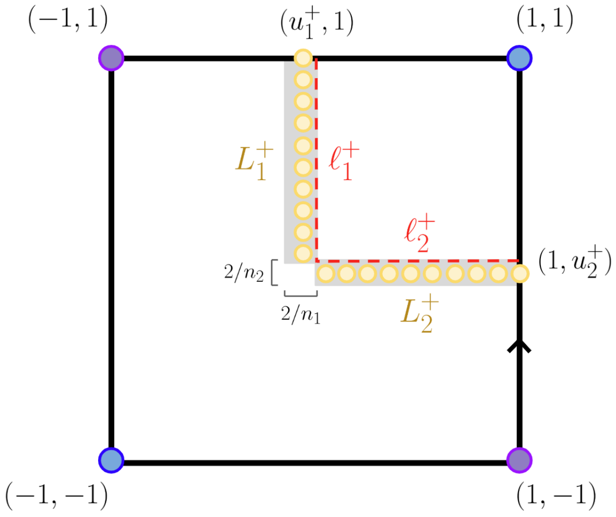


Figure 32: The construction of the set of lattice points L^+ (yellow); this set is constructed from the line segment ℓ^+ (red).

Let us fix any $\varepsilon > 0$. By continuity of $H_{N,\boldsymbol{X}_i}$ as a function on $\mathbb{R}^2$, we can take $n_1^{(i)}, n_2^{(i)}$ large enough such that both

1) for any $\boldsymbol{m}' = (m_1', m_2')^\intercal \in L_1^+$, $|H_{N,\boldsymbol{X}_i}(\boldsymbol{m}) - H_{N,\boldsymbol{X}_i}\left((u_1^+, m_2')^\intercal\right)| < \varepsilon$, and

2) for any $\boldsymbol{s}' = (s_1', s_2')^\intercal \in L_2^+$, $|H_{N,\boldsymbol{X}_i}(\boldsymbol{s}) - H_{N,\boldsymbol{X}_i}\left((s_1', u_2^+)^\intercal\right)| < \varepsilon$.

This implies that for $n_1^{(i)}, n_2^{(i)}$ large enough,

$$\min_{\boldsymbol{m}' \in L^+} H_{N,\boldsymbol{X}_i}(\boldsymbol{m}') > \min_{\boldsymbol{m}' \in \ell^+} H_{N,\boldsymbol{X}_i}(\boldsymbol{m}') - \varepsilon. \tag{43}$$

As $H_{N,\boldsymbol{X}_i}$ is strictly concave, its restriction to ℓ_1^+ is also strictly concave. As ℓ_1^+ is a compact convex set, the minimum of $H_{N,\boldsymbol{X}_i}$ on ℓ_1^+ is at the extremal points of ℓ_1^+, which are $\{(u_1^+, 1), (u_1^+, u_2^+)\}$. We compute the difference in energy between these states. Using that $u_1^+ > -1$ and $u_2^+ \in (-1,1) \subset \mathbb{R}$,

$$\begin{aligned} H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal) - H_{N,\boldsymbol{X}_i}((u_1^+,u_2^+)^\intercal) &= -\frac{N}{2}\left[(2-2\alpha+\gamma)(1-X_i)^2(1-(u_2^+)^2) + 2\gamma X_i(1-X_i)\,u_1^+(1-u_2^+)\right] \\ &< -\frac{N}{2}\left[(2-2\alpha+\gamma)(1-X_i)^2(1-(u_2^+)^2) - 2\gamma X_i(1-X_i)(1-u_2^+)\right]. \end{aligned}$$

So if

$$(2-2\alpha+\gamma)(1-X_i)^2(1-(u_2^+)^2) > 2\gamma X_i(1-X_i)(1-u_2^+), \tag{44}$$

then $H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal) - H_{N,\boldsymbol{X}_i}((u_1^+,u_2^+)^\intercal) < 0$.

Using (42), after some treatment we can rewrite inequality (44) as

$$3u_2^+ - 2u_2^+ - 1 < 0,$$

which holds if $u_2^+ \in (-1/3, 1) \subset \mathbb{R}$. This is true by our starting assumptions, and so $H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal) < H_{N,\boldsymbol{X}_i}((u_1^+,u_2^+)^\intercal)$.

Similarly, the locations of the minima of $H_{N,\boldsymbol{X}_i}$ restricted to ℓ_2^+ are $(1,u_2^+)$ and (u_1^+,u_2^+). One can then use the exact same approach, using that $u_2^+ > -1$ and $u_1^+ \in (-1,1) \subset \mathbb{R}$, to show that $H_{N,\boldsymbol{X}_i}((1,u_2^+)^\intercal) < H_{N,\boldsymbol{X}_i}((u_1^+,u_2^+)^\intercal)$ if $u_1^+ \in (-1/3,1) \subset \mathbb{R}$; and this was true by our starting assumptions.

The minima of $H_{N,\boldsymbol{X}_i}$ restricted to the full line segment ℓ^+ must then be located at either $(u_1^+,1),(1,u_2^+)$, or both. We will see different scenarios of metastable behaviour for the different possibilities.

Notice that the Hamiltonian has a useful 'parity' symmetry: $H_{N,\boldsymbol{X}_i}(\boldsymbol{m}) = H_{N,\boldsymbol{X}_i}(-\boldsymbol{m})$. Let us construct the 'parity-flipped' versions of ℓ^+ and L^+: $\ell^- = \ell_1^- \cup \ell_2^-$, with

$$\begin{aligned}\ell_1^- &:= \{u_1^-\} \times [-1, u_2^-],\\ \ell_2^- &:= [-1, u_1^-] \times \{u_2^-\},\end{aligned}$$

and $L^- = L_1^- \cup L_2^-$, with

$$\begin{aligned}L_1^- &:= \mathcal{L}_{N,\boldsymbol{X}_i} \cap ([u_1^-, u_1^- + 2/n_1^{(i)}] \times [-1, u_2^-]),\\ L_2^- &:= \mathcal{L}_{N,\boldsymbol{X}_i} \cap ([-1, u_1^-] \times [u_2^-, u_2^- + 2/n_2^{(i)}]),\end{aligned}$$

where we recalled that $u_1^- = -u_1^+, u_2^- = -u_2^+$. By the parity symmetry we then immediately obtain the following results: for any $\varepsilon > 0$ there are $n_1^{(i)}, n_2^{(i)}$ large enough such that

$$\min_{\boldsymbol{m}' \in L^-} H_{N,\boldsymbol{X}_i}(\boldsymbol{m}') > \min_{\boldsymbol{m}' \in \ell^-} H_{N,\boldsymbol{X}_i}(\boldsymbol{m}') - \varepsilon, \tag{45}$$

and the minima of $H_{N,\boldsymbol{X}_i}$ restricted to ℓ^- are located at either $(u_1^-,-1),(-1,u_2^-)$, or both.

We will use these results to derive the maximum stability level of the network. First, we would like to know what the set of states with lower energy than the presumed metastable states is. Note that $H_{N,\boldsymbol{X}_i}((1,-1)^\intercal) = H_{N,\boldsymbol{X}_i}((-1,1)^\intercal)$, and so we define $E^- := E^-((1,-1)^\intercal) = E^-((-1,1)^\intercal)$. Let $B_1^+ := [-1,1] \times \{+1\}$ be the top boundary of the square $[-1,1]^2 \in \mathbb{R}^2$, and similarly define the left boundary $B_2^- := \{-1\} \times [-1,1]$, right boundary $B_2^+ := \{+1\} \times [-1,1]$ and bottom boundary $B_1^- := [-1,1] \times \{-1\}$.

Let $M_1' \subset (-1,1)^2 \subset \mathbb{R}^2$ be the interior of the rectangle $B_1^+ \cup B_2^- \cup \ell_1^+ \cup \ell_2^-$. Let M_1 be the closure of M_1'. M_1 is a compact convex set, so the minima of $H_{N,\boldsymbol{X}_i}$ restricted to M_1 are at the boundary of the set M_1. Given our starting assumptions on the parameters, the minima of $H_{N,\boldsymbol{X}_i}$ on ℓ_1^+, ℓ_2^- were located at $(u_1^+,1),(-1,u_2^-)$ respectively. However, $(u_1^+,1),(-1,u_2^-)$ were the locations of the maxima on the boundary $B_1^+ \cup B_2^-$ respectively, and so the minima of $H_{N,\boldsymbol{X}_i}$ restricted to M_1 must be located on the boundary $B_1^+ \cup B_2^-$. By strict convexity, the minimum of $H_{N,\boldsymbol{X}_i}$ restricted to M_1 must then be at the corner $(-1,1)$. This in turn means that $E^- \cap M_1 = \varnothing$.

Any path from $(-1,1) \in M_1$ to E^- must therefore have at least one state outside $M_1 \cap \mathcal{L}_{N,\boldsymbol{X}_i}$. By definition of L_1^+ and L_2^-, and Corollary 6.3.1, this path must therefore have at least one state in $L_1^+ \cup L_2^-$.

Similarly, we let M_2' be the interior of be the interior of the rectangle $B_1^- \cup B_2^+ \cup \ell_1^- \cup \ell_2^+$. Let M_2 be the closure of M_2'. By a similar argument as before, the minimum of $H_{N,\boldsymbol{X}_i}$ restricted to M_2 must be located at the corner $(1,-1)$. This in turn means that $E^- \cap M_2 = \varnothing$.

Any path from $(1,-1) \in M_2$ to E^- must have at least one state outside $M_2 \cap \mathcal{L}_{N,\boldsymbol{X}_i}$. By definition of L_1^- and L_2^+, and Corollary 6.3.1, this path must therefore have at least one state in $L_1^- \cup L_2^+$.

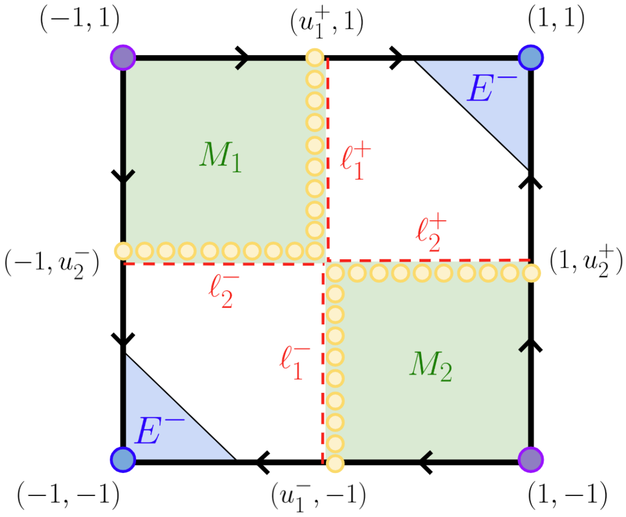


Figure 33: Any path from the presumed metastable states (purple) to the set of lower energy states that contains the stable states $(1,1),(-1,-1)$ must cross the states in the set $L^+ \cup L^-$ (yellow); this set is constructed from the line segments ℓ^+ and ℓ^- (red).

Before we go on, we need the following calculation. As $H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal), H_{N,\boldsymbol{X}_i}((1,u_2^+)^\intercal) < 0$, we have that $H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal) < H_{N,\boldsymbol{X}_i}((1,u_2^+)^\intercal)$ iff $|H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal)| > |H_{N,\boldsymbol{X}_i}((1,u_2^+)^\intercal)|$ iff $|H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal)|/|H_{N,\boldsymbol{X}_i}((1,u_2^+)^\intercal)| > 1$. After some algebra, we find

$$\frac{|H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal)|}{|H_{N,\boldsymbol{X}_i}((1,u_2^+)^\intercal)|} = \frac{(1-X_i)^2}{X_i^2}\frac{2-2\alpha+\gamma}{2+2\alpha+\gamma}.$$

There are now three scenarios:

1.
$$\frac{X_i^2}{(1-X_i)^2} < \frac{2-2\alpha+\gamma}{2+2\alpha+\gamma}. \tag{46}$$

Then, $H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal) < H_{N,\boldsymbol{X}_i}((1,u_2^+)^\intercal)$, and so

$$\min_{\boldsymbol{m}'\in\ell^+} H_{N,\boldsymbol{X}_i}(\boldsymbol{m}') = H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal) = -\frac{N}{2}(1-X_i)^2\left[2-2\alpha+\gamma-\frac{\gamma^2}{2+2\alpha+\gamma}\right].$$

We had that $H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal) = H_{N,\boldsymbol{X}_i}((u_1^-,-1)^\intercal)$, and $H_{N,\boldsymbol{X}_i}((1,u_2^+)^\intercal) = H_{N,\boldsymbol{X}_i}((-1,u_2^-)^\intercal)$, so we also have

$$\min_{\boldsymbol{m}'\in\ell^-} H_{N,\boldsymbol{X}_i}(\boldsymbol{m}') = H_{N,\boldsymbol{X}_i}((u_1^-,-1)^\intercal) = -\frac{N}{2}(1-X_i)^2\left[2-2\alpha+\gamma-\frac{\gamma^2}{2+2\alpha+\gamma}\right].$$

We saw that any path from $(-1,1)$ to E^- must have at least one state in $L_1^+\cup L_2^-$. Recall (43) and fix an $\varepsilon>0$. There are $n_1^{(i)}, n_2^{(i)}$ large enough such that

$$\Phi_{N,\boldsymbol{X}_i}\left[\{(-1,1)^\intercal\},E^-\}\right] \geq \min_{\boldsymbol{m}'\in L_1^+\cup L_2^-} H_{N,\boldsymbol{X}_i}(\boldsymbol{m}') = \min_{\boldsymbol{m}'\in L^+} H_{N,\boldsymbol{X}_i}(\boldsymbol{m}') > H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal)-\varepsilon, \tag{47}$$

where the middle equality comes from parity symmetry.

Now, consider the path $\boldsymbol{r} = \{\boldsymbol{r}(t) = (-1+\frac{2t}{n_1^{(i)}},1)\in\mathbb{R}^2 : t=0,...,n_1^{(i)}\}$, which starts in $(-1,1)$ and runs along the boundary until it reaches $(1,1)$. As $(u_1^+,1)$ is the location of the maximum on B_1^+, and $\boldsymbol{r}\subseteq B_1^+$ for all $n_1^{(i)}\in\mathbb{N}$, we have that

$$\max_{t=0,\ldots,n_1^{(i)}} H_{N,\boldsymbol{X}_i}(\boldsymbol{r}(t)) \leq H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal). \tag{48}$$

However, as $(1,1)\in E^-$, $\boldsymbol{r}$ is a path from $(-1,1)$ to E^-, and so

$$\Phi_{N,\boldsymbol{X}_i}\left[\{(-1,1)^\intercal\},E^-\}\right] \leq \max_{t=0,\ldots,n_1^{(i)}} H_{N,\boldsymbol{X}_i}(\boldsymbol{r}(t)) \leq H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal).$$

Combining (47) and (48) then gives

$$H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal) - \varepsilon < \Phi_{N,\boldsymbol{X}_i}\left[\{(-1,1)^\intercal\}, E^-\}\right] \leq H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal). \tag{49}$$

We now find the minimal limit gate. Let

$$W_{(n_1^{(i)},n_2^{(i)})}((u_1^+,1)^\intercal)) := \{\boldsymbol{m} \in \mathrm{Im}(\boldsymbol{m}) : |H_{N,\boldsymbol{X}_i}(\boldsymbol{m}) - H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal)| \leq \varepsilon\},$$

where $n_1^{(i)}, n_2^{(i)}$ are large enough such that (49) holds. By (49), and the fact that any path from $(-1,1)$ to E^- has at least one state in $L_1^+ \cup L_2^-$, it follows that the set $W_{(n_1^{(i)},n_2^{(i)})}((u_1^+,1)^\intercal)) \cap (L_1^+ \cup L_2^-)$ is a gate for $(-1,1)$ and E^-. However, as $W_{(n_1^{(i)},n_2^{(i)})}((u_1^+,1)^\intercal)) \cap L_2^- = \varnothing$ for small enough ε, the set

$$W_1^+((n_1^{(i)},n_2^{(i)})) := W_{(n_1^{(i)},n_2^{(i)})}((u_1^+,1)^\intercal)) \cap L_1^+$$

is a smaller gate for $(-1,1)$ and E^-. Importantly, the minimal gate for $(-1,1)$ and E^- is a subset of $W_1^+((n_1^{(i)},n_2^{(i)}))$.

By continuity of $H_{N_i,\boldsymbol{X}_i}$, for arbitrary $\delta > 0$ there are $n_1^{(i)}, n_2^{(i)}$ large enough such that

$$W_1^+((n_1^{(i)},n_2^{(i)})) \subset \mathcal{B}_\delta((u_1^+,1)),$$

and so the minimal gate for $(-1,1)$ and E^- is a subset of $\mathcal{B}_\delta((u_1^+,1))$ for large enough $n_1^{(i)}, n_2^{(i)}$. As there exist limit gates for $(-1,1)$ and E^-, the minimal limit gate is nonempty. However, it can also not be smaller than the singleton $\{(u_1^+,1)\}$, which is a limit gate. Therefore $\{(u_1^+,1)\}$ is the minimal limit gate.

We continue the derivation of the maximum stability level. We have that

$$H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal) - H_{N,\boldsymbol{X}_i}((-1,1)^\intercal) = \frac{N}{2}\left[(2+2\alpha+\gamma)X_i^2 - 2\gamma X_i(1-X_i) + \frac{\gamma^2}{2+2\alpha+\gamma}(1-X_i)^2\right] > 0. \tag{50}$$

By parity symmetry, we also immediately obtain

$$H_{N,\boldsymbol{X}_i}((u_1^-,-1)^\intercal) - \varepsilon < \Phi_{N,\boldsymbol{X}_i}\left[\{(1,-1)^\intercal\}, E^-\}\right] \leq H_{N,\boldsymbol{X}_i}((u_1^-,-1)^\intercal), \tag{51}$$

and

$$H_{N,\boldsymbol{X}_i}((u_1^-,-1)^\intercal) - H_{N,\boldsymbol{X}_i}((1,-1)^\intercal) = H_{N,\boldsymbol{X}_i}((u_1^+,1)^\intercal) - H_{N,\boldsymbol{X}_i}((-1,1)^\intercal). \tag{52}$$

We will now take a limit by increasing partition sizes. The increasing partition sizes $n_1^{(i)}, n_2^{(i)}$ are a sequence $(n_1^{(i)})_{i\in\mathbb{N}}, (n_2^{(i)})_{i\in\mathbb{N}}$ where $(n_1^{(i)})_{i+1} \geq (n_1^{(i)})_i + 1$ and $(n_2^{(i)})_{i+1} \geq (n_2^{(i)})_i + 1$ $\forall i \in \mathbb{N}$. This ensures that if i is large, then $n_1^{(i)}, n_2^{(i)}$ and $N_i := n_1^{(i)} + n_2^{(i)}$ are all large.

Assume that (46) holds for all $X \in \{X_i : i \in \mathbb{N}\}$. Fix an $\varepsilon > 0$. For i large enough, equations (49), (50), (51), (52) all hold, and recalling the definition of the stability level $V_{N,\boldsymbol{X}}$, we conclude that

$$\begin{aligned}\lim_{i\to\infty}\frac{1}{N_i}V_{N_i,\boldsymbol{X}_i}((-1,1)^\intercal) &= \lim_{i\to\infty}\frac{1}{N_i}V_{N_i,\boldsymbol{X}_i}((1,-1)^\intercal)\\ &= \frac{1}{2}\left[(2+2\alpha+\gamma)Y^2 - 2\gamma Y(1-Y) + \frac{\gamma^2}{2+2\alpha+\gamma}(1-Y)^2\right].\end{aligned}$$

We now consider the limit of the maximum stability level $\Gamma_{N,\boldsymbol{X}}$. Notice that the only states with nonzero stability level were the corner states $\mathcal{C}$. $\Gamma_{N,\boldsymbol{X}}$ is the maximum of the stability level over all non-stable states, and so for any $i \in \mathbb{N}$ we have that $\Gamma_{N_i,\boldsymbol{X}_i} = \max\{V_{N_i,\boldsymbol{X}_i}((-1,1)^\intercal), V_{N_i,\boldsymbol{X}_i}((1,-1)^\intercal)\}$. Furthermore, as both $V_{N_i,\boldsymbol{X}_i}((-1,1)^\intercal)$ and $V_{N_i,\boldsymbol{X}_i}((1,-1)^\intercal)$ have the same limit, we conclude

$$\begin{aligned}\lim_{i\to\infty}\frac{1}{N_i}\Gamma_{N_i,\boldsymbol{X}_i} &= \lim_{i\to\infty}\frac{1}{N_i}V_{N_i,\boldsymbol{X}_i}((-1,1)^\intercal)\\ &= \lim_{i\to\infty}\frac{1}{N_i}V_{N_i,\boldsymbol{X}_i}((1,-1)^\intercal)\\ &= \frac{1}{2}\left[(2+2\alpha+\gamma)Y^2 - 2\gamma Y(1-Y) + \frac{\gamma^2}{2+2\alpha+\gamma}(1-Y)^2\right].\end{aligned} \tag{53}$$

Lastly, we find the metastable states. From (53) we conclude that as $i \to \infty$, $E^{\mathrm{m}} = \{(-1,1)^\intercal, (1,-1)^\intercal\}$.

Next, we look at scenario

2.
$$\frac{X^2}{(1-X)^2} > \frac{2-2\alpha+\gamma}{2+2\alpha+\gamma}. \tag{54}$$

Then, $H_{N,\boldsymbol{X}}((1,u_2^+)^\intercal) < H_{N,\boldsymbol{X}}((u_1^+,1)^\intercal)$, and so

$$\min_{\boldsymbol{m'}\in\ell^+} H_{N,\boldsymbol{X}}(\boldsymbol{m'}) = H_{N,\boldsymbol{X}}((1,u_2^+)^\intercal) = -\frac{N}{2}X^2\left[2+2\alpha+\gamma-\frac{\gamma^2}{2-2\alpha+\gamma}\right],$$

and

$$\min_{\boldsymbol{m'}\in\ell^-} H_{N,\boldsymbol{X}}(\boldsymbol{m'}) = H_{N,\boldsymbol{X}}((-1,u_2^-)^\intercal) = -\frac{N}{2}X^2\left[2+2\alpha+\gamma-\frac{\gamma^2}{2-2\alpha+\gamma}\right].$$

One can compute that

$$H_{N,\boldsymbol{X}}((1,u_2^+)^\intercal) - H_{N,\boldsymbol{X}}((-1,1)^\intercal) = \frac{N}{2}\left[(2-2\alpha+\gamma)(1-X)^2 - 2\gamma X(1-X) + \frac{\gamma^2}{2-2\alpha+\gamma}X^2\right] > 0.$$

Assume that (54) holds for all $X \in \{X_i : i \in \mathbb{N}\}$. Then, with a very similar analysis as in scenario 1, we obtain

$$\lim_{i\to\infty}\frac{1}{N_i}\Gamma_{N_i,\boldsymbol{X}_i} = \frac{1}{2}\left[(2-2\alpha+\gamma)(1-Y)^2 - 2\gamma Y(1-Y) + \frac{\gamma^2}{2-2\alpha+\gamma}Y^2\right]. \tag{55}$$

Similarly, we get that in the large partition sizes limit, $E^{\mathrm{m}} = \{(-1,1)^\intercal, (1,-1)^\intercal\}$. The limit gate for $(-1,1)$ and E^- is $\{(1,u_2^-)\}$, and the limit gate for $(1,-1)$ and E^- is $\{(1,u_2^+)\}$.

The last scenario is

3.
$$\frac{X^2}{(1-X)^2} = \frac{2-2\alpha+\gamma}{2+2\alpha+\gamma}. \tag{56}$$

As then $H_{N,\boldsymbol{X}}((u_1^+,1)^\intercal) = H_{N,\boldsymbol{X}}((1,u_2^+)^\intercal)$, we obtain the same maximum stability level as in (53) and (55), which are equal in this scenario. Again, we find in the large partition sizes limit, $E^{\mathrm{m}} = \{(-1,1)^\intercal, (1,-1)^\intercal\}$. The limit gate for $(-1,1)$ and E^- is $\{(u_1^+,1),(1,u_2^-)\}$, and the limit gate for $(1,-1)$ and E^- is $\{(u_1^-,-1),(1,u_2^+)\}$.

□

6.5 Metastability of the 2-pattern switching network

We apply the result of Theorem 6.10 to study the metastable behaviour of the 2-pattern switching network. Let us consider the case where the partition sizes $n_1^{(i)}$, $n_2^{(i)}$ are fixed at $N/2$: $n_1^{(i)} = n_2^{(i)} = N/2\ \forall i \in \mathbb{N}$. Let $\gamma > 0$. Then, from Theorem 6.10, we get

$$\lim_{N\to\infty}\frac{1}{N}\Gamma_{N,(1/2,1/2)} = \frac{1}{8}\left[2-2\alpha-\gamma+\frac{\gamma^2}{2-2\alpha+\gamma}\right]. \tag{57}$$

By Theorem 3.3, the Hamiltonian of the 2-gCW switching network with equal partition sizes is equal to the Hamiltonian of the 2-pattern switching network with orthogonal patterns. This means that the metastability level of the 2-pattern switching network with orthogonal patterns is also given by (57). Furthermore, it follows that the stable states of the 2-pattern switching network are $(1,0)$ and $(-1,0)$, and the metastable states are $(0,1)$ and $(0,-1)$.

We use approximation (37) to learn about the influence of the α parameter on the metastable network behaviour. As γ is usually of the order $1/N$, we can neglect it in the expression for the metastability level. The approximate expected time it takes the network to transition from state 2 $(0,1)$ to state 1 $(1,0)$ is then

$$\mathbb{E}_N^{(0,1)}[\tau_{(1,0)}] \approx e^{\frac{1}{4}\beta N(1-\alpha)}.$$

Keep in mind that this estimate is accurate for large β and N.

There are two important observations:

1. No matter the size of β and N, we can decrease the expected transition time arbitrarily by making α large enough. This is a very satisfactory result: nonzero (positive) values of the parameter α is really what allows the switching behaviour to appear in the 2-pattern switching network.
2. As we increase N, the expected transition time increases exponentially. This means that the state switching in the 2-pattern switching network only occurs for not too large values of N: the switching is a *finite-size effect*. Furthermore, if we think of the switching as random oscillations between state 1 and state 2, we see that the expected period of these oscillations increases exponentially with N.

7 Sparse 2-pattern switching network and embedded Markov chain

7.1 Embedding of two-state Markov chain in sparse block network

The embedding weights were proposed to embed an arbitrary two-state Markov chain in an ANN. When implemented into a Hopfield network, the network showed undesirable behaviour: during switching between two edge patterns, the third and fourth edge pattern could sometimes interfere. As a consequence, the transition probabilities of the embedded Markov chain can be different than the pre-programmed transition probabilities (as implemented in the network through the energy perturbation parameters γ). The Hopfield implementation of the embedding weights is not reliable.

For a successful implementation of the embedding weights, we instead turn to the sparse block network (Section 1.4.1). The sparsity and winner-take-all dynamics of this network circumvent the problems that arise in a 'dense' network such as the Hopfield network. We first demonstrate with a theoretical argument that the problems with a Hopfield implementation do not arise in a sparse block network implementation. After that, we present simulations of three different two-state Markov chains, embedded in a sparse block network, and we compare the results with 'ordinary' Markov chains (not embedded in an ANN).

7.2 The case for sparsity

In Section 5.5 we saw that the Hopfield implementation contained minima in the energy landscape that interfered with proper functioning of the embedded Markov chain, and the existence of such states was shown with a theoretical argument. The unwanted minima were states which had overlap with all four edge patterns (an example is the state $(1/2, 1/2, 1/4, 1/4)$).

Here we investigate heuristically where the unwanted overlap with edge patterns three and four comes from. This is done by analyzing the effect of the crossterm parameter α on the PSP of neurons of a Hopfield network with three patterns, of which the first two will mix through α; the third pattern will play the role of an unwanted pattern. After this analysis, we analyze the same construction for a sparse block network. In the Hopfield network, the third pattern will interfere for large α, while in the sparse block network it will not.

7.2.1 Hopfield network PSP

Consider the weights

$$w_{xy} = \xi_x^1\xi_y^1 + \xi_x^2\xi_y^2 + \alpha(\xi_x^1\xi_y^2 + \xi_x^2\xi_y^1) + \xi_x^3\xi_y^3, \tag{58}$$

We split the set of all neurons $V_N = \{1, ..., N\}$ into two groups: a group of all neurons for which the patterns agree and a group of neurons where the patterns don't agree (the usual trick, again). Let $\mathcal{V}_\mathrm{a} = \{x \in V_N : \xi_x^1 = \xi_x^2\}$ be the group of all neurons that agree on patterns 1 and 2, and $\mathcal{V}_\mathrm{d} = \{x \in V_N : \xi_x^1 \neq \xi_x^2\}$ be the group of all neurons that don't agree on patterns 1 and 2.

First we analyze the PSP in a Hopfield setting. It is crucial to remember that in a Hopfield network, neurons can only have two states: $+1$ (fire) and -1 (silent). Therefore, if patterns 1 and 2 disagree at neuron x (so $\xi_x^1 \neq \xi_x^2$), we must have that $\xi_x^1 = -\xi_x^2$. This fact is the origin of the trouble that's to come.

We can write the weight w_{xy} into different ways, depending on what groups x and y belong to.

- if x and y both belong to $\mathcal{V}_\mathrm{a}$, then $\xi_x^2 = \xi_x^1$, $\xi_y^2 = \xi_y^1$ and so
$$w_{xy} = (2 + 2\alpha)\xi_x^1\xi_y^1 + \xi_x^3\xi_y^3. \tag{59}$$
- if x and y both belong to $\mathcal{V}_\mathrm{d}$, then $\xi_x^2 = -\xi_x^1$, $\xi_y^2 = -\xi_y^1$ and so
$$w_{xy} = (2 - 2\alpha)\xi_x^1\xi_y^1 + \xi_x^3\xi_y^3. \tag{60}$$
- if x belongs to $\mathcal{V}_\mathrm{a}$, but y belongs to $\mathcal{V}_\mathrm{d}$ then $\xi_x^2 = \xi_x^1$, but $\xi_y^2 = -\xi_y^1$, and so
$$w_{xy} = \xi_x^3\xi_y^3. \tag{61}$$

Recall that the PSP of a neuron x at time t was defined as the sum of activity of all other neurons $y \neq x$, weighed by the weights matrix. As the self-weights w_{xx} are zero, we can just write the PSP as a sum over all neurons. As all neurons belong either to group $\mathcal{V}_\mathrm{a}$ or group $\mathcal{V}_\mathrm{b}$, we can write

$$h_x(t) = \sum_{y \in V_N} w_{xy}\sigma_y(t) = \sum_{y \in \mathcal{V}_\mathrm{a}} w_{xy}\sigma_y(t) + \sum_{y \in \mathcal{V}_\mathrm{d}} w_{xy}\sigma_y(t).$$

Assume neuron x belongs to $\mathcal{V}_\text{a}$. Then the PSP consists of two sums: one where x and y are both in group $\mathcal{V}_\text{a}$, and one where x is in group $\mathcal{V}_\text{a}$ but y is in group $\mathcal{V}_\text{d}$. Using the re-written weights above (equations (59), (60), (61)), we find

$$h_x(t) = \frac{1}{N}\left((2+2\alpha)\xi_x^1 \sum_{y\in\mathcal{V}_\text{a}} \xi_y^1\sigma_y(t)\right) + \frac{1}{N}\left(\xi_x^3 \sum_{y\in\mathcal{V}_\text{d}} \xi_y^3\sigma_y(t)\right). \tag{62}$$

For large amounts of neurons, the amount of neurons at which patterns 1 and 2 agree is approximately the same as the amount at which the patterns disagree. Equation (62) tells us that if $\sigma(t)$ agrees somewhat with ξ^1 on the neurons in group $\mathcal{V}_\text{a}$, increasing α will increase the influence of the term ξ_x^1 on the value of $h_x(t)$, and so the neuron value σ_x will more likely tend to ξ_x^1 than to ξ_x^3.

Now, assume neuron x belongs to $\mathcal{V}_\text{d}$. Then we can similarly as before rewrite the PSP:

$$h_x(t) = \frac{1}{N}\left((2-2\alpha)\xi_x^1 \sum_{y\in\mathcal{V}_\text{a}} \xi_y^1\sigma_y(t)\right) + \frac{1}{N}\left(\xi_x^3 \sum_{y\in\mathcal{V}_\text{d}} \xi_y^3\sigma_y(t)\right).$$

We see that as we increase α, the term with ξ_x^1 becomes smaller, and loses influence over the value of the PSP. In contrast, the term with ξ_x^3 is unaffected and thus takes over the value of the PSP for large enough α. This means that the probability that a neuron x in the group $\mathcal{V}_\text{d}$ tends to ξ_x^3 increases as we increase α.

From simulations of time series of the overlaps at different points on the phase diagram (figure 19), we concluded that we need the network to operate at reasonably large values of α (to be precise, α should be just below the border between the ordered and mixed phase). The problem of a third pattern taking over the neuron activity in the group $\mathcal{V}_\text{d}$ (which contains approximately half of all neurons!) cannot be avoided.

The origin of this problem is that in the weights matrix, at the neurons where patterns 1 and 2 compete, the contributions of these competing patterns cancel each other out. The remaining pattern is the *tertius gaudens.*

7.2.2 Sparse block network PSP

In the sparse block network, patterns 1 and 2 will compete without cancelling each other out. To construct the network, we partition the set of all neurons V_N into blocks $b^{(1)} = (x_1, ..., x_L), b^{(2)} = (x_{L+1}, ..., x_{2L}), ...$ of length L (where we assume that L divides $|V_N|$). The set of all blocks is B.

The weights of the network are

$$w_{xy} = \zeta_x^1\zeta_y^1 + \zeta_x^2\zeta_y^2 + \alpha(\zeta_x^1\zeta_y^2 + \zeta_x^2\zeta_y^1) + \zeta_x^3\zeta_y^3,$$

where $\zeta_x^i = \xi_x^i - 1/L$.

We use the following notation. Denote by $\sigma_{b^{(k)}}$ the block-index of the active neuron in block k: if $\sigma_{b_l^{(k)}} = 1$, $\sigma_{b^{(k)}} = l \in \{1, ..., L\}$. Similarly define $\xi_{b^{(k)}}^i$ to be the block-index of the active neuron of pattern i in block k. We define a delta function: $\delta(x, y) = 1$ if $x = y$, and $\delta(x, y) = 0$ otherwise.

The PSP of the sparse block network of neuron x at time t is given by

$$h_x(t) = \frac{L}{N}\sum_{y\in V_N} w_{xy}\sigma_y(t) = \frac{L}{N}\sum_{k=1}^{N/L}\sum_{l=1}^{L} w_{xb_l^{(k)}}\sigma_{b_l^{(k)}}(t). \tag{63}$$

Now, note that if there is a neuron $b_l^{(k)}$ in block k that is active at time t in the network and in pattern j, that is, $\sigma_{b_l^{(k)}}(t) = \xi_{b_l^{(k)}}^j$, then $\delta(\xi_{b^{(k)}}^j, \sigma_{b^{(k)}}(t)) = 1$, and if there is no neuron in block b that is both active at time t and in pattern ξ^j, then $\delta(\xi_{b^{(k)}}^j, \sigma_{b^{(k)}}(t)) = 0$. For any block, only one neuron can be active in the block at time t, and so

$$\begin{aligned}\sum_{l=1}^{L} \zeta_{b_l^{(k)}}^j \sigma_{b_l^{(k)}}(t) &= \sum_{l=1}^{L}\left(\xi_{b_l^{(k)}}^j - \frac{1}{L}\right)\sigma_{b_l^{(k)}}(t) \\ &= \begin{cases} 1 - \frac{1}{L}, & \text{if } \delta(\xi_{b^{(k)}}^j, \sigma_{b^{(k)}}(t)) = 1, \\ -\frac{1}{L}, & \text{if } \delta(\xi_{b^{(k)}}^j, \sigma_{b^{(k)}}(t)) = 0, \end{cases} \\ &= \delta(\xi_{b^{(k)}}^j, \sigma_{b^{(k)}}(t)) - \frac{1}{L}.\end{aligned}$$

Using this fact to rewrite the weights (63), we get

$$\begin{aligned}\sum_{l=1}^{L} w_{xb_l^{(k)}}\sigma_{b_l^{(k)}} = \zeta_x^1 &\left[\delta(\xi^1_{b^{(k)}}, \sigma_{b^{(k)}}(t)) - \frac{1}{L} + \alpha\left(\delta(\xi^2_{b^{(k)}}, \sigma_{b^{(k)}}(t)) - \frac{1}{L}\right)\right] \\ +\zeta_x^2 &\left[\delta(\xi^2_{b^{(k)}}, \sigma_{b^{(k)}}(t)) - \frac{1}{L} + \alpha\left(\delta(\xi^1_{b^{(k)}}, \sigma_{b^{(k)}}(t)) - \frac{1}{L}\right)\right] \\ +\zeta_x^3 &\left[\delta(\xi^3_{b^{(k)}}, \sigma_{b^{(k)}}(t)) - \frac{1}{L}\right].\end{aligned}$$

Again, we separately look at the case when patterns 1 and 2 are equal at neuron x ($\xi_x^1 = \xi_x^2$) and when they are not equal (either $\xi_x^1 = 1, \xi_x^2 = 0$ or $\xi_x^1 = 0, \xi_x^2 = 1$). Note that if L is of reasonable size, most of the neurons satisfy $\xi_x^1 = \xi_x^2$, as for most neurons $\xi_x^1 = \xi_x^2 = 0$ (the patterns are sparse).

If $\xi_x^1 = \xi_x^2$:

$$h_x(t) = \left(\xi_x^1 - \frac{1}{L}\right)\left[(1+\alpha)\sum_{k=1}^{N/L}\left(\delta(\xi^1_{b^{(k)}}, \sigma_{b^{(k)}}(t)) + \delta(\xi^2_{b^{(k)}}, \sigma_{b^{(k)}}(t)) - \frac{2}{L}\right)\right] + \left(\xi_x^3 - \frac{1}{L}\right)\sum_{k=1}^{N/L}\left(\delta(\xi^1_{b^{(k)}}, \sigma_{b^{(k)}}(t)) - \frac{1}{L}\right).$$

For a pattern of blocks ξ^i and random configuration of blocks σ, on average a fraction $1/L$ of the blocks of σ agree with the blocks of ξ^i, so the term $\left(\delta(\xi^1_{b^{(k)}}, \sigma_{b^{(k)}}(t)) + \delta(\xi^2_{b^{(k)}}, \sigma_{b^{(k)}}(t)) - \frac{2}{L}\right)$ is expected to be nonnegative. From this it is clear that increasing α also increases the influence of the ξ^1 term over the value of the PSP. Thus, if $\xi_x^1 = \xi_x^2 = 1$, increasing α increases the probability that in the next timestep, $\sigma_x = 1$, and if $\xi_x^1 = \xi_x^2 = 0$, increasing α decreases the probability that in the next timestep $\sigma_x = 1$.

If $\xi_x^1 = 1, \xi_x^2 = 0$:

$$\begin{aligned}h_x(t) = \sum_{k=1}^{N/L}\left(\left(1 - \frac{1+\alpha}{2}\right)\left(\delta(\xi^1_{b^{(k)}}, \sigma_{b^{(k)}}(t)) - \frac{1}{L}\right)\right) + \left(\left(\alpha - \frac{1+\alpha}{2}\right)\left(\delta(\xi^2_{b^{(k)}}, \sigma_{b^{(k)}}(t)) - \frac{1}{L}\right)\right) \\ + \left(\xi_x^3 - \frac{1}{L}\right)\sum_{k=1}^{N/L}\left(\delta(\xi^1_{b^{(k)}}, \sigma_{b^{(k)}}(t)) - \frac{1}{L}\right).\end{aligned}$$

We see that increasing α decreases the PSP, which implies that increasing α actually decreases the probability to have $\sigma_x = 1$ in the next timestep. This is the opposite of what we want, as we had $\xi_x^1 = 1$ on neuron x. Increasing α thus has the effect of increasing the probability that σ_x attains the value of the unwanted pattern ξ_x^3. The case $\xi_x^1 = 0, \xi_x^2 = 1$ is similar.

For uniformly random block patterns, given some block, the probability that pattern 1 and pattern 2 are equal on that block is $1/L$. The probability that they are not completely equal (that there is one neuron x in the block for which $\xi_x^1 = 1, \xi_x^2 = 0$, and a neuron y in the block for which $\xi_y^1 = 0, \xi_y^2 = 1$) is then $(L-1)/L$. In such blocks, there are two neurons for which the pattern values ξ_x^1, ξ_x^2 don't agree. As there are N/L blocks in total, there are thus $2 \cdot \frac{L-1}{L}\frac{N}{L}$ neurons on which patterns 1 and 2 don't agree. Therefore, by increasing the block length L, we can arbitrarily reduce the amount of neurons at which increasing α has the undesirable effect of letting an unwanted pattern take over.

The blocks of patterns still compete, but at the level of the individual neuron, the patterns are actually almost completely equal. This decreases the presence of the 'cancelling-out' effect that occurred in the Hopfield network.

7.3 Sparse 2-pattern switching network

The sparse 2-pattern switching network is a sparse implementation of the switching network as discussed in 5. It has two stored block-pattens ξ^1, ξ^2, and weights

$$w_{xy}^{\text{switch}}[\xi_x^1, \xi_x^2] = \zeta_x^1\zeta_y^1 + \zeta_x^2\zeta_y^2 + \alpha(\zeta_x^1\zeta_y^2 + \zeta_x^2\zeta_y^1),$$

again with $\zeta_x^i = \xi_x^i - 1/L$.

The 2-pattern switching network will again be the building block of our two-state Markov chain embedding weights. In the sparse case, the crossterm parameter α does not introduce interference from unwanted patterns during switching, and so one can be more confident in combining two 2-state switching networks.

The parameters α, β, γ, and L need to be determined before an embedded Markov chain can be simulated. We will analyze the influence of α, β and L on network behaviour through the phase diagram of the network at $\gamma = 0$. After that, we pick promising values for these parameters, and then analyze the parameter γ to relate energy perturbations to transition probabilities in the embedded Markov chain.

7.3.1 Phase diagram

The phase diagram of the sparse 2-pattern switching network (with $\gamma = 0$) has been simulated for $L = 4, 8, 16$ (figures 34, 35 and 36, left and middle), by running a time series of m_1 and m_2 for 50 different values of both α and β, giving a total of $50 \times 50 = 2500$ simulations. The simulations were performed with synchronous dynamics, and $N = 1024$ neurons. For each point (α, β), a new set of two patterns were generated (with an overlap of exactly $1/L$) and a time series of 1000 timesteps per neuron was run, starting the network in state $m_1 = 1, m_2 = 0$. A burn-in period of 500 timesteps was used, and during the remaining 500 timesteps the values of m_1 and m_2 were stored. In figures 34, 35 and 36, on the left, the average of the stored values of $\tilde{m}_1 = m_1 + m_2$ (blue) are shown for each point (α, β), and in the middle, the average of the stored values of $\tilde{m}_2 = m_1 - m_2$ (blue) are shown for each point (α, β).

As in the phase diagram of the Hopfield implementation of the 2-pattern switching network, there appear to be three phases: a *disordered* phase, in which both patterns have on average $1/L$ overlap with the network state; a *mixed* phase, in which the two patterns have equal average overlap with the network state, larger than $1/L$; and lastly an *ordered* phase, in which both patterns have an overlap larger than $1/L$ with the network state, but one pattern has larger overlap than the other.

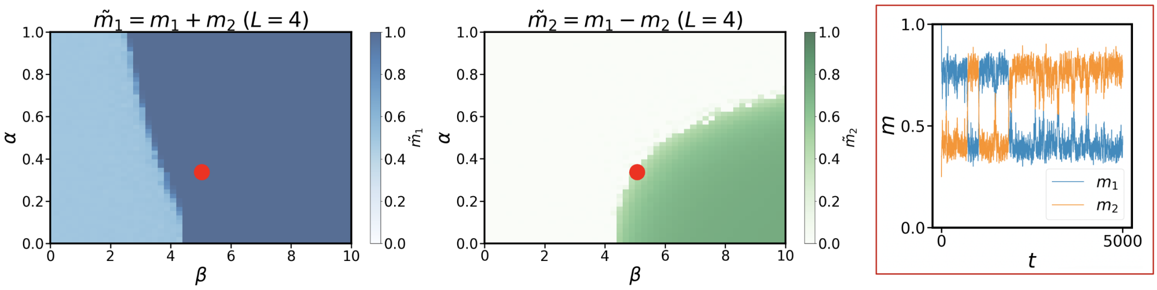


Figure 34: L=4. Left and middle: Phase diagram of the sparse 2-pattern switching network. Right: time series of the order parameters m_1 (blue) and m_2 (orange), which shows the typical behaviour of the network order parameters just below the phase transition from the ordered to the mixed phase. This particular simulation corresponds to $\alpha = 0.34, \beta = 5.0$ (red dot).

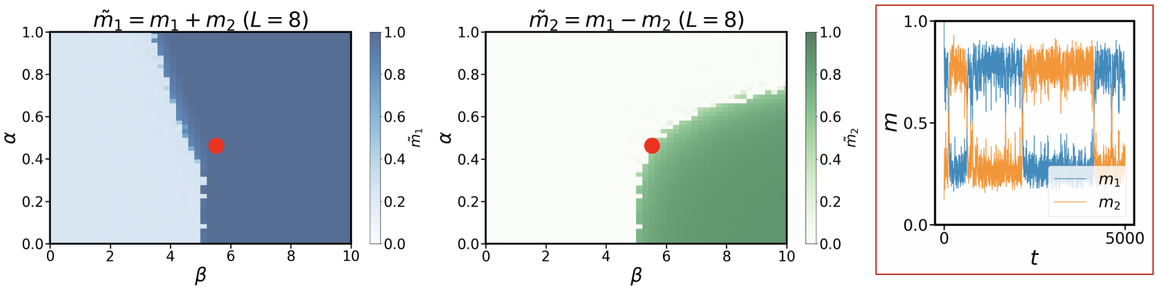


Figure 35: L=8. Left and middle: Phase diagram of the sparse 2-pattern switching network. Right: time series of the order parameters m_1 (blue) and m_2 (orange), which shows the typical behaviour of the network order parameters just below the phase transition from the ordered to the mixed phase. This particular simulation corresponds to $\alpha = 0.45, \beta = 5.5$ (red dot).

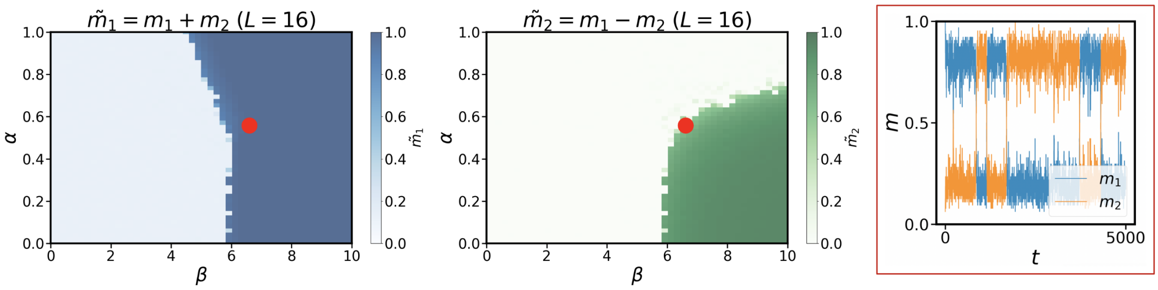


Figure 36: L=16. Left and middle: Phase diagram of the sparse 2-pattern switching network. Right: time series of the order parameters m_1 (blue) and m_2 (orange), which shows the typical behaviour of the network order parameters just below the phase transition from the ordered to the mixed phase. This particular simulation corresponds to $\alpha = 0.55, \beta = 6.5$ (red dot).

As in the Hopfield implementation of the 2-pattern switching network, the time series of the order parameters m_1, m_2 shows random oscillatory behaviour whenever the parameters α, β are close to the critical values of the phase transition between the ordered and mixed phase. This behaviour explains the ‘rough’ edges of the middle plots (green) at the border of the ordered and mixed phase. Simulations were run for 1000 timesteps per neuron, but the oscillatory behaviour appears on longer timescales (see right plots of time series; there are no more than 5 switches in 5000 timesteps in these examples). Thus, when the values of m_1, m_2 are recorded over the last 500 timesteps, the oscillatory behaviour can give radically different results even for nearby points in the phase diagram. Note that during such oscillatory behaviour, the value of $\tilde{m}_1 = m_1 + m_2$ (left plots) is unaffected.

7.3.2 State switching probabilities

Recall that during a single timestep in the embedded Markov chain, there is switching between two edge patterns for a fixed time interval, called the *switching time*, which we denote as T. At the end of the switching time, the new state (either `state 1` or `state 2`) of the embedded chain is chosen by checking which edge pattern has largest overlap with the network. The probability that the embedded chain transitions to a specific state, is thus the probability that at the end of the switching time, the corresponding edge pattern has largest overlap with the network.

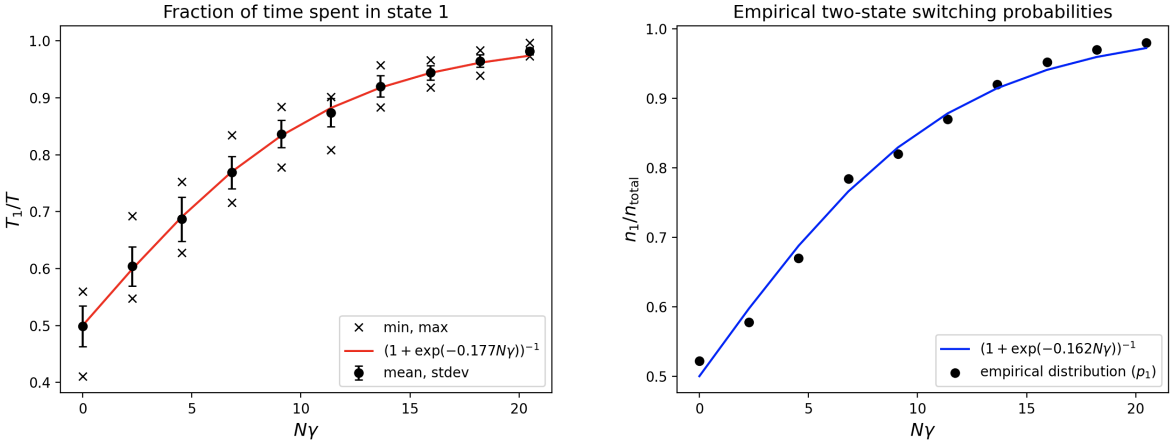


Figure 37: All simulations were done with $N = 1024, L = 8, \alpha = 0.5, \beta = 5.5$, and all patterns have overlap of exactly $1/L$. Left: The time the sparse 2-pattern switching network spent in state 1 ($m_1 > m_2$) as a fraction of the total time. 10 values of γ are plotted; for each value of γ, 20 simulations were performed. Each simulation was $T = 20000$ timesteps per neuron long. The mean, standard deviation, min and max of these 20 simulations are shown. Right: The amount of simulations which were in state 1 at their last timestep (n_1) as a fraction of the total amount of simulations (n_{total}). 10 values of γ are plotted; for each value of γ, $n_{\text{total}} = 100$ simulations were performed. Each simulation was $T = 1000$ timesteps per neuron long. The fraction n_1/n_{total} is the empirical probability that after 1000 timesteps per neuron, the network is in state 1.

Simulations were performed to obtain the probability that the sparse 2-pattern switching network is in state 1 ($m_1 > m_2$). Each simulation starts with full overlap with pattern 1; this biases the network to be in state 1 on short timescales. If we want to operate the embedded Markov chain with a not too long switching time (say, 1000 timesteps per neuron for each transition in the embedded chain), we need to check the behaviour of the network in equilibrium against the behaviour on shorter timescales. If the network reached equilibrium, there is no bias towards state 1 anymore (the network has ‘forgotten’ its initial state). If the network has approximately the same behaviour in equilibrium as it has when 1000 timesteps per neuron have passed, then it follows that the initial state is not relevant to the behaviour of the network at the timescale of 1000 timesteps per neuron.

The behaviour of the network for long timescales at $L = 8, \alpha = 0.5, \beta = 5.5$ can be deduced from the left plot of figure 37. There, the average time spent in state 1 (T_1) as a fraction of the total simulation time $T = 20000$ is plotted, along with standard deviation and the minima and maxima of 20 simulations per value of γ. We see that on this timescale, most simulations spent approximately the same amount of time in state 1 (the standard deviation is not too large), and so after 20000 timesteps the network is close to equilibrium behaviour in most simulations. However, the outliers are still relatively far away from the average, and so one can also argue that even after 20000 timesteps, the network behaviour is not close to equilibrium behaviour consistently enough. In the embedded Markov chain, to have the edge pattern switching run in equilibrium consistently would take more than 20000 timesteps per neuron.

In equilibrium, the probability to be in either state 1 or 2 is fixed, and independent of time. The empirical probability that we find the network in state 1 or 2 is equal to the fraction of time spent in state 1 or 2 respectively. Therefore, if the probability to find the network in state 1 at the end of 1000 timesteps per neuron is the same as the fraction of time spent in state 1 in equilibrium, we know that the probability to find the network in state 1 after 1000 timesteps is roughly equal to the probability to find the network in state 1 in equilibrium. By the previous reasoning, that would in turn imply that after 1000 timesteps, the network has mostly ‘forgotten’ its initial state.

On the right of figure 37, the amount of times state 1 was the final state of a simulation after 1000 timesteps per neuron was plotted as a fraction of the total amount of simulations. This is the empirical probability to find the network in state 1 after 1000 timesteps per neuron. The theory in Section 5.4 predicted that in equilibrium this probability should follow a Boltzmann distribution, which is fitted to the data. Furthermore, the fraction of time spent in state 1 in equilibrium (left plot) should also follow a Boltzmann distribution, and this appears to be true.

Both Boltzmann distributions $(1+\exp(-cN\gamma))^{-1}$ were fitted by optimizing the parameter c for the data. For the $T = 20000$ timescale data (left plot), $c = 0.177$, and for the $T = 1000$ timescale data (right plot), $c = 0.162$. As these distributions are rather similar, we can proceed to construct the embedded Markov chain using a switching time in the range of 1000 timesteps per neuron.

The probability to find the sparse 2-pattern switching network in state 1 is given by

$$p_1 \approx \frac{1}{1+e^{-0.162N\gamma}},$$

and so if one wants to implement the probability to get state 1 to be p_1 (for $p_1 \geq 0.5$), one should take

$$\gamma \approx \frac{6.17}{N} \log\left(\frac{p_1}{1-p_1}\right). \tag{64}$$

If one wants to implement a $p_1 < 0.5$, the energy perturbation γ should be added to pattern 2 instead of pattern 1, and then γ determines the desired value of $p_2 = 1 - p_1 \geq 0.5$.

7.4 Embedded Markov chains

Using the simulation data of Section 7.3, we find the parameter values $N = 1024, L = 8, \alpha = 0.5, \beta = 5.5$ suitable for constructing the embedded Markov chain network, which had the embedding weights 16. Furthermore, we derived in Section 7.2 that these weights, which contain two copies of a sparse 2-pattern switching network, will most likely not give unwanted behaviour. Finally, equation 64 describes the energy perturbation needed to make the embedded chain transition with predetermined probabilities.

We embed the Markov chain with transition matrix

$$P = \begin{pmatrix} P_{11} & 1-P_{11} \\ P_{21} & 1-P_{21} \end{pmatrix}$$

into a sparse block network as follows.

Let $\zeta^1 = \phi^1 - 1/L, \zeta^2 = \phi^2 - 1/L$ be the patterns that correspond to the embedded chain states `state 1` and `state 2` respectively. Let $\zeta^3 = \eta^{11} - 1/L,\ \zeta^4 = \eta^{12} - 1/L,\ \zeta^5 = \eta^{21} - 1/L,\ \zeta^6 = \eta^{22} - 1/L$ be the patterns that correspond to the edges of the embedded chain (recall figure 16). We can now summarize the entire embedding weights matrix as follows:

$$w_{xy} = \sum_{i,j=1}^{6} Q_{ij}\zeta_x^i \zeta_x^j,$$

where

$$(Q)_{ij} = \begin{pmatrix} 1 & 0 & \delta^{\downarrow} & 0 & \delta^{\downarrow} & 0 \\ 0 & 1 & 0 & \delta^{\downarrow} & 0 & \delta^{\downarrow} \\ \delta^{\uparrow} & 0 & 1+f_N(P_{11}) & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1+f_N(P_{21}) & 0.5 \\ 0 & \delta^{\uparrow} & 0 & 0 & 0.5 & 1 \end{pmatrix}, \tag{65}$$

and

$$f_N(x) = \frac{6.17}{N}\log\left(\frac{x}{1-x}\right).$$

Here we assumed that $P_{11} \geq 0.5$ and $P_{21} \geq 0.5$. If one wants instead a chain with $P_{12} \geq 0.5$, the term $f_N(P_{11})$ needs to be removed from Q_{33} and $f_N(P_{12})$ needs to be added to Q_{44}. Similarly, if one wants a chain with $P_{22} \geq 0.5$, the term $f_N(P_{21})$ needs to be removed from Q_{55} and $f_N(P_{22})$ needs to be added to Q_{66}.

7.4.1 Performance

Three different Markov chains have been embedded in the sparse block network to test the functionality and performance of the embedding weights matrix (65). Each Markov chain has been simulated 100 times, and each simulation consisted of 100 embedded chain timesteps (100 updates of the embedded Markov chain). The switching time of the network was 500 timesteps per neuron, and after switching another 500 timesteps per neuron were spent in the retrieved pattern, during which the corresponding embedded chain state was stored. See figure 38 for an example.

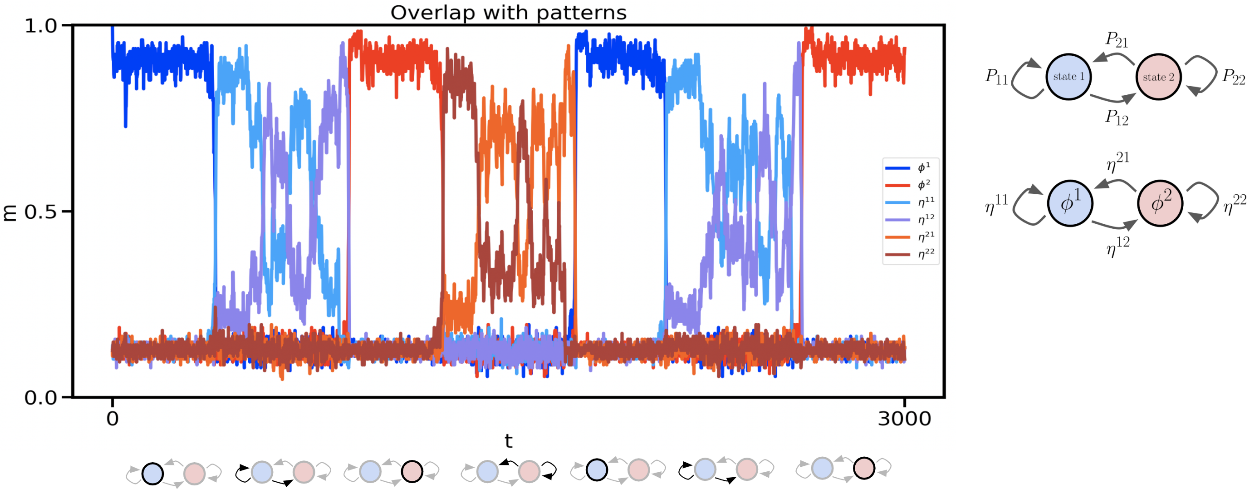


Figure 38: Example of the first three timesteps of an embedded Markov chain. Left: The blue and red timeseries are overlaps corresponding to the state attractors ϕ^1 and ϕ^2 respectively; the light blue and purple timeseries are the overlaps with the edge attractors η^{11} and η^{12} respectively, and orange and brown are the overlaps with the edge attractors η^{21} and η^{22} respectively. Right: the Markov chain, and the role of each attractor in the embedded Markov chain. Bottom: the first three timesteps of the Markov chain, associated with the time series above.

The most important aspect of an (ergodic) Markov chain is its stationary distribution. The stationary distribution of the embedded Markov chain should approach the correct stationary distribution as we increase the amount of timesteps in the chain. For each of the 100 embedded chain simulations, the empirical distribution at the last timestep was stored. The empirical distributions at the last timestep of 100 'standard' Monte Carlo simulations of the same Markov chain have also been simulated. This way the empirical distribution of the embedded chain can be visually compared to the empirical distribution that one expects from the Markov chain.

The first Markov chain for which the embedding weights have been tested has transition matrix P and stationary distribution π, given by

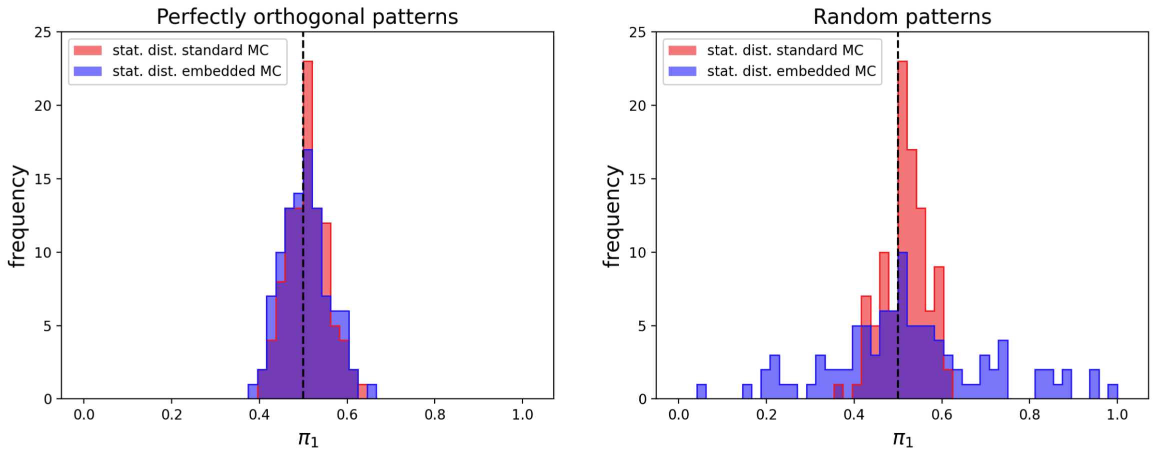
$$P = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}, \qquad \pi = (\pi_1, 1 - \pi_1) = (0.5, 0.5). \tag{66}$$

The stationary distribution is fully described by the value of π_1.

The resulting histogram of empirical distributions at the last timestep has been plotted in figure 39, for both the embedded chain simulations (blue) and standard (Monte Carlo) chain simulations (red). As all simulations are 100 chain steps long, it is expected that there is a sampling error in the obtained empirical distributions; they will sometimes deviate somewhat from the correct stationary distribution (black dashed line). However, both the embedded chain and the standard chain should have sampling errors of similar sizes.

When the embedding weights are prepared with perfectly orthogonal patterns (all patterns have overlap of exactly $1/L$), the sampling errors of the embedded chain and the standard chain are similar. Both chains have that most obtained empirical distributions are very close to the correct stationary distribution.

The embedding weights have also been constructed using random patterns (each simulation is done with new patterns). Here, a strong increase in deviation from the correct stationary distribution becomes visible for the embedded chain. This deviation cannot be explained by sampling errors alone, as having only sampling errors results in a histogram like the one for the standard chain (red).



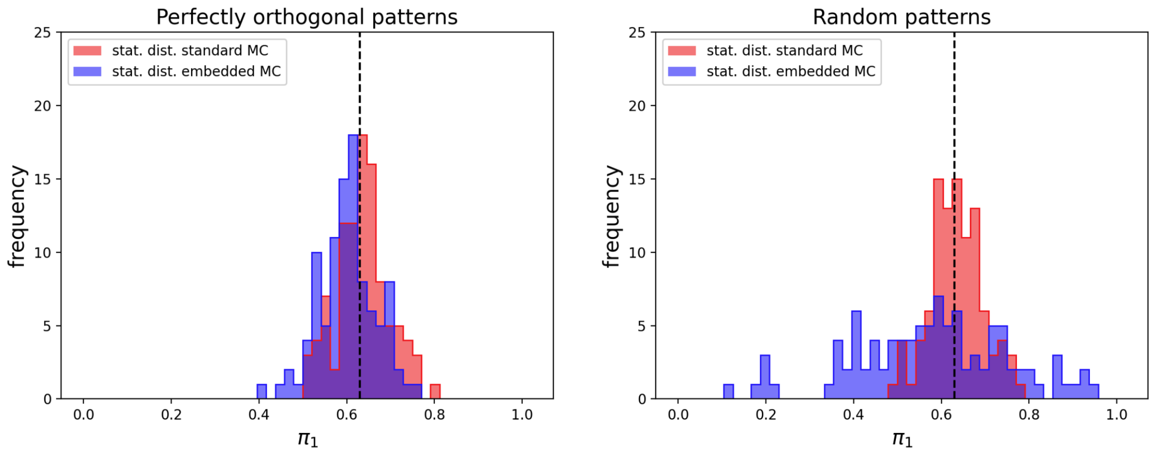
$$P = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

Figure 39: 100 simulations of the transition matrix P (bottom middle) have been performed, for both the standard (Monte Carlo) simulated Markov chain (red) and the embedded Markov chain (blue). The histograms of final distributions (π_1) after 100 chain timesteps show similarity or difference in large time behaviour between the standard and embedded chain. The correct stationary distribution (black dashed line) has also been plotted for reference.

The first tested transition matrix (66) had two equal columns, which directly implies that the eigenvalues of that matrix were $\lambda_1 = 1, \lambda_2 = 0$. By standard Markov chain theory, the second eigenvalue controls the rate of convergence to the stationary distribution. The rate of convergence for this Markov chain is zero, which means that in theory, the chain converges almost immediately to the stationary distribution.

We now test a Markov chain for which the second eigenvalue is nonzero, and so this chain has different convergence behaviour. The second Markov chain for which the embedding weights have been tested is

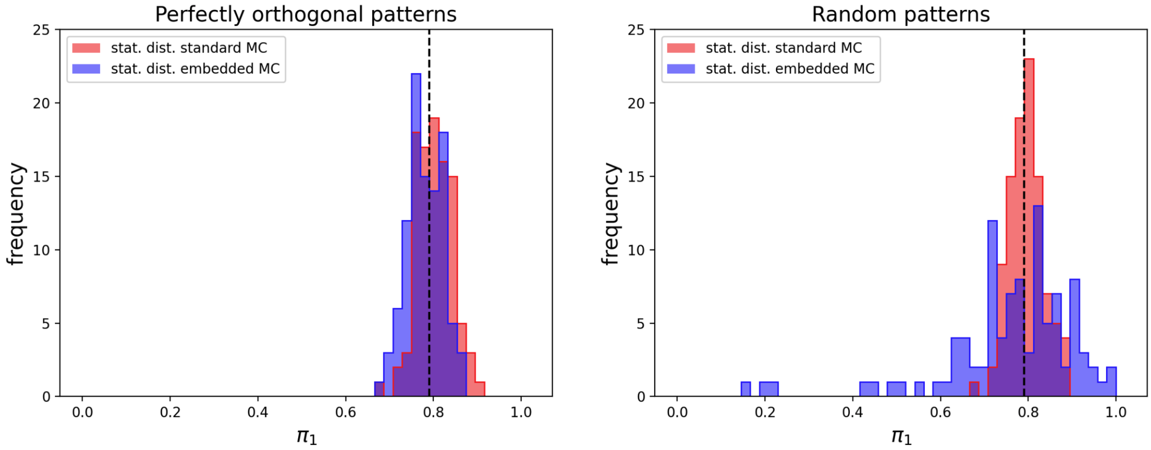
$$P = \begin{pmatrix} 0.7 & 0.3 \\ 0.5 & 0.5 \end{pmatrix}, \qquad \pi = (0.625, 0.375).$$



$$P = \begin{pmatrix} 0.7 & 0.3 \\ 0.5 & 0.5 \end{pmatrix}$$

Figure 40: 100 simulations of the transition matrix P (bottom middle) have been performed, for both the standard (Monte Carlo) simulated Markov chain (red) and the embedded Markov chain (blue). The histograms of final distributions (π_1) after 100 chain timesteps show similarity or difference in large time behaviour between the standard and embedded chain. The correct stationary distribution (black dashed line) has also been plotted for reference.

The remarks for the first tested Markov chain are also true here: the embedded chain with perfectly orthogonal patterns agrees well with the standard chain, and the embedded chain with random patterns agrees poorly. However, the histogram of the embedded chain with perfectly orthogonal patterns appears to be shifted to the left with respect to the standard chain and the correct stationary distribution. This result could be explained by a difference in transition probabilities for the embedded and standard chain: if the probability to go to/stay in `state 1` is lower in the embedded chain than in the standard chain, the stationary distribution for the embedded chain would be shifted to the left. The embedded chain might have slightly wrong transition probabilities, as these probabilities were constructed from the energy perturbations using figure 37, which was obtained using simulations (not analytically).



$$P = \begin{pmatrix} 0.79 & 0.21 \\ 0.79 & 0.21 \end{pmatrix}$$

Figure 41: 100 simulations of the transition matrix P (bottom middle) have been performed, for both the standard (Monte Carlo) simulated Markov chain (red) and the embedded Markov chain (blue). The histograms of final distributions (π_1) after 100 chain timesteps show similarity or difference in large time behaviour between the standard and embedded chain. The correct stationary distribution (black dashed line) has also been plotted for reference.

lastly, we test a Markov chain in which transitions are more asymmetric and rare (the chain mostly stays in `state 1`) than in the last two Markov chains. The third Markov chain for which the embedding weights have been tested is

$$P = \begin{pmatrix} 0.79 & 0.21 \\ 0.79 & 0.21 \end{pmatrix}, \qquad \pi = (0.79, 0.21).$$

The fact that transitions are more rare does not seem to qualitatively affect the embedded chain: we can make the same observations for this tested Markov chain as for the other two tested chains. Again we note that the histogram appears to be shifted to the left with respect to the standard chain and the correct stationary distribution, for the same reason as given for the previous Markov chain.

8 Discussion

In this work, a Hopfield network and a sparse ANN with two attractor states have been proposed that are able to switch stochastically between its attractor states in a controllable way. We call the Hopfield implementation the '2-pattern switching network', and the sparse implementation the 'sparse 2-pattern switching network'. The sparse 2-pattern switching network has been used to construct a new ANN, in which arbitrary two-state Markov chains can be embedded. The same construction for a Hopfield implementation fails to properly embed a Markov chain, due to the dense neuron activity.

The 2-pattern switching network is a Hopfield network with modified symmetric synaptic weights. The symmetric weights allow for the introduction of a Hamiltonian (energy-based) description of the dynamics. The Hamiltonian is used to equip the network with stochastic Glauber dynamics, in which a parameter β controls the amount of noise present. The stationary distribution of the dynamics is a Gibbs distribution, which allows the application of powerful statistical mechanics tools to analyze the network.

Besides the presence of the usual Hebbian weights, a symmetric crossterm, controlled with a parameter α, has been added. The parameter α introduces an extra dimension in the phase diagram of the Hopfield network. The (α, β) phase diagram has been obtained analytically using existing statistical mechanics techniques, and was verified with simulations. Besides the standard 'disordered' phase' and 'ordered' phase (also called 'memory retrieval' phase), the phase diagram contains a new phase, which we call the 'mixed' phase. In the mixed phase, the network converges to an attractor state, which is a mixture of the two 'usual' attractor states of the ordered phase. If the network is operated at values of (α, β) in the ordered phase, but very close to the critical values of the ordered-mixed phase transition, simulations showed that the network switches stochastically between the usual attractor states. By introducing an energy perturbation γ, which energetically favors the first attractor state, it was possible to bias the switching toward the network staying longer in the first attractor state. The 2-pattern switching network is thus a successful model of controllable metastable dynamics.

The metastability properties of the 2-pattern switching network have been analyzed with non-equilibrium statistical mechanics. The pathwise approach to metastability was employed to prove results on the existence of metastable states, the mean transition times and the trajectories taken from metastable to stable states.

An essential tool for the analysis of the 2-pattern switching network was the Hopfield-gCW correspondence, which is developed and proven in this work. It was shown that Hopfield Hamiltonians can be transformed into Hamiltonians of another spin system, namely those of multi-group Curie-Weiss networks, which are much more mathematically tractable. All behaviour of Hopfield networks can then be derived from studying the behaviour of multi-group Curie-Weiss networks. In this work, the correspondence is applied to 1) find a large deviations limit for the Hopfield network, and from it, derive the phase diagram of the 2-pattern switching network, 2) derive the state space of Hopfield networks, 3) show that the 2-pattern switching network cannot be used to embed Markov chains. The metastability results of the 2-pattern switching network were first proven for the 2-group Curie-Weiss network, and then translated to a result for the 2-pattern switching network through the correspondence.

A scheme for embedding a two-state Markov chain into an ANN has also been introduced in this work. Each state of the chain, and each arrow that connects two states, is stored in the ANN as an attractor state. If the Markov chain is in `state x`, then in the next timestep we push the ANN out of the attractor corresponding to `state x` and into a switching mixture of the attractor states corresponding to the outgoing edges of `state x`. After a fixed 'switching time', we push the ANN out of the edge attractor with largest overlap, and into the attractor state corresponding to the chain state the arrow points to. Two energy perturbation parameters can control the average amount of time spent in each edge attractor, and thus the probabilities of transitions in the embedded Markov chain.

It is shown that this construction gives trouble when the average network activity (how many neurons fire) is too large. In a Hopfield implementation, it is shown by analyzing the PSP that at the neurons where two switching edge attractors disagree, a third pattern has all the influence, and this leads to faulty switches to attractors that should not be able to be reached in the embedded Markov chain. Any two patterns in the Hopfield network have on average half of their neurons disagreeing, so this effect is significant. It has also been shown using the Hopfield-gCW correspondence that the Hopfield implementation contains spurious minima in the energy landscape with lower energy than the desired attractors.

In a sparse ANN, any two patterns have on average almost all neurons agreeing, and so the problem of a third pattern taking over is negligible for sparse enough patterns. It is found that in the sparse block network, which was introduced in this work and based on existing ANNs, networks with block lengths of $L = 8$ are sparse enough for the successful embedding of a two-state Markov chain via the construction above. Three embedded Markov chains have been tested by comparing the stationary distribution of the embedded chain with the expected stationary distribution (including sampling error); the stationary distribution of the embedded chain agrees with the expected result.

8.1 (Sparse) 2-pattern switching network

The weights of the 2-pattern switching network are a very straightforward and minimal modification to the widely used 'Hebbian weights', by which we mean weights formed by a Hebbian learning rule. For an ANN with two patterns and Hebbian weights, after implementing the symmetric crossterm described in this work, controllable metastable dynamics is added to the network. An interesting question is if the crossterm can also be 'learned' by the network via a simple learning rule.

The fact that metastability arises after a simple modification of the weights is a satisfactory result, as the metastability was not 'forced' onto the network via external sources of dynamics. For example, there exist models of metastable networks where the weights are asymmetric [15] [47] [59], resulting in not only an energy landscape, but also a *curl flux*, which drives the network from one attractor state to the other [60]. The metastable behaviour that follows from these asymmetric weight is the emergence of a sequence of attractor states that the network visits in a fixed order, and with a fixed time between visits. To control the time between visits, one needs fast and slow synapses, which creates an adiabatic free energy landscape [15]; the metastable dynamics follows from the dynamics of the energy landscape, which was imposed by hand. Other models study state switching induced by presenting external stimuli [28] [61], or by setting local temporal thresholds in a noisy setting [62], again increasing the amount of dynamical components of the network. Remarkably, most of these models are studied as deterministic models, and the emphasis has not been on exploring how noise and intrinsic sources of randomness might give rise to metastability. In the case of stochastic dynamics of spin systems (like the ANNs in this work), there exist well-established mathematical frameworks that prove the existence of metastability phenomena [55] [56] [57] [63]. This mathematical framework demonstrates that metastability need not be forced into the network by hand, but can be a consequence of noise. In the 2-pattern switching network, metastability was a result of setting the parameters (α, β) around critical values of a phase transition. Thus, the following approach to metastability might be more favorable, also for different sorts of ANNs: 1) implement noise into the ANN, which is controlled by some parameter. 2) Derive the phase diagram of the noisy ANN. 3) Set the parameters of the ANN close to critical values.

Furthermore, such a noisy critical ANN satisfies the *critical brain hypothesis* - the idea that neural systems operate at criticality, which optimizes its function [64] [65]. The pattern switching in the 2-pattern switching network also has the characteristics of a critical phenomenon. Switches between patterns occur through sudden, drastic changes in neural activity (see the sharp changes in m_1, m_2 in figure 22 and figures 34, 35, 36). Furthermore, the timescales of convergence to an attractor state are of the order of 10-100 timesteps per neuron, while the average period between switches is of the order of 1000-10000 timesteps per neuron; there are dynamics at multiple timescales. The dynamics are also at multiple scales of size: individual neurons fluctuate in state, but the entire network also fluctuates in state.

The 2-pattern switching network has a major drawback in its application, namely in that (as the name suggests) it only stores two patterns. Controllable stochastic switching between more than two patterns might be more complicated than the two-pattern case, as every new attractor state increases the complexity of the energy landscape, and a random walk on this landscape from one attractor to the next might get stuck in spurious attractors. Furthermore, it is expected that the (α, β) phase diagram changes. A hint that the 3-pattern case will be much more difficult is given by the Hopfield-gCW correspondence: the 2-pattern network had as corresponding multi-group Curie-Weiss model a model consisting of two disjoint 1-group Curie-Weiss networks, while a 3-pattern network results in a multi-group Curie-Weiss model that has four 1-group Curie-Weiss models, which are fully connected with each other, and both negative and positive couplings are present. More analysis is needed for switching networks with more than two patterns.

In terms of biological realism, the 2-pattern switching network has some pros and cons. Both the Hopfield and sparse implementation have some biological interpretation, as the individual neurons of the network are modeled to somewhat resemble real-world neurons. A gross simplification is of course that the model neurons only have two discrete states. More modern networks are *spiking neural networks*, which model the flow of current through the neuron with differential equations (see [66] for a review). A future work might explore if the 2-pattern switching network can be made as a spiking network, and importantly, if making the neurons spiking changes the metastable behaviour.

The average lifetime of a metastable state in the 2-pattern switching network depends on α, β and N. On the mesoscale ($N = 1000$) with reasonable noise ($\beta = 1.5$), we saw in figure 22 that the average lifetime of a metastable state is of the order of 1000-10000 timesteps per neuron. Assuming that neurons in biological systems change state on timescales of milliseconds, the 2-pattern switching network thus predicts the lifetime of metastable states to be of the order of seconds, which agrees with HMM analysis of spike trains [60].

The sparse 2-pattern switching network is superior to the Hopfield implementation (the 2-pattern switching network). Sparse ANNs are more biologically plausible [67] and much more energy efficient, as each spike brings with it a high energy cost [68]. But most importantly, the sparse 2-pattern network is able to store extra patterns in such a way that they do not interfere with the switching, while the 'dense' 2-pattern switching network suffers from spurious attractors when more patterns are added. This fact provides evidence for the idea that sparsity might be a necessity when one tries to model metastable dynamics with many attractors. It is not clear yet whether this fact is specific to our implementation of metastability, or whether it is true in general.

8.2 Embedded Markov chain

8.2.1 Generalizations

Our approach to embedding a two-state Markov chain can be generalized to embedding arbitrary Markov chains, by splitting the updating from a state with n possible transition directions into $n-1$ binary choices. This reduces any Markov chain to one with only two outgoing arrows at each state, and we can implement these into an ANN with our approach (see figure 42). If the current Markov chain state `state 1` has transition directions to `state 2`, `state 3`, ..., `state` k, we can first compute the probability that we get `state 2` or `not state 2`; the two options correspond to an arrow and have an edge attractor. After switching between the two edge attractors, if we get `not state 2`, the next options are `state 3` or `not state 3`; again, these correspond to edges and have an edge attractor. We push the network towards switching between these two edge attractors, and so on. If every time the `not`-edge was followed, the last options are `state` $k-1$ and `state` k.

The major drawback of this approach is scalability. If we split the updating from a state with n transition directions into binary choices, we introduce a new edge for each binary choice (except for the binary choice between the last states, `state` $n-1$ and `state` n). With n directions, we have $n-1$ binary choices, so $n-2$ extra edges are added; the amount of edge attractors in the network doubles. As ANNs with a finite amount of neurons have finite storage capacity, the size of the largest possible Markov chain that can be embedded is limited.

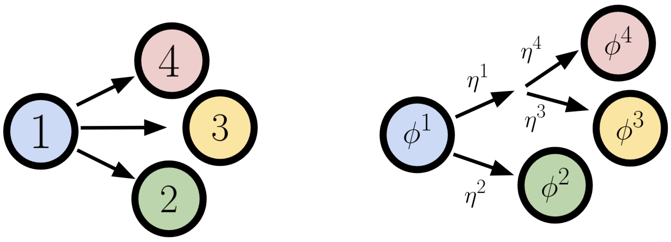


Figure 42: Example of splitting outgoing edges of a state into binary choices. The edges have attractor states η^1 (`not state 2`), η^2 (`state 2`), η^3 (`state 3`) and η^4 (`state 4`).

A second way to generalize to arbitrary Markov chains with states with more than two outgoing edges is to first generalize the 2-pattern switching network to an n-pattern switching network, in which the network stochastically jumps between n attractor states in a controlled fashion. One would need $n-1$ parameters similar to the energy perturbation parameter γ in order to control the depths of the energy wells of the attractors. It is at present unclear how a generalized crossterm should be implemented (whether it should contain more than one parameter for example) and to what sort of phase diagram it leads (which will surely depend on the particular choice of crossterm). While n-pattern switching networks could be used to implement larger Markov chains without resorting to the addition of many extra attractors, more research is needed on what such an n-pattern switching network should look like, and in particular how controllable its metastable dynamics are (as mentioned in the previous subsection).

8.2.2 Robustness

The embedded Markov chain is robust to reasonably small amounts of noise and faulty neurons, due to its attractor state dynamics. However, some neurons are more important than others during the attractor state switching: the neurons at which the two switching patterns disagree are the ones that stochastically flip their states, resulting in the switching between overlaps, while the neurons at which the two switching patters agree are frozen in their state. This suggests that the embedded Markov chain functioning is more vulnerable to malfunctioning of neurons at which patterns disagree, and less vulnerable at neurons that are frozen during the switching. It is important for applications that the robustness of the embedded Markov chain is tested; not only by letting random neurons malfunction, but also by explicitly attacking groups of neurons at which two switching patterns disagree.

Furthermore, the robustness against noise must be tested. The switching probabilities depend on the amount of neurons N, the energy perturbation γ and the inverse temperature β; an increase in noise can be modeled by decreasing β, and a decrease in functioning neurons by decreasing N.

The network can be made more robust against noise as follows: we are free to choose the crossterm parameter α in the switching weights, as long as the network operates close to the critical (α, β) values between the ordered and mixed phase; the relationship is given by $\alpha = 1 - 1/\beta$. This means that we can pick α to be rather small (close to 1), which means that the inverse temperature $\beta = 1/T$ is also not too large; this implies the temperature T is high (close to 1). The function $1/x$ is almost unaffected by small perturbations $x \to x+\delta$ when x is large, so a small increase in temperature does not affect β much when temperatures are high. The switching probabilities were estimated to follow a Boltzmann distribution, which is a continuous function in β, so a small perturbation in β is only a small perturbation in probabilities. We conclude that a smaller α value allows the embedded Markov chain ANN to run at higher temperatures, which makes it robust to small noise. The exact relation between increase in noise and change in switching probabilities should also be simulated.

An important conclusion from Section 7.4.1 is that the embedded two-state Markov chains as constructed in this work are sensitive to the orthogonality of the six patterns (two patterns for the Markov states, four for the edges) stored in the network. Figures 39, 40 and 41 show that if the sparse patterns are created such that they are exactly orthogonal (they have exactly $1/L$ overlap with each other), the embedded Markov chain has the desired stationary distribution. However, if the patterns are created by random sampling from all possible configurations, the overlap between patterns can deviate from the value $1/L$. Such overlapping patterns create different state switching dynamics, which in turn affect switching probabilities. In effect, drawing random patterns implies that we model a Markov chain with a *random transition matrix* [69]. How the overlap relates to the transition probabilities is not known yet. However, we can already draw the conclusion that the embedded Markov chain is not very robust against changes in overlaps of the stored patterns. While the large dependence of the embedded Markov chain on the pattern overlaps seems undesirable, one might also exploit it for applications. For example, one can introduce extra width in the long-term state distribution in a controlled way by varying the overlaps between stored patterns.

8.2.3 Dependence on external stimulus

Each timestep of the embedded Markov chain needs to be initiated and terminated by turning an asymmetric weight on and off again; one parameter controls the initiation of the timestep, and one controls the termination. Turning one of these parameters on and off again can be viewed as the application of a single stimulus. An important property of this approach is that the embedded Markov chain can be updated with the correct state-dependent transition probabilities, without the need of a state-dependent stimulus; each update only requires the application of two state-independent stimuli. The updating of the embedded Markov chain with the correct state-dependent transition probabilities is thus purely facilitated by the attractor state dynamics.

The necessity of having two parameters in the weights that need to change value in order for the embedded Markov chain to update is a desirable feature if one wants to be able to control when the Markov chain updates, but in terms of biological realism it is a very ad hoc mechanism. In [70], Cotteret et al. mimic the effect of similar asymmetric weights by introducing fixed weights that push the network to a new state when a mask is applied to a specific set of neurons. It is more desirable to use this masking mechanism than to modulate weights by hand, as the masking can be applied asynchronously and with random delays [71], which makes the deterministic state switching more robust.

8.2.4 Comparison with literature

In the literature, there already exists a different way in which an ANNs can be modified or combined so that on a higher level, the switching dynamics between attractor states can be described as a Markov chain. In [72], Bernstein et al. combine three separate networks: a deterministic, noisy and mixed Hopfield network, each with a different state and update rule, so that the noise state and the deterministic state combine to move stochastically to a new deterministic state.

Our approach is to associate attractor states to both the Markov chain states, and the arrows between Markov chain states. All attractor states are stored in the same network, which makes it more biologically plausible than an implementation that requires three separate networks to interact.

8.2.5 Applications

The embedded Markov chain ANN in this work is a demonstration of the fact that with minor modifications, ANNs can perform multi-timescale computation: neurons update their state on the order of 1 update per neuron, the attractor state dynamics is on timescales on the order of 10-100 timesteps per neuron, and the embedded Markov chain updating in this work is on the order of at least 1000 timesteps per neuron.

The ability of modified ANNs to perform multi-timescale computation has already been demonstrated in the literature, in particular through embeddings of finite state machines (FSM) in ANNs [70] [73], in which the FSM state switching occurs on different timescales than the attractor dynamics. The relation of FSMs with this work is that a Markov chain can be interpreted as a probabilistic FSM, where the input at each timestep is chosen according to some probability distribution.

A future model of a network that can perform multi-timescale computation could be an ANN in which embedded Markov chains are combined with an FSM, so that the network can emulate a wide class of probabilistic FSMs. Such a probabilistic FSM would be robust to imperfect inputs and faulty neurons, due to its attractor state dynamics.

An ANN that has the ability to emulate arbitrary Markov chains can also be used to draw samples from arbitrary probability distributions through a Markov Chain Monte Carlo (MCMC) algorithm. Approaches to performing MCMC sampling with neural networks already exist in the literature, for example with stochastic spiking neural networks [74], deterministic spiking neural networks [75] or through inhibition mechanisms [76]. While we suspect that a generalization of the embedded Markov chain ANN proposed in this work can be used to perform MCMC sampling, its scalability issues make it a less suitable candidate for MCMC sampling than other networks that already exist in the literature.

Neuromorphic hardware is able to simulate ANNs [77] and SNNs [78] while having ultra-low power consumption. The embedding of a Markov chain in an ANN is a first step towards a general approach to emulating Markov chains and stochastic FSMs on neuromorphic devices. Low-power, event driven devices that can simulate stochastic FSMs have applications in for example robotics [79] and healthcare [80].

While we suspect that our approach to embedding a Markov chain in an ANN is applicable to a range of different ANNs, which makes it a general approach, it is clear that the embedded Markov chain is explored by the network in series, and it is unclear at present how one can explore the embedded Markov chain in parallel within the same ANN. Parallel computation has been achieved for a similar problem in the literature, namely simulation of multiple random walkers [81]. Future work on applications of embedded Markov chains should focus on implementing parallel simulations of the same Markov chain for increased efficiency.

The embedded Markov chain in this work only updates after an external stimulus is applied. When the embedded Markov chain is implemented in neuromorphic hardware, it can be used to create a random state-dependent output in an event-driven way (where the event is the stimulus). However, if one wants for example to sample from the stationary distribution, an average over many updates of the embedded Markov chain needs to be computed, which requires consistently updating the Markov chain for some time interval. Each update requires an external stimulus, so sampling from a stationary distribution requires an external clock that updates the Markov chain many times, before presenting output. Furthermore, recording the Markov chain state at every timestep requires an external memory. Sampling from the stationary distribution in an event-driven way, without an external clock and memory, would require a modification of the current approach so that the network locally stores the sequence of visited states, and starts and terminates Markov chain timesteps without an external stimulus. A possible solution to the second problem could be to implement slow temporal synaptic weights that push the network in and out of the attractors that correspond to the Markov chain states.

8.3 Hopfield-gCW correspondence

The main idea behind the Hopfield-gCW correspondence is to group neurons that have the same pattern values together, in order to transform the Hopfield network into a multi-group Curie-Weiss network. This work is not the first to apply such an idea: The 1-pattern Hopfield network (also called *Mattis model*) was transformed into a 1-group Curie-Weiss network by Mattis in 1976 [52]. Van Hemmen showed in 1982 that the 2-pattern Van Hemmen network (Section 3.8) is effectively a 2-group Curie-Weiss network, where the two groups are disjoint [53]. In 1986, Grensing and Kühn obtain a free energy for random pattern spin models by partitioning spins depending on pattern values at those spins, and defining 2^pCurie-Weiss order parameters on those partitions [82]. In 2009, Shkolnikov introduced 'extended order parameters' to study metastability of generalized Hopfield networks [58]. These extended order parameters are also defined on partitions of neurons, and are similar to the Curie-Weiss order parameters as constructed in the proof of Theorem 3.2. The new insight of this work is that for a p-pattern Hopfield network one only needs to construct 2^{p-1} order parameters instead of 2^p or more, and that the new 2^{p-1} order parameters are those of a spin model that currently exists in the literature in parallel, namely the multi-group Curie-Weiss model. The reduction of necessary order parameters by a factor 2 makes the analysis of Hopfield networks with a small amount of patterns but complicated weights more accessible.

The Hopfield-gCW correspondence played an important role in the mathematical analyses of the 2-pattern switching network. Its main use in this work was to provide a simpler state space for metastability analysis, namely the rectangular state space $[-1, +1]^2 \subset \mathbb{R}^2$ of the 2-gCW network. This allowed for a more convenient application of the pathwise approach of metastability, as paths that minimized their energy were paths running along the boundary of the state space, and the boundaries of the rectangular state space are particularly simple. For a Hopfield network with two patterns, the Hopfield state space is a disk in the L^1 metric (a diamond), and changing to the 2-gCW state space thus comes down to applying a rotation of 45 degrees on the state space.

While the state space of two-pattern Hopfield networks is rather elementary, the state spaces of general p-pattern Hopfield networks can become complicated. To apply the pathwise approach to metastability, it is crucial to know the space of all possible paths, and thus to know exactly what the state space looks like, also for finite N. In Corollary 3.2.1, we derive the exact Hopfield state space for any finite N. However, for metastability analysis of general Hopfield networks the Hopfield state space is no longer needed, as through the corerspondence, one can work more conveniently in the state space of the corresponding 2^{p-1}-gCW network, which is always a grid in $[-1,1]^{p-1}$.

Knowledge of the exact state space has other important applications. In Section 5.5, simulations show that two 2-pattern switching networks cannot be combined to create larger robust state-switching networks. The state space and energy landscape of the combination of two 2-pattern switching networks is complicated. Without proper mathematical tools, this network becomes a black box, and it becomes tempting to keep tuning parameters until one hopes to find the desired robust switching behaviour. However, through the correspondence and exact knowledge of the state space, we were able to examine the energy landscape of this network, and show that there are unavoidable spurious minima: the desired behaviour cannot be obtained by tuning parameters.

One can conjecture that similar correspondences between other ANNs and similar statistical mechanics models with more tractable state spaces could exist; if it is true, this could help in relieving some ANNs with complicated energy landscapes of their black box status. Theorem 3.2 shows that any state $\boldsymbol{m}$ of a Hopfield network can be written as a linear transformation of a state $\tilde{\boldsymbol{m}}$ of a multi-group Curie-Weiss network. Modern Hopfield networks have Hamiltonians of the form $H(\boldsymbol{m}) = -\sum_i F(\boldsymbol{m})$, with F some smooth non-quadratic function [45] [46] ; H is a function of $\boldsymbol{m}$, so modern Hopfield networks might also have a corresponding multi-group Curie-Weiss network, which would also have a non-quadratic Hamiltonian.

Not only the transformation from a Hopfield network to a multi-group Curie-Weiss network has applications. The fact that one can also transform multi-group Curie-Weiss networks into Hopfield networks (the other way around) shows immediately that some multi-group Curie-Weiss networks are suitable for storing and retrieving a finite amount of patterns, which makes it a useful ANN. It even has a biological interpretation: a single group of the network is a macroscopic collection of neurons, which are positively (excitatory) coupled among themselves. Different neuron groups are also coupled to each other, and the coupling can be excitatory or inhibitory between groups. In Section 5.5.1 we saw the graph of connections between neuron groups of two 2-pattern switching networks: it has some interesting features, like disconnected components and cliques. An open question is how the neuron group connections graph and their couplings relate to the behaviour of the network. Multi-group Curie-Weiss networks might prove to be very general and rich ANNs, with a host of different sorts of behaviour emerging from the connection structures between neuron groups.

The Hopfield-gCW correspondence was used to prove Theorem 3.5, which resembles a Large Deviations Principle (LDP). However, it is not a full LDP, as the Theorem is only proven for closed sets. It is expected that a full LDP can also be obtained through the Hopfield-gCW correspondence, combined with the result of Knöpfel et al. [37]. An LDP for Hopfield networks has already been proven by Bovier and Gayrard in [83]; it should be checked whether Theorem 3.5 of this work agrees with their LDP. Theorem 3.5 was used in Section 5.3 to show concentration of measure for the generalized Hopfield network (with low loading); a full LDP was not needed to achieve this.

8.4 Metastability

Metastable behaviour of Hopfield networks has been explored in the literature, for example by An der Heiden in [84] and Shkolnikov in [58]. Both use the potential theoretic approach to metastability (see [63] for an overview), in which one studies spectral properties of the transition matrix. In this work we were interested in the metastability properties of the 2-pattern switching network. Due to the low dimensionality of the state space and the straightforward energy landscape of its corresponding 2-gCW network, which is essentially a quadratic form on the square $[-1,+1]^2$, it was possible to employ the pathwise approach to metastability. The advantage of the pathwise appraoch is that it provides a clear geometric and physical picture of what paths the network takes from metastable to stable states. The tractable energy landscape of the 2-pattern switching network and the results of Cirillo et al. [57] made it straightforward to obtain estimates on transition times, gates and metastable states. The disadvantage of this approach was that unlike the potential theoretic approach, it was not possible to immediately find a transition time estimate with correct pre-factor. Another disadvantage is that we have only obtained results for large β. In the operation of the 2-pattern switching network we were most interested in metastable behaviour around $\beta = 1.5$; the theory is not strong enough to predict transition times in that regime.

Technicalities that needed to be dealt with arose from the finite N, which made the state space a grid and the energy landscape a discrete object. We have estimated the metastability phenomena for finite N by first analyzing the network for infinite N (on a continuous state space). On this continuous state space, we derive the largest energy *density* barriers the network has to overcome (which are finite), and show that in the limit $N \to \infty$, the maximum stability level *density* equals one of these largest energy density barriers.

Most interestingly, we find that the metastable behaviour of the network depends heavily on how the two partitions of the 2-gCW network are grown as $N \to \infty$. The maximum stability level depends on the crossterm parameter α. For most ratios of partition sizes n_1/n_2, the maximum stability level is that of one of the four scenarios in Theorem 6.10, and it is continuous in α. But for some ratios of partition sizes, as demonstrated in

Remark 6.10.1, the maximum stability level has a discontinuity in α, and is an increasing function of α below, and decreasing above this discontinuity. The expected transition time could spontaneously change orders of magnitude when passing through the discontinuity.

This radical change in metastable behaviour does not occur when the partitions are grown equally fast ($n_1/n_2 = 1$). A Hopfield network has almost perfectly orthogonal patterns in the large N limit, so its corresponding 2-gCW network has almost equal partition sizes; we don't expect to see the radical change in metastable behaviour. Yet, simulations show that for finite and not too large β (but larger than 1), as α is increased from 0 to 1, there is a small α window just below the critical values from the ordered to mixed phase where sharp metastable transitions suddenly occur on timescales visible to the simulation. Whether transition times spontaneously change orders of magnitude in some small window should be investigated further, both with simulations and theory.

9 Conclusion

A stochastic dense and sparse ANN with controllable metastable attractor state dynamics have been proposed. The 2-pattern switching network is a Hopfield network with generalized synaptic weights, while the sparse 2-pattern switching network is a sparse ANN with generalized synaptic weights and local inhibition. A method to embed an arbitrary two-state Markov chain in ANNs has also been proposed. The method works for the sparse ANN, but fails for the dense ANN.

Both networks contain two stored patterns. A parameter α controls the energy landscape, and a parameter β controls the noise in the network. For almost critical values of the parameters (α, β), the network shows sharp transitions between its two attractor states. A third parameter γ controls the transition probabilities. The synaptic weights of the proposed networks are symmetric; in the dense case, this allowed for a rigorous analysis of the phase diagram, transition probabilities and transition dynamics by statistical mechanics theory. Combined with another mathematical tool developed in this thesis, the Hopfield-gCW correspondence, it was possible to analyze the energy landscape of the embedded Markov chain network in the dense case, and shown that the embedded Markov chain network contains spurious attractors. The sparse case was analyzed with simulations, which verified the absence of spurious attractors.

The mathematically rigorous approach to metastability that was applied in this work provides insights into different aspects of metastability in biological neural circuits. It was possible to draw conclusions on the influence of noise and network size on attractor state switching timescales, and on the trajectories the network takes through state space from metastable to stable states. These theoretical results may in turn inform neuroscientific experiment, and contribute to a bottom-up approach to understanding higher-level function of biological neural circuits.

The embedding of an arbitrary two-state Markov chain in a sparse ANN is an example of controllable, multi-timescale, state-dependent computation with ANNs. The attractor state dynamics makes the embedded Markov chain robust against small noise and faulty neurons, but not against varying overlaps in the stored patterns of the network. Future work should investigate whether a Markov chain can be embedded such that it is robust against varying stored pattern overlaps, or if the dependence on overlaps can actually be exploited for applications.

An external stimulus is required to update the embedded Markov chain, which makes it suitable for applications that need to draw one random sample from a Markov chain with each input. The external stimulus modulates the synaptic weights, which might be less biologically plausible. Different mechanisms for initiating and terminating the Markov chain timesteps should be investigated.

Generalizations of the two-state Markov chain can be made in two ways. One possibility is to construct embedded Markov chains with arbitrarily many states from embedded two-state Markov chains. However, this approach doubles the amount of stored patterns in the network which makes it less scalable. The second is to generalize the 2-pattern switching network. It should be investigated whether controllable metastable attractor state dynamics can also be achieved for more than two patterns at the same time; this would provide an alternative to embedding Markov states with arbitrarily many states.

Lastly, perhaps the most important conclusion is on methodology: the combination of simulations and rigorous mathematics has been very fruitful in this thesis. The Hopfield-gCW correspondence and existing statistical mechanics techniques provided an exact expression for the phase diagram in the dense case, which allowed efficient parameter selection and simulation. Simulations provided evidence for a metastability phenomenon, which was then verified rigorously with the pathwise approach to metastability. The obtained expression for the expected transition time gave key insights into the metastable phenomena visible in the simulations. The Hopfield-gCW correspondence and analysis of PSPs helped in understanding the drawbacks of dense networks, ultimately leading to a successful embedding of a two-state Markov chain in a sparse network. Estimates, simulations and heuristic arguments provided an intuition for what mathematics should be developed; theorems and rigorous derivations provided new powerful ways to reason about the models, leading to the development and simulation of more interesting models, and so on. In a highly multidisciplinary field like neuromorphic intelligence, good progress can be made through collaborative research at all levels of mathematical rigour.

10 Acknowledgements

I would like to thank all my supervisors for their warm and caring guidance in this Master's project. This experience has been the perfect conclusion to my Master's, and your help has made me feel prepared for the start of my academic career.

I thank the first supervisors, Elisabetta Chicca and Réka Szabó, for their continued support of the project, and their insightful remarks and discussions. In particular, Réka Szabó made important contributions to getting the mathematical proofs in the thesis to conform to a good standard of rigour. I thank my daily supervisor, Madison Cotteret, for a very enjoyable and productive collaboration. Many results of the thesis were a product of insights gained during our weekly meetings, in particular the results on the sparse block network. He suggested that introducing sparsity into the 2-pattern switching network allows one to add more patterns into the network without creating spurious attractors.

Aernout van Enter is thanked for pointing out useful references and helpful discussions, and Wioletta Ruszel and Cristian Spitoni are thanked for pointing out the thesis of An der Heiden as a useful reference. Ivan Kryven gave continued support to the project during the last months, which allowed me to finish the thesis while also pursuing a PhD; for this I am very grateful.

To everyone in the BICS group and the Probability group: thank you for welcoming me as a student; I have really enjoyed being part of the team.

References

[1] Pentti Kanerva. “Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors”. In: *Cogn. Comput.* 1 (2009), pp. 139–159. DOI: 10.1007/s12559-009-9009-8.

[2] L.F. Abbott and W.G. Regehr. “Synaptic computation”. In: *Nature* 56A (431 2004), pp. 796–803. DOI: 10.1038/nature03010.

[3] L. Smirnova et al. “Organoid intelligence (OI): the new frontier in biocomputing and intelligence-in-a-dish”. In: *Front. Sci.* 1 (2023). DOI: 10.3389/fsci.2023.1017235.

[4] C. Mead. “Neuromorphic electronic systems”. In: *Proceedings of the IEEE* 87 (10 1990), pp. 1629–1636.

[5] M. Oh and D.F. Weaver. “Alzheimer’s disease as a fundamental disease of information processing systems: An information theory perspective”. In: *Front. Neurosci.* 17 (2023). DOI: 10.3389/fnins.2023.1106623.

[6] M.R. Khalife, R.C. Scott, and A.E. Hernan. “Mechanisms for Cognitive Impairment in Epilepsy: Moving Beyond Seizures.” In: *Front. Neurol.* 13 (2022). DOI: 10.3389/fneur.2022.878991.

[7] K. Kurata. “Information processing for motor control in primate premotor cortex”. In: *Behav Brain Res.* 61 (2 1994), pp. 135–42. DOI: 10.1016/0166-4328(94)90154-6.

[8] Lemke S.M., M. Celotto, R. Maffulli, K. Ganguly, and S. Panzeri. “Information flow between motor cortex and striatum reverses during skill learning”. In: *Curr Biol.* 34 (9 2024), pp. 1831–1843. DOI: 10.1016/j.cub.2024.03.023.

[9] G. Piccinini and A. Scarantino. “Information processing, computation, and cognition”. In: *J Biol Phys.* 37 (1 2011), pp. 1–38. DOI: 10.1007/s10867-010-9195-3.

[10] P. Miller. “Itinerancy between attractor states in neural systems”. In: *Current Opinion in Neurobiology* 40 (2016), pp. 14–22. DOI: http://dx.doi.org/10.1016/j.conb.2016.05.005.

[11] M. Abeles, H. Bergman, I. Gat, I. Meilijson, E. Seidemann, N. Tishby, and E. Vaadia. “Cortical Activity Flips Among Quasi-Stationary States”. In: *Proc. Natl. Acad. Sci. USA* 92 (19 1995), pp. 8616–8620.

[12] C. Koch. *Biophysics of Computation: Information Processing in Single Neurons.* Oxford University Press, 1998.

[13] K.H. Cohrs. “Investigation of Feedback Mechanisms in Visual Cortex using Deep Learning Models”. Master’s Thesis. University of Göttingen, 2021.

[14] S. Coombes and Kyle C. A. Wedgwood. *Neurodynamics: An Applied Mathematics Perspective.* Oxford University Press, 1998.

[15] D. J. Amit. *Modeling Brain Function: The World of Attractor Neural Networks.* Cambridge University Press, 1989.

[16] J.W. Milnor. *Attractor.* Scholarpedia. 2006.

[17] M. Khona and I.R. Fiete. “Attractor and integrator networks in the brain”. In: *Nat Rev Neurosci* 23 (2022), pp. 744–766. DOI: https://doi.org/10.1038/s41583-022-00642-0.

[18] J. S. Taube, R. U. Muller, and J. B. Jr. Ranck. “Head-direction cells recorded from the postsubiculum in freely moving rats. I. description and quantitative analysis”. In: *J. Neurosci.* 10 (1990), pp. 420–435.

[19] J. S. Taube, R. U. Muller, and J. B. Jr. Ranck. “Loss of the neural integrator of the oculomotor system from brain stem lesions in monkey”. In: *J. Neurophys.* 57 (1987), 1383–1409.

[20] J. W. Gnadt and R. A. Andersen. “Memory related motor planning activity in posterior parietal cortex of macaque”. In: *Exp. Brain Res.* 70 (1988), 216–220.

[21] P. Ashwin, M. Fadera, and C. Postlethwaite. “Network attractors and nonlinear dynamics of neural computation”. In: *Current Opinion in Neurobiology* 84 (2024). DOI: https://doi.org/10.1016/j.conb.2023.102818.

[22] J.J. Hopfield. *Hopfield network.* Scholarpedia. 2007.

[23] J.J. Hopfield. “Neural networks and physical systems with emergent collective computational abilities”. In: *Proc. Natl. Acad. Sci. USA* 79 (1982), pp. 554–2558.

[24] A.F.T. Martins, V. Niculae, and D. McNamee. *Sparse Modern Hopfield Networks.* Associative Memory & Hopfield Networks in 2023. NeurIPS 2023 workshop. 2023.

[25] G. La Camera, A. Fontanini, and L. Mazzucato. “Cortical computations via metastable activity”. In: *Current Opinion in Neurobiology* 58 (2019), pp. 37–45.

[26] D. Durstewitz and G. Deco. “Computational significance of transient dynamics in cortical networks”. In: *European Journal of Neuroscience* 27 (2008).

[27] J. Creaser, P. Ashwin, C. Postlethwaite, and J. Britz. “Noisy network attractor models for transitions between EEG microstates”. In: *J. Math. Neurosc.* 11 (1 2021).

[28] P. Miller and D.B. Katz. “Stochastic Transitions between Neural States in Taste Processing and Decision-Making”. In: *The Journal of Neuroscience* 30 (7 2010), pp. 2559–2570. DOI: 10.1523/JNEUROSCI.3047-09.2010.

[29] W. Sun, J. Winnubst, M. Natrajan, and et al. “Learning produces an orthogonalized state machine in the hippocampus”. In: *Nature* 640 (2025), pp. 165–175. DOI: https://doi.org/10.1038/s41586-024-08548-w.

[30] D.E. Rumelhart and J.L. McClelland. *Parallel Distributed Processing, Volume 1: Explorations in the Microstructure of Cognition: Foundations.* The MIT Press, 1986.

[31] J.L. McClelland. “Emergence in Cognitive Science”. In: *Topics in Cognitive Science 2* (2010), pp. 751–770. DOI: DOI:10.1111/j.1756-8765.2010.01116.x.

[32] S. Friedle and Y. Velenik. *Statistical Mechanics of Lattice Systems: A Concrete Mathematical Introduction.* Cambridge University Press, 2017.

[33] J. Hertz, A. Krogh, and R. Palmer. *Introduction To The Theory of Neural Computation.* CRC Press, 1991.

[34] C. Francesca. “Macroscopic limit of a bipartite Curie–Weiss model: a dynamical approach”. In: *J. Stat. Phys.* (2014), 1301–1319.

[35] P. Contucci and I. Gallo. “Bipartite mean field spin systems. Existence and solution”. In: *Math. Phys. Elec. J.* (2008), pp. 1–22.

[36] A. Barra, G. Genovese, and F. Guerra. “Equilibrium statistical mechanics of bipartite spin systems”. In: *J. Phys. A: Math. Theor.* 44 (2011). DOI: 10.1088/1751-8113/44/24/245002.

[37] H. Knöpfel, M. Löwe, and K. et al Schubert. “Fluctuation results for general block spin Ising models”. In: *J. Stat. Phys.* (2020), pp. 1175–1200.

[38] M. Michael Fleermann, W. Kirsch, and G. Toth. “Local Central Limit Theorem for Multi-group Curie–Weiss Models”. In: *Journal of Theoretical Probability* (2021). DOI: https://doi.org/10.1007/s10959-021-01122-4.

[39] D. Sherrington and S. Kirkpatrick. “Solvable Model of a Spin-Glass”. In: *Phys. Rev. Lett.* 35 (1972), pp. 1–22. DOI: https://doi.org/10.1103/PhysRevLett.35.1792.

[40] W.A. Little. “The existence of persistent states in the brain”. In: *Mathematical Biosciences* 19 (1-2 1972), pp. 101–120. DOI: https://doi.org/10.1016/0025-5564(74)90031-5.

[41] S. Amari. “Learning patterns and pattern sequences by self-organizing nets of threshold elements”. In: *IEEE Transactions* 21 (1972), 1197–1206.

[42] P. Peretto. “Collective Properties of Neural Networks: A Statistical Physics Approach”. In: *Biol. Cybern.* 50 (1984), pp. 51–62.

[43] Johan Jarnestad/The Royal Swedish Academy of Sciences. *The Nobel Prize in Physics 2024: Popular Information.* test. https://www.nobelprize.org/prizes/physics/2024/popular-information/. 2024.

[44] M. Löwe. “On The Storage Capacity Of Hopfield Models With Correlated Patterns”. In: *The Annals of Applied Probability* 8 (4 1998), 1216–1250.

[45] H Ramsauer and et al. *Hopfield Networks is All You Need.* International Conference on Learning Representations. arXiv:2008.02217. 2021.

[46] D. Krotov and J.J. Hopfield. *Dense Associative Memory for Pattern Recognition.* arXiv preprint. arXiv:1606.01164v2. 2016.

[47] J. Buhmann and K. Schulten. “Noise-Driven Temporal Association in Neural Networks”. In: *Europhys. Lett.* 4 (10 1987), pp. 1205–1209.

[48] M.V. Tsodyks and M.V. Feigel’man. “Information storage in neural networks with low levels of activity”. In: *Phys. Rev. A* 35 (1987). DOI: https://doi.org/10.1103/PhysRevA.35.2293.

[49] M.V. Tsodyks and M.V. Feigel’man. “The Enhanced Storage Capacity in Neural Networks with Low Activity Level”. In: *Europhys. Lett.* 6 (2 1988), pp. 101–105.

[50] G.M. Shim, D. Kim, and M.Y. Choi. “Statistical-mechanical formulation of the Wiiishaw model with local inhibition”. In: *Phys. Rev. A* 43 (12 1991).

[51] N. Golomb D. Rubin and H. Sompolinsky. “Willshaw model: Associative memory with sparse coding and low firing rates”. In: *Phys. Rev. A* 41 (4 1990).

[52] D.C. Mattis. “Solvable Spin Systems with Random Interactions”. In: *Physics Letters* 56A (5 1976), pp. 421–422.

[53] J.L. Van Hemmen. “Classical Spin-Glass Model”. In: *Phys. Rev. Let.* 49 (6 1982), pp. 409–412.

[54] J.L. Van Hemmen, A.C.D. Van Enter, and J. Canisius. “On a Classical Spin-Glass Model”. In: *Z. Phys. B - Condensed Matter* 50 (1983), pp. 311–336.

[55] M.E. Olivieri E. Vares. *Large deviations and metastability.* Cambridge University Press, 2005.

[56] E.N.M Cirillo, V. Jacquier, and C. Spitoni. “Metastability of Synchronous and Asynchronous Dynamics”. In: *Entropy* 24 (2022). DOI: `https://doi.org/10.3390/e24040450`.

[57] E.N.M Cirillo, F.R. Nardi, and J. Sohier. “Metastability for General Dynamics with Rare Transitions: Escape Time and Critical Configurations”. In: *J Stat Phys* 161 (2015), 365–403. DOI: `10.1007/s10955-015-1334-6`.

[58] M. Shkolnikov. *Metastability in the generalized Hopfield model with finitely many patterns.* arXiv preprint. arXiv:0903.3050v2. 2021.

[59] M. Shino, H. Nishimori, and M. Ono. “Nonlinear Master Equation Approach to Asymmetrical Neural Networks of the Hopfield-Hemmen Type”. In: *J. Phys. Soc. Jpn* 58 (3 1989).

[60] B.A.W. et al. Brinkman. “Metastable dynamics of neural circuits and networks”. In: *Appl. Phys. Rev.* 9 (2022).

[61] B. Chen and P. Miller. “Attractor-state itinerancy in neural circuits with synaptic depression”. In: *J. Math. Neurosc.* 10 (15 2020). DOI: `https://doi.org/10.1186/s13408-020-00093-w`.

[62] D. Horn and M. Usher. “Neural networks with dynamical thresholds”. In: *Phys. Rev. A* 40 (2 1989), 1036–1044. DOI: `https://doi.org/10.1103/PhysRevA.40.1036`.

[63] A. Bovier and F.D. Hollander. *Metastability: A Potential-Theoretic Approach.* Springer, 2016.

[64] M.A. Muñoz. “Colloquium: Criticality and dynamical scaling in living systems”. In: *Rev. Mod. Phys.* 90 (2018). DOI: `10.1103/RevModPhys.90.031001`.

[65] L. Cocchi, L.L. Gollo, A. Zalesky, and M. Breakspear. “Criticality in the brain: A synthesis of neurobiology, models and cognition”. In: *Prog Neurobiol.* 158 (2017), pp. 132–152. DOI: `10.1016/j.pneurobio.2017.07.002`.

[66] S. et al. Sanaullah. “Exploring spiking neural networks: a comprehensive analysis of mathematical models and applications”. In: *Comput. Neurosci* 17 (2023). DOI: `https://doi.org/10.3389/fncom.2023.1215824`.

[67] G. Palm. “Neural associative memories and sparse coding”. In: *Comput. Neurosci* 37 (2013), pp. 165–171. DOI: `10.1016/j.neunet.2012.08.013`.

[68] P. Lennie. “The Cost of Cortical Computation”. In: *Current Biology* 13 (6 2003), pp. 493–497. DOI: `https://doi.org/10.1016/S0960-9822(03)00135-0`.

[69] Y. Takahashi. “Markov Chains With Random Transition Matrices”. In: *Kodai Math. Sem. Rep.* 21 (1969), pp. 426–447.

[70] C. Cotteret, H. Greatorex, A. Renner, J. Chen, E. Neftci, H. Wu, I. Indiveri, M. Ziegler, and E. Chicca. “Distributed representations enable robust multi-timescale symbolic computation in neuromorphic hardware”. In: *Neuromorph. Comput. Eng.* 5 (2025). DOI: `10.1088/2634-4386/ada851`.

[71] M. Cotteret, H. Greatorex, M. Ziegler, and E. Chicca. “Vector Symbolic Finite State Machines in Attractor Neural Networks”. In: *Neural Computation* 36 (4 2024), pp. 549–595. DOI: `https://doi.org/10.1162/neco_a_01638`.

[72] J. Bernstein, D. Rolnick, I. Dasgupta, and H. Sompolinsky. *Markov Transitions between Attractor States in a Recurrent Neural Network.* Conference paper. AAAI Conference. 2017.

[73] PB Ashwin and C Postlethwaite. “Sensitive finite state computations using a distributed network with a noisy network attractor”. In: *IEEE Transactions on Neural Networks and Learning Systems* 29 (12 2018), pp. 5847 –5858. DOI: `https://doi.org/10.1016/S0960-9822(03)00135-0`.

[74] L Buesing, J Bill, B Nessler, and W Maass. “Neural Dynamics as Sampling: A Model for Stochastic Computation in Recurrent Networks of Spiking Neurons”. In: *PLoS Comput Biol* 7 (11 2011). DOI: `10.1371/journal.pcbi.1002211`.

[75] M.A. Petrovici, J. Bill, I. Bytschok, J. Schemmel, and K. Meier. *Stochastic inference with deterministic spiking neurons.* arXiv preprint. arXiv:1311.3211v1. 2013.

[76] L.K. Müller and G. Indiveri. *Neural Sampling by Irregular Gating Inhibition of Spiking Neurons and Attractor Networks.* arXiv preprint. arXiv:1605.06925v2. 2016.

[77] S. Fu, L. Wu, T. Li, C. Zhang, J. Zhang, and S. Ma. *Spin-NeuroMem: A Low-Power Neuromorphic Associative Memory Design Based on Spintronic Devices.* arXiv preprint. arXiv:2404.02463v2. 2025.

[78] G. Orchard, E. P. Frady, D. B. D. Rubin, S. Sanborn, S. B. Shrestha, F. T. Sommer, and M. Davies. "Efficient Neuromorphic Signal Processing with Loihi 2". In: *IEEE Workshop on Signal Processing Systems (SiPS)* (2021), 254–259. DOI: 10.1109/SiPS52927.2021.00053.

[79] A. Martinoli, K. Easton, and W. Agassounon. "Modeling swarm robotic systems: A case study in collaborative distributed manipulation". In: *Int. J. Robotics Res.* 23 (2004), pp. 415–436. DOI: 10.1177/0278364904042197.

[80] C. De Luca, M. Tincani, G. Indiveri, and E. Donati. "A neuromorphic multi-scale approach for real-time heart rate and state detection". In: *npj Unconv. Comput.* 2 (6 2025). DOI: 10.1038/s44335-025-00024-6.

[81] W. Severa, R. Lehoucq, O. Parekh, and J.B. Aimone. *Spiking Neural Algorithms for Markov Process Random Walk.* arXiv preprint. arXiv:1805.00509v1. 2018.

[82] D. Grensing and R. Kuhn. "Random-site spin-glass models". In: *J. Phys. A: Math. Gen* 56A (19 1986), pp. 1163–1157.

[83] A. Bovier and V. Gayrard. "An Almost Sure Large Deviations Principle for the Hopfield Model". In: *The Annals of Probability* 24 (3 1996), pp. 1444–1475. DOI: 10.1038/s44335-025-00024-6.

[84] M. An der Heiden. "Metastability of Markov Chains and in the Hopfield Model". PhD thesis. TU Berlin, 2006.